UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

"ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA BASE DE DATOS PARA CONTROL DE GASTOS EN VIAJES Y REPRESENTACIÓN EMPRESARIAL"

T R A B A J O E S C R I T O

EN LA MODALIDAD DE SEMINARIOS Y CURSOS DE ACTUALIZACIÓN Y CAPACITACIÓN PROFESIONAL QUE PARA OBTENER EL TÍTULO DE: INGENIERO EN COMPUTACIÓN

P R E S E N T A :

J o r g e A l b e r t o
P a c h e c o M e l é n d e z

ASESOR: M. En C. Marcelo Pérez Medel

MÉXICO, 2008.

Agradecimientos.

A mi madre Julieta Meléndez,
A mi padre Arturo Pacheco (in memoriam),
Gracias a su amor y su ejemplo fue posible llegar a este punto en mi vida.

A mis profesores del diplomado:
Ing. Paola Medina Mora Barrera
Ing. Pedro Gabriel Ramírez Hernández
Por su apoyo incondicional.

A mi tutor:
M. en C. Marcelo Pérez Medel
Por su invaluable ayuda en la realización de este trabajo.

A mis revisores por su tiempo y sus valiosos comentarios:
Ing. Silvia Vega Muytoy
Ing. Juan Gastaldi Pérez
M. en C. Marcelo Pérez Medel
M. en I. Carlos Omar de la Rosa Delfín
Ing. Gabriel Ortiz Cordero

Tabla de contenido.

Justificaciones y alcances.

Justificación del diplomado.

La motivación de haber tomado el diplomado en diseño de sistemas de información orientado a negocios con SQL Server y Oracle fue principalmente el hecho de ofrecer la opción de titulación, la cual en mi caso ya se había postergado por demasiado tiempo debido principalmente a factores económicos que me obligaron a comenzar a trabajar en la iniciativa privada para financiar la segunda mitad de mis estudios. En este sentido debo mencionar que estoy eternamente agradecido con mi Universidad y con mis profesores de la que en ese entonces se llamaba ENEP Aragón, ahora Facultad de Estudios Superiores (FES) Aragón, ya que con lo que aprendí aquí en los primeros seis semestres de la carrera, fue suficiente para integrarme a la comunidad económicamente activa y poder así ayudarme en mis gastos y los de mi familia.

Justificación del proyecto elegido para este trabajo.

La razón de haber escogido el manejo y control de los gastos de viajes y representación de una empresa es porque estos temas me atraen desde el punto de vista de que significan un reto para mi, ya que es la primera vez que tengo la oportunidad de hacer todo el ciclo de vida de una aplicación y en especial porque se trata de una aplicación enfocada a la web ya que actualmente es una tendencia el que los sistemas empresariales se estén abriendo a tener acceso y administración vía web.

Alcances de este trabajo.

El proyecto desarrollado en este trabajo es un sistema pensado para ser una aplicación comercial aunque por razones de tiempo que habría que invertir para llevarlo a ese grado de complejidad, he propuesto, como se verá mas adelante, que el sistema se desarrollo en base al modelo de espiral, el cual va generando versiones cada vez mas completas y depuradas de un producto, de tal manera que como alcance de este trabajo voy a llevarlo hasta la primera iteración o versión del sistema la cual será un conjunto reducido pero funcional de la versión final proyectada que cumple con el propósito de mostrar la aplicación de la teoría y conocimientos prácticos aprendidos en este diplomado y a la vez presentar un producto “vivo” de tales conocimientos. Ahora bien, el curso en el que se basa este reporte fue acerca de manejo de proyectos en forma general y específicamente, aplicado a bases de datos por lo que aquí, de igual manera lo estoy enfocando a un caso de análisis, diseño y desarrollo de base de datos sin embargo, la base de datos por si sola no tiene gran utilidad, por lo que he decidido agregar una interfase de usuario y poder así hacer mas “tangible” el producto final, aunque, por cuestión de tiempo no ha sido posible incluir el diseño de la misma.

A propósito de la interfase de usuario, y debido al tipo de aplicación desarrollado en este trabajo, la mejor opción sin duda es hacerlo sobre la plataforma web, dado que está destinada a servir a usuarios que no están nunca en un lugar fijo, por lo que una aplicación basada en otra plataforma no sería adecuada.

Tanto la empresa como los empleados mencionados son ficticios pero sirven para enmarcar la problemática y las entidades con las cuales se relaciona el sistema. Por supuesto tratando de apegarse lo mas posible a la dinámica que, en mi experiencia, tiene una empresa.

Capítulo I. Generalidades del control de gastos en viajes y representación empresarial.

1.1. Introducción.

El presente trabajo aborda de manera general el tema de los gastos en viajes y representación en que incurren los empleados de las empresas e instituciones, ya sea gubernamentales o privadas, en el cumplimiento de sus labores y cómo estos pueden controlarse para que no se salgan de los límites y/o políticas establecidas, si es que las hay.

Hablando en términos de control, los gastos de viajes y representación son uno de los más controlables[1], el problema es que no siempre tienen las empresas o instituciones los controles adecuados o eficaces para hacerlo. Sin embargo, en un mundo en el que los precios de los combustibles están incrementando su costo constantemente, y por ello el de los boletos de avión, autobús e incluso un simple recorrido en automóvil cuesta más cada día; es importante entonces tener bajo control las erogaciones por este concepto. Si a lo anterior sumamos que los empleados no siempre están muy interesados en “cuidar” el dinero de la empresa, tenemos que este es un rubro por el que cada vez mas empresas están preocupándose.

Según los expertos, no es una gran ciencia el meter bajo control estos gastos; ya que existen ciertas directrices que se pueden seguir para realizarlo:

- Definir reglas mediante el desarrollo de una política de viajes y gastos de representación sólida y ponerla por escrito. Entre más específico sea, mejor el control que va a ayudar a proporcionar

[1] Según múltiples fuentes citadas en la sección de bibliografía.

- Utilizar bien el poder adquisitivo de la empresa o institución consolidando el gasto a través de una empresa de viajes corporativa y buscando establecer convenios de descuentos a cambio de lealtad.
- Justificar mejor los viajes y comidas relacionadas con labores profesionales. A veces es suficiente con una simple llamada o inclusive, una videoconferencia.
- Inspeccionar detenidamente los informes de gastos. Este es un punto medular para el control de este tipo de cargos ya que es en los informes de reembolso donde se genera una gran cantidad de “errores” y fraudes por parte de empleados poco éticos. La mejor manera de garantizar el cumplimiento de las políticas y reducir errores y fraudes es implantar un sistema automatizado de control de viajes y representación.

Es precisamente en el primero y último de los puntos anteriores en los que se basa este trabajo, el cual mediante las técnicas aprendidas en el diplomado, desarrolla un análisis del problema en términos de manejo de proyectos, propone, desarrolla e implementa una solución informática mediante una base de datos que tomando en cuenta las políticas y lineamientos de gastos de una empresa, automatiza el proceso de solicitud, autorización y reembolsos de gastos de viaje y representación, evitando así una gran cantidad de situaciones que se pueden dar en contra del patrimonio de la misma.

Capítulo II. Análisis y diseño de la base de datos.

Introducción.

Derivado del diplomado que da origen a este trabajo, tenemos ahora las herramientas necesarias para llevar a cabo el análisis, diseño e implementación tanto de la base de datos como del sistema de información que la gestionará y que será la interfase con el usuario final. A continuación vamos a destacar las técnicas que vamos a usar para dicha tarea.

Vamos a usar el análisis y diseño estructurados de sistemas de información que ven al sistema como una superposición de 3 capas, a saber:

- Capa de Negocios
- Capa de Aplicación
- Capa de Base de Datos

Por otra parte, también vamos a utilizar la vista de componentes del análisis estructurado, el cual contempla los siguientes:

- Recursos humanos
- Datos
- Actividades
- Conectividad
- Tecnología

Para el desarrollo de la solución nos vamos a basar en el ciclo de vida del sistema de información compuesto por las siguientes fases:

1. Planificación
2. Análisis
3. Diseño
4. Implementación
5. Soporte

El modelo de ciclo de vida[1] que se eligió para este proyecto fue el de Espiral[2] combinado con el de cascada[3], esto, debido a que se quiere tener un desarrollo formal del proyecto pero dándole versatilidad y la posibilidad de entregar un producto al cliente en cada iteración de la espiral. Las ventajas son que se le irán entregando avances sustanciales y útiles al cliente así como la posibilidad de tener retroalimentación y poder modificar cualquier parte del sistema antes de que esté totalmente terminado. Siendo así, dentro de cada iteración de la espiral se seguirá el ciclo de vida de la cascada como si se tratase de un sub-proyecto.

[1] Un modelo de ciclo de vida define el estado de las fases a través de las cuales se mueve un proyecto de desarrollo de software.
[2] El modelo espiral de los procesos software es un modelo del ciclo de *meta-vida*. En este modelo, el esfuerzo de desarrollo es iterativo. Tan pronto como uno completa un esfuerzo de desarrollo, otro comienza.
[3] La visión del modelo cascada del desarrollo de software es muy simple; dice que el desarrollo de software puede ser a través de una secuencia simple de fases. Cada fase tiene un conjunto de metas bien definidas, y las actividades dentro de una fase contribuye a la satisfacción de metas de esa fase o quizás a una subsecuencia de metas de la fase.

2.1. Planificación.

2.1.1. Planteamiento del problema.

La empresa “Corporación Mexicana del Chocolate S. A. De C. V.” (el cliente), es una empresa que se dedica a la manufactura y comercialización de diversos productos de chocolate fino y que ha tenido un crecimiento importante en los últimos años. Sus operaciones abarcan toda la república mexicana y a últimas fechas, ha estado realizando exportaciones a los Estados Unidos, Centro y Suramérica así como varios países de Europa. Tiene una plantilla de personal entre ejecutivos de ventas y agentes de cobranza, entre otros, que para realizar sus funciones tienen que estar viajando continuamente tanto al interior de la república como fuera de ella. Estos viajes le representan a la empresa un gasto importante que según un estudio reciente tiende a incrementarse considerablemente ante la subida de los precios en el petróleo (que repercute directamente en los precios de transportes ya sea terrestres o aéreos) y los crecientes viajes internacionales.

Preocupados por esta situación, los directivos encargaron a una firma externa la realización de un estudio al respecto encontrando que uno de los puntos de mejora es la implementación de un sistema automatizado de solicitud, autorización y comprobación de gastos de viajes y representación.

Por su parte el departamento de informática en conjunto con el de métodos y procedimientos, estudiaron las alternativas disponibles y acordaron que la mejor sería un desarrollo a la medida de la empresa. De esta forma, después de estudiar varios candidatos y sus propuestas, han optado por encargar el desarrollo a nuestra firma considerando las siguientes características que se deben de cumplir:

- Debe ser un sistema capaz de llevar el control de todos los gastos relacionados con viajes y representación de la empresa, pudiendo configurar diferentes conceptos como anticipos y reembolsos, Hoteles, transportes, alimentación, renta de autos y otros.

- Contar con una interfase hacia los sistemas de la empresa que se necesite, como contabilidad y recursos humanos.

- Debe tener la capacidad también de poder configurar e implementar las políticas de viajes y gastos de representación así como opción de modificarlas según las necesidades de la empresa.

- También debe ser capaz, en los casos que aplique, de recabar y almacenar en medio electrónico todas las facturas, recibos y cualquier otro comprobante que ampare un determinado gasto.

- Con el fin de que la aplicación se pueda acceder desde cualquier lugar ya sea localidad nacional o internacional y para agilizar la operación, además, tomando en cuenta la naturaleza fuera de sitio de los usuarios del sistema, este tendrá que tener interfase vía web, de manera que los usuarios puedan hacer sus movimientos desde cualquier punto.

- Las operaciones principales que van a realizar los usuarios son:

    - Solicitud de anticipos para viajes, gastos de representación como comidas o cenas, eventos y otros.

- Solicitud de reembolsos por gastos hechos sin anticipo previa integración de la comprobación del gasto.
- Autorización automática y/o manual dependiendo del monto del gasto y el presupuesto asignado al centro de costos[4] .
- Dependiendo del monto, el gasto puede tener que autorizarse por varios niveles de mando, inclusive el director general.

[4] CENTRO DE COSTOS Son las áreas de la empresa que tienen manejo y control sobre el consumo de recursos (material, mano de obra, etc.). En la práctica, se divide la empresa en áreas de costos para facilitar el control presupuestal y los gastos de producción.

2.1.2. Objetivo general.

El objetivo principal del cliente es reducir al máximo los costos asociados a viajes y representación de la empresa, y en especial los ocasionados por descuidos, errores, o dolo del personal usuario.

2.1.3. Objetivos específicos:

Con el fin de alcanzar cabalmente el objetivo general, se identifican una serie de objetivos específicos o sub-objetivos, estos son:

- Implementación de una base de datos que guarde los movimientos relacionados a gastos en viajes y representación de la empresa.
- Desarrollo e implementación de una interfase de usuario que funcione mediante la web para poder acceder desde cualquier lugar la información en dicha base de datos.
- Desarrollo e implementación de interfases con las áreas de la empresa con que tiene contacto el sistema.
- Dado que la web es un ambiente de dominio público, establecer los medios adecuados de seguridad para el acceso al sistema.
- Desarrollo de reportes que permitan a la dirección y la alta gerencia el poder auditar los gastos relacionados con este rubro y en su caso tomar decisiones para corregir problemas o replantear las políticas aplicables.

2.1.4. Diagrama Organizacional.

Este diagrama se realiza para tener una visión clara y general de las áreas de la empresa directamente involucradas con el sistema. Se aprecian los niveles de alta dirección y de los niveles ejecutivos sólo se detallan los que tienen que ver directamente con el sistema que vamos a implementar.

Figura 2.1. Organigrama de la empresa (Parcial)

2.1.5. Solución Propuesta

Tomando en cuanta la problemática del cliente y la necesidad de entregar resultados de forma rápida, la solución que proponemos, tiene la encomienda de entregar resultados lo mas rápidamente posible. Para esto, estamos proponiendo el desarrollo del sistema con entregables por etapas (como lo mencionamos anteriormente, estamos hablando del modelo en espiral).

En la primera entrega (primera iteración de la espiral) vamos a entregar una versión mínima pero funcional del sistema que incluye la base de datos y las pantallas (interfase de usuario) esenciales para poder hacer solicitudes, autorizaciones y comprobación de gastos para reembolsos. Cualquier interfase necesaria con el sistema de la empresa se realizará a través de archivos planos de manera manual.

En la segunda entrega, se dotará al sistema de la capacidad de interactuar con el módulo de contabilidad del sistema de la empresa. Esto con el fin de poder contar con datos frescos de los catálogos contables así como de los movimientos de las tarjetas corporativas de los empleados y devolver a contabilidad la información necesaria para la realización de sus pólizas por concepto de egresos e ingresos que pueda generar el proceso. Así mismo, se implementarán los reportes para los directivos.

En la tercera y última entrega se implementará el módulo de administración del que permitirá al administrador o encargado del sistema llevar acabo tareas como mantener catálogos, modificar el comportamiento de las autorizaciones y parametrizar diversas variables así como modificar las políticas. También en esta entrega se implementa la interfase con recursos humanos.

Cabe mencionar que desde la primera entrega del sistema, este ya tiene todos los elementos necesarios para empezar a funcionar inmediatamente. También que podría haber una cuarta entrega del sistema aún no negociada con el cliente en la que se implementaría un módulo de recepción de cotizaciones de proveedores de bienes y servicios previamente registrados como tales. De manera que cuando un usuario realiza una solicitud, esta la pueden ver los proveedores interesados y proponer a consideración una cotización que puede ser autorizada manual o automáticamente con lo cual se consigue recortar el tiempo de respuesta para las solicitudes de los usuarios y a la vez negociar precios y condiciones preferenciales con dichos proveedores.

2.1.6. Justificación del cliente.

Es evidente que el proyecto tiene justificación para el cliente desde el momento en que este ha manifestado su necesidad de abatir los crecientes gastos de viajes y representación de la empresa. La dirección está consciente de que se han detectado muchas irregularidades en cuanto a la comprobación de gastos de los empleados que viajan sobretodo fuera del país, lo cual deja abierta una fuga importante de los recursos financieros con los que la empresa cuenta. Por lo tanto, el cumplimiento y satisfacción de dicha necesidad se convierte en la meta de este proyecto, el cual es decidida y abiertamente apoyado por la alta dirección de la empresa y para lo cual contamos con la cooperación y participación entusiasta de las áreas involucradas.

2.1.7. Alcances.

La primera entrega de la solución pactada con el cliente tendrá los siguientes alcances y limitaciones:

El sistema tendrá la capacidad de recibir solicitudes de anticipos y reembolsos por concepto de viajes y representación, así como de almacenar en la base de datos toda la información relevante necesaria.

Para las autorizaciones se tendrá la opción de realizarlas de manera manual con la posibilidad de que en una futura entrega sean automáticas cuando cumplan con las políticas dictadas por la dirección.

Los usuarios podrán capturar y almacenar sus movimientos antes de mandarlos para autorización.

El sistema tendrá incorporado un modulo de seguridad para controlar el acceso al mismo por medio del uso de una clave de usuario y un password o contraseña.

Las interfases del sistema con el sistema de administración de la empresa serán mediante archivos planos que pueden hacerse manual o automáticamente mediante el servicio "Data Transformation Services[5]" (DTS) de la base de datos.

Todo el manejo administrativo en esta entrega será mediante la manipulación directa de las tablas de la base de datos o bien mediante cargas con DTS, por lo mismo, estas tareas en un principio solo podrá realizarlas el encargado de la base de datos o DBA[6].

[5] Data Transformation Services (DTS) o Servicio de Transformación de Datos de SQL Server, es un conjunto de objetos y utilerías que permiten automatizar las operaciones de extracción, transformación y carga de datos desde o hacia una base de datos.
[6] DBA (Data Base Administrador) es el encargado de realizar las labores de administración y mantenimiento de las bases de datos.

2.1.8. Recursos humanos y tecnológicos.

En este apartado se detallan los recursos humanos que intervienen en el proyecto, tanto por el lado del cliente como del proveedor de la solución:

Recursos humanos por parte del cliente.

Por el lado del cliente vamos a dividir los recursos humanos que intervienen en 4 grupos:

- **Usuarios nivel 1**. Son los directivos, quienes van a recibir y valorar los reportes generados por el sistema y en base a tal información podrán tomar acciones como modificar políticas y procedimientos.

- **Usuarios nivel 2**. Estos son los usuarios encargados de autorizar (o denegar, en su caso) las solicitudes que asi lo requieran por parte de los usuarios de nivel 3. Cabe mencionar que estos usuarios a su vez serán los encargados de las áreas de la empresa que tienen a su cargo a los usuarios nivel 3 y por lo tanto son los dueños de su respectivo centro de costos.

- **Usuarios nivel 3**. Son los empleados o personal externo que va a realizar las solicitudes.

- **Administrador del sistema**. Ese es el usuario encargado de mantener los catálogos, parámetros, variables, archivos, realizar cargas de información a las tablas, etc. Este usuario se propone que sea alguien del departamento de informática ya que en un principio tendrá que tener contacto con la base de datos y realizar varias tareas propias de un DBA. Así mismo, se recomienda que sea una sola persona y que sea empleado de tiempo completo dentro de la empresa.

Recursos humanos por parte del proveedor de la solución (nosotros)

Por parte del proveedor de la solución estarán involucrados los siguientes recursos:

- El Líder de proyecto (LP). Será el encargado de supervisar el avance en tiempo y costo del proyecto así como de establecer la comunicación y la imagen ante los directivos de la empresa.

- Un analista de sistemas (AN1). Será el encargado de realizar las entrevistas con los usuarios clave, así como de realizar las especificaciones para el diseño de la base de datos y la interfase de usuario.

- Un especialista en bases de datos (BD1). Será quien se encargue de diseñar y construir la base de datos.

- Un desarrollador web (DEV1). Será él quien se dedique a diseñar las pantallas de la aplicación y realizar la codificación requerida para el funcionamiento de la interfase de usuario.

- Un documentador (DOC1). Será el encargado de recopilar, ordenar y redactar la información del desarrollo del proyecto así como los manuales del sistema.

Recursos tecnológicos.

Los recursos tecnológicos involucrados en el proyecto los vamos a dividir en dos ambientes que vamos a manejar para mantener separados los datos de la operación del cliente (ambiente productivo) de nuestro entorno de desarrollo (ambiente de desarrollo) evitando así caer en situaciones que podrían poner en riesgo la información y los datos del cliente. Finalmente, estos recursos tecnológicos los vamos a dividir en:

Hardware.

Para el ambiente productivo:

- 1 servidor de la base de datos.
- 1 servidor web.
- 1 Servidor de correo electrónico.

Para el ambiente de desarrollo:

- 1 estación de trabajo para el líder de proyecto.
- 1 estación de trabajo para el analista.
- 1 estación de trabajo para el especialista de la base de datos/desarrollador web configurada como servidor web y de base de datos.
- 1 estación de trabajo para el documentador.

Nota: En el caso de los tres servidores del ambiente productivo, estos pueden residir en un mismo equipo físico si cumple con los requerimientos del software correspondiente de cada servidor como por ejemplo la cantidad de memoria ram, el espacio en disco duro, la velocidad del procesador, etc.

Software:

- SQL Server 2000 Professional edition (1 licencia para el ambiente de desarrollo).
- SQL Server Standard Edition (1 licencia para el servidor de la base de datos del ambiente productivo).
- Microsoft Project o equivalente (1 licencia para el líder de proyecto).
- Microsoft Office o equivalente (4 licencias, una por cada estación de trabajo de las mencionadas en la sección de hardware).
- Windows 2000 o Windows Server 2003 o 2005 con IIS[7] versión 5 o 6 instalado. (2 licencias: ambientes de desarrollo y producción)
- Visual InterDev o Visual Studio (cualquier versión que soporte ASP 3.0) o equivalente (1 licencia para el ambiente de desarrollo web).

[7] IIS (Internet Information Services) Es el componente del sistema operativo que actúa, entre otras funciones, como servidor web

2.1.9. Lista de actividades.

A grandes rasgos, las actividades que se van a realizar para el desarrollo de la solución son:

- Análisis.
- Diseño (Base de datos y aplicación).
- Desarrollo y codificación (Base de datos y aplicación).
- Pruebas.
- Documentación.
- Capacitación.
- Liberación del sistema al ambiente productivo.

En el siguiente punto se detallan estas actividades y se establece el orden en que se pueden realizar así como los tiempos que tomará llevar a cabo cada una.

2.1.10. Matriz de actividades.

Num	Task Name	Predecessors	Duration
0	**Análisis**		**9 days**
1	Entrevistas con usuarios		3 days
2	Determinar requerimientos	2	2 days
3	Elaborar documento de requerimientos	3	2 days
4	Autorización del desarrollo	4	2 days
0	**Diseño**		**4 days**
5	Elaborar diagrama ER	5	1 day
6	Elaborar casos de uso	5	2 days
7	Elaborar diagrama de estructuras	8	2 days
8	Elaborar diagrama de flujo de datos	8	2 days
0	**Desarrollo de la Base de datos**		**13 days**
9	Creación de la base de datos	10	2 days
10	Desarrollo de Stored Procedures Altas	12	2 days
11	Desarrollo destored procedures bajas	13	2 days
12	Desarrollo de stored procedures consultas	14	3 days
13	Desarrollo de stored procedures cambios	15	2 days
14	Desarrollo de vistas y queries	16	2 days
0	**Desarrollo de la interfase de usuario**		**10 days**
15	Diseño y desarrollo de pantallas	17	6 days
16	Diseño y desarrollo de reportes	19	4 days
0	**Documentación**		**12 days**
17	Manual técnico	20	6 days
18	Manual de usuario	22	6 days
0	**Pruebas**		**8 days**
19	Pruebas en desarrollo	20	3 days
20	Pruebas con usuarios clave	28	2 days
0	**Capacitación**		**8 days**
21	Capacitar usuarios clave	25	3 days
22	Capacitar usuarios finales	26	3 days
0	**Liberación**		**3 days**
23	Elaborar documentos y reabar firmas	29	0 days
24	Liberar objetos a producción	31	1 day
25	Arranque en producción	32	2 days

Tabla 2.1. Matriz de actividades

2.1.11. Asignación de recursos.

Num	Task Name	Predecessors	Resource Names
0	**Análisis**		
1	Entrevistas con usuarios		AN1
2	Determinar requerimientos	2	AN1
3	Elaborar documento de requerimientos	3	AN1
4	Autorización del desarrollo	4	LP
0	**Diseño**		
5	Elaborar diagrama ER	5	AN1
6	Elaborar casos de uso	5	AN1
7	Elaborar diagrama de estructuras	8	AN1
8	Elaborar diagrama de flujo de datos	8	AN1
0	**Desarrollo de la Base de datos**		
9	Creación de la base de datos	10	DB1
10	Desarrollo de Stored Procedures Altas	12	DB1
11	Desarrollo destored procedures bajas	13	DB1
12	Desarrollo de stored procedures consultas	14	DB1
13	Desarrollo de stored procedures cambios	15	DB1
14	Desarrollo de vistas y queries	16	DB1
0	**Desarrollo de la interfase de usuario**		
15	Diseño y desarrollo de pantallas	17	DEV1
16	Diseño y desarrollo de reportes	19	DEV1
0	**Documentación**		
17	Manual técnico	20	DOC1
18	Manual de usuario	22	DOC1
0	**Pruebas**		
19	Pruebas en desarrollo	20	AN1,DEV1,LP
20	Pruebas con usuarios clave	28	AN1,LP,DEV1
0	**Capacitación**		
21	Capacitar usuarios clave	25	AN1
22	Capacitar usuarios finales	26	AN1
0	**Liberación**		
23	Elaborar documentos y reabar firmas	29	LP
24	Liberar objetos a producción	31	DEV1
25	Arranque en producción	32	AN1,LP,DB1

Tabla 2.2. Asignación de recursos a tareas

2.1.12. Matriz de tiempos.

MATRIZ DE TIEMPOS

Actividad	O	M	P	T
1	2	2	4	3
2	1	1	2	2
3	2	1	3	2
4	1	1	2	2
5	1	1	1	1
6	2	2	2	2
7	1	1	2	2
8	2	2	2	2
9	2	1	2	2
10	1	1	2	2
11	2	2	2	2
12	2	2	3	3
13	2	2	2	2
14	1	1	2	2
15	5	5	6	6
16	3	3	4	4
17	5	5	6	6
18	5	5	6	6
19	2	2	3	3
20	1	1	2	2
21	2	2	3	3
22	3	2	3	3
23	0	0	0	0
24	1	1	1	1
25	1	1	2	2

Tabla 2.3. Matriz de tiempos

2.1.13. Cálculo del tiempo del proyecto

Para Calcular el tiempo total que va a durar el proyecto antes de poder realizar nuestra entrega usaremos la siguiente fórmula:

$$T = (O + 4M + P) / 6$$

DONDE: O = Tiempo Optimo
M = Tiempo Medio
P = Tiempo Pésimo
T = Tiempo Estándar

Tenemos entonces:

TOTAL TIEMPO ÓPTIMO = **50**

TOTAL TIEMPO MEDIO = **47**

TOTAL TIEMPO PÉSIMO = **67**

TOTAL TIEMPO ESTÁNDAR = **61**

El tiempo total del proyecto será el tiempo estándar, **61 Días**[8].

[8] Esta fórmula está calculada para darle al tiempo medio una proporción mayor que los tiempos optimo y pésimo que influyen. Esta proporción es de cuatro(4) a seis(6).

2.1.14. Costos y tiempos finales.

Para el cálculo de los costos tenemos la siguiente tabla de tarifas para los recursos externos al cliente que intervienen en el desarrollo del proyecto:

Clave	Descripción	% Asignación	Tarifa	Tipo tarifa
AN1	Analista de sistemas	100%	3,000.00	día
LP	Líder de proyecto	100%	850.00	hora
DB1	Data Architect	100%	550.00	hora
DEV1	Desarrollador web	100%	550.00	hora
DOC1	Documentador	100%	350.00	hora

Tabla 2.5. Tarifas de recursos asignados al proyecto

En la tabla anterior podemos ver que existen diferentes tipos de tarifas, por ejemplo en el caso del analista de sistemas la tarifa es diaria en vez de por hora, esto se decidió por el hecho de que este recurso está involucrado en una gran parte del desarrollo del proyecto y era mas conveniente contratarlo por día que por hora. De esta manera se logra reducir el costo de los recursos externos.

En el caso del resto de los recursos fue mejor la opción de tarifa por hora.

En la siguiente tabla se puede apreciar por cada grupo de tareas cuales serán los costos, tanto en tiempo como en dinero, de a cuerdo a las tarifas pactadas con cada recurso.

Num	Task Name	Predecessors	Duration	Cost	Resource Names
0	**Análisis**		**9 days**	**$34,600.00**	
1	Entrevistas con usuarios		3 days	$9,000.00	AN1
2	Determinar requerimientos	2	2 days	$6,000.00	AN1
3	Elaborar documento de requerimientos	3	2 days	$6,000.00	AN1
4	Autorización del desarrollo	4	2 days	$13,600.00	LP
0	**Diseño**		**4 days**	**$21,000.00**	
5	Elaborar diagrama ER	5	1 day	$3,000.00	AN1
6	Elaborar casos de uso	5	2 days	$6,000.00	AN1
7	Elaborar diagrama de estructuras	8	2 days	$6,000.00	AN1
8	Elaborar diagrama de flujo de datos	8	2 days	$6,000.00	AN1
0	**Desarrollo de la Base de datos**		**13 days**	**$57,200.00**	
9	Creación de la base de datos	10	2 days	$8,800.00	DB1
10	Desarrollo de Stored Procedures Altas	12	2 days	$8,800.00	DB1
11	Desarrollo destored procedures bajas	13	2 days	$8,800.00	DB1
12	Desarrollo de stored procedures consultas	14	3 days	$13,200.00	DB1
13	Desarrollo de stored procedures cambios	15	2 days	$8,800.00	DB1
14	Desarrollo de vistas y queries	16	2 days	$8,800.00	DB1
0	**Desarrollo de la interfase de usuario**		**10 days**	**$44,000.00**	
15	Diseño y desarrollo de pantallas	17	6 days	$26,400.00	DEV1
16	Diseño y desarrollo de reportes	19	4 days	$17,600.00	DEV1
0	**Documentación**		**12 days**	**$33,600.00**	
17	Manual técnico	20	6 days	$16,800.00	DOC1
18	Manual de usuario	22	6 days	$16,800.00	DOC1
0	**Pruebas**		**8 days**	**$71,000.00**	
19	Pruebas en desarrollo	20	3 days	$42,600.00	AN1,DEV1,LP
20	Pruebas con usuarios clave	28	2 days	$28,400.00	AN1,LP,DEV1
0	**Capacitación**		**8 days**	**$18,000.00**	
21	Capacitar usuarios clave	25	3 days	$9,000.00	AN1
22	Capacitar usuarios finales	26	3 days	$9,000.00	AN1
0	**Liberación**		**3 days**	**$32,800.00**	
23	Elaborar documentos y reabar firmas	29	0 days	$0.00	LP
24	Liberar objetos a producción	31	1 day	$4,400.00	DEV1
25	Arranque en producción	32	2 days	$28,400.00	AN1,LP,DB1

Tabla 2.4. Tiempos y costos finales.

De la tabla podemos ver que la sumas nos arrojan:

Total en tiempo: 67 días

Total en Dinero: $ 312,200.00

Tanto el tiempo como el monto en dinero deberán ser aprobados por la dirección de la empresa para poder finalizar el desarrollo e implementación del sistema.

Aclaración acerca de los tiempos.

Aquí es necesario aclarar que el tiempo calculado en este punto, es “tiempo corrido” y que por lo regular no tiene por que coincidir con el tiempo calculado en la matriz de tiempos, ya que aquel es el tiempo total de desarrollo (días-hombre) y en este afecta el traslape de las actividades que se realizan en paralelo (días-calendario).

2.1.15. Diagrama de Gantt.

Ya con los datos que se tienen se puede dibujar el diagrama de Gantt que queda de la siguiente manera:

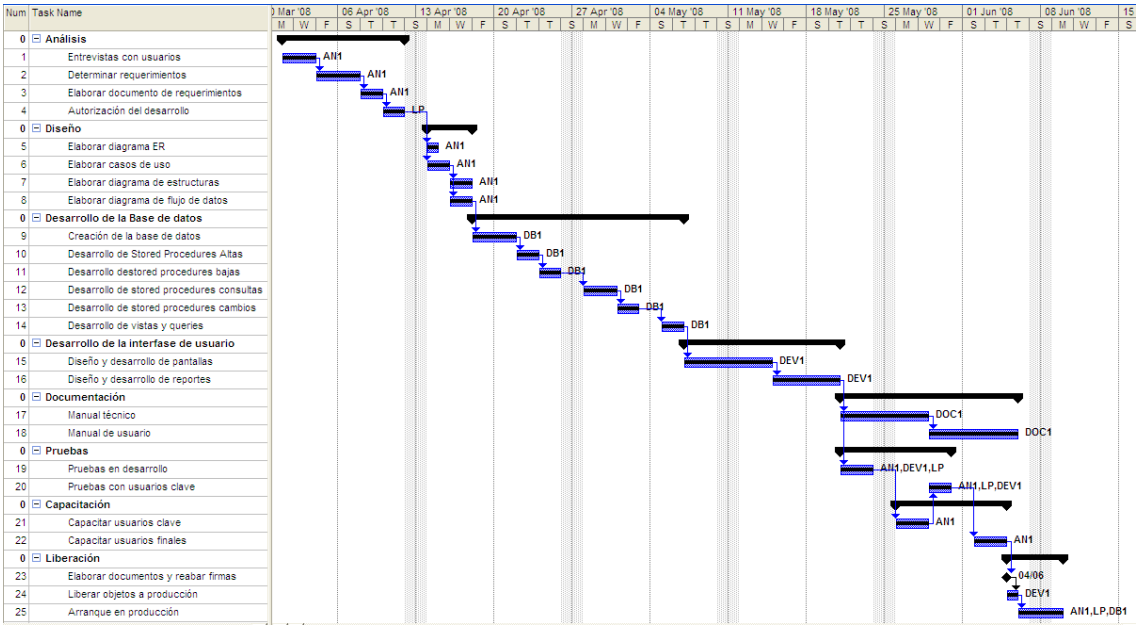


Fig. 2.2. Diagrama de Gantt

Este diagrama es el propuesto en base a las actividades y tiempos determinados anteriormente. Por supuesto que si alguna actividad no se termina en el tiempo establecido, podría impactar el tiempo final de entrega del proyecto, aunque no es este el caso.

2.2. Análisis.

2.2.1. Entrevistas con usuarios clave.

Elaboración de cuestionarios.

Se elabora un cuestionario especial para cada nivel de usuario del sistema:

Cuestionario usuarios nivel 3 (quienes realizan los gastos)

1. Cómo se realiza actualmente el proceso de solicitudes de anticipo/reembolso de gastos, ya sea de viaje o por representación de la empresa?

2. Después de que envías tu solicitud que sucede?

3. Que pasa si no es autorizado el monto de una solicitud?

4. Que problemas has tenido con el proceso?

4. Crees que el proceso de gastos de viaje y representación es actualmente claro?,

5. Consideras que es Funcional?

6. Piensas que es oportuno?

7. Crees que se puede mejorar?, cómo?

Cuestionario a Usuarios de nivel 2: Los autorizadores

1. Actualmente, cómo procesan las solicitudes de anticipos/reembolsos de gastos, ya sea de viaje o representación?

2. Que clase de problemas pueden ocurrir en dicho proceso?

3. Por cuántos VoBo's pasa la solicitud antes de ser aprobada?

4. Hay alguna política en cuanto a que gastos pueden venir sin factura y hasta que monto?, o del monto máximo para boletos de avión o reservaciones en hoteles o comidas?, o de cuáles proveedores son los que deben usarse preferentemente?

5. Crees que el proceso de gastos de viaje y representación es actualmente claro?,

6. Consideras que es Funcional?

7. Piensas que es oportuno?

8. Crees que se puede mejorar?, cómo?

Cuestionario para usuarios de nivel 1: Directores

1. Cual es el criterio para aprobar el reporte de gastos que les pasan los jefes de departamento?

2. En base a que se asigna el presupuesto de viajes y representación para los departamentos que lo requieren?

3. Que pasa cuando el departamento rebasa su presupuesto asignado?

4. Crees que el proceso de gastos de viaje y representación es actualmente claro?,

5. Consideras que es Funcional?

6. Piensas que es oportuno?

7. Crees que se puede mejorar?, cómo?

Resultados de las entrevistas.

Tomando en cuenta la cantidad de respuestas y analizando las mismas para cada una de las preguntas se llega al siguiente resumen:

1. La mayoría de los usuarios considera que la forma actual en que se lleva el proceso es poco clara.
2. Actualmente el proceso tiene muchas deficiencias.
3. No existen políticas por escrito de gastos de viajes y representación.
4. Es un proceso altamente susceptible a errores humanos.

5. Debido a la falta de control y políticas se generan pérdidas económicas cuantiosas.
6. El proceso es actualmente lento y tedioso, poco funcional.
7. Todos los usuarios coinciden en que el proceso se puede mejorar.

Conclusiones de las entrevistas.

De las entrevistas con los usuarios y el análisis posterior concluimos que es necesario establecer por escrito y publicar las políticas de gastos de viaje y representación, ya que en estas políticas es en las que el sistema automatizado que estamos desarrollando va a basarse para llevar a cabo su función de controlar y proveer mecanismos de revisión y auditoría a los gastos.
Dentro de estas políticas, es necesario cubrir, por lo menos, los siguientes puntos.

- Se tiene que definir una cadena de autorización por la que tendrá que pasar cualquier solicitud antes de ser pagada.
- Es necesario establecer una lista de conceptos de gasto autorizados y los montos máximos permitidos para cada uno.
- definir el caso de los gastos sin comprobante.
- Definir quienes serán los responsables de autorizar los gastos, estableciendo por lo menos dos personas en cada nivel por si una no puede llevar a cabo sus funciones.

2.2.2. Capas del sistema de información.

Como una primera división del sistema que vamos a analizar, lo vamos a considerar como la interacción entre tres capas principales que son:

a. Capa de negocios.
b. Capa de la base de datos
c. Capa de aplicación

Análisis de la capa de negocios.

Del trabajo realizado en la sección anterior, se desprende que el cliente por medio de su consejo de administración ha decidido implementar las siguientes políticas relativas a los gastos de viajes y representación, mismas que habrán de ser observadas y cumplidas por los usuarios a través del sistema objeto de este análisis.

Políticas del negocio.

1. Para los departamentos que lo requieran será asignado un presupuesto de gastos de viaje y representación que deberá ser justificado por el responsable del área
2. Se realizará una lista de proveedores preferidos con los cuales la empresa ha negociado tarifas preferenciales y descuentos. Los usuarios del sistema se apegarán en la medida de lo posible a esta lista.
3. Se establecerá un sistema automatizado para el manejo y autorización de estos gastos que reemplazará y ampliará las funciones del sistema actual basado en hojas de Excel.
4. Cada jefe de área será responsable del control de los gastos de su área.
5. Cualquier solicitud de gasto o reembolso, deberá pasar por un proceso de autorización, ya sea manual o en los casos en que aplique, automático.

6. Cualquier gasto no autorizado por medio de los mecanismos citados deberá ser a cargo del usuario.
7. Los anticipos y/o reembolsos autorizados serán pagados en un plazo no mayor a 15 días contados a partir de la fecha de autorización de la solicitud.
8. La forma de pago de los anticipos/reembolsos será por medio de cheque nominativo a nombre del beneficiario o por medio de transferencia electrónica de fondos a la cuenta de cheques del beneficiario.
9. Los usuarios están obligados a cumplir con las "políticas de comprobantes de gastos" para poder ser sujetos de aprobación de reembolso o bien que no se les cargue el importe de los gastos que no cumplan con dicha política.
10. Se publicará una lista de los conceptos y montos máximos autorizados por concepto de viajes y eventos de representación, misma que se implementará en el sistema automatizado, el cual obligará a los usuarios a apegarse rigurosamente a la misma.

Políticas de comprobantes de viaje.

1. Todo gasto deberá ser comprobado mediante factura u otro comprobante fiscal emitido a nombre de la empresa con los datos fiscales correspondientes. La fecha deberá ser válida según el periodo del viaje y el proveedor de acuerdo al motivo del mismo.
2. La excepción al punto anterior es en el caso de servicios que por lo general no emiten comprobantes fiscales como es el caso de taxis, propinas, transporte publico, etc., hasta por el monto autorizado dependiendo del concepto. (Tabla de conceptos y montos sin comprobante).

Condiciones para autorización de gastos sin factura:

1. que esté dentro de la lista de gastos permitidos sin factura.
2. que el monto del gasto no rebase el tope impuesto según su tipo.

Análisis de la base de datos

Entidades principales

Con la información que hasta el momento tenemos recabada, podemos empezar a bosquejar lo que será la base de datos.

Como una primera aproximación vamos a listar las entidades principales que se han identificado como necesarias para el funcionamiento del sistema. En la Columna “Origen” se aclara si la entidad será generada dentro del sistema o si viene del sistema externo usado por la empresa.

Entidad	Descripción	Origen
Anticipo	Se refiere a los datos del anticipo que se van a necesitar	SISTEMA
Área de responsabilidad	Contiene las áreas de responsabilidad a las que pueden pertenecer los conceptos de gasto.	SISTEMA
Banco	Datos de los bancos con los que tiene operaciones el cliente	EXTERNO
Centro de costo	Datos de las entidades de la empresa que afectan el costo para los cuales se podrán efectuar solicitudes	EXTERNO
Compra	Contiene los datos particulares de las solicitudes de compra.	SISTEMA
Concepto de gasto	Esta entidad guarda las categorías de los gastos que se pueden usar en el sistema.	SISTEMA
Contrato	Datos de las solicitudes tipo "contrato"	
Cuenta contable	Contendrá los datos de las cuentas contables que se afectarán como resultado de las operaciones del sistema	EXTERNO
Deposito	Datos de los depósitos hechos a favor de la empresa. Se usará para ajustar importes excedentes de los anticipos	SISTEMA
Empresa	Datos de las empresas que tendrán operaciones en el sistema	EXTERNO
Flujo de autorización	Contendrá los datos de la forma, tipo y orden en que se autorizarán las solicitudes	SISTEMA
Moneda	Datos de las diferentes monedas con las cuales los usuarios podrán realizar operaciones	EXTERNO
Nivel	Contiene los niveles de responsabilidad de los usuarios. El objetivo de este nivel es evaluar el monto límite de cada usuario que realiza solicitudes.	SISTEMA
Pago	Maneja los datos particulares de las solicitudes de pago a proveedores.	SISTEMA
Perfil	Esta entidad va a manejar los perfiles de usuario que pueden usarse en la aplicación.	SISTEMA
Presupuesto	Servirá para almacenar el presupuesto destinado, comprometido y usado por cada empresa, centro de costo cuenta contable, moneda, mes y año	SISTEMA
Proveedor	Es el catálogo de proveedores	EXTERNO
Reservación	Datos particulares de las solicitudes de reservación.	SISTEMA
Solicitud	Contendrá los datos generales de las solicitudes como el tipo de solicitud, el solicitante, la moneda de la solicitud, etc.	SISTEMA

Tipo de concepto	tipo de concepto de gasto para las solicitudes. Sirve para determinar que datos específicos aplican para cada tipo de gasto	SISTEMA
Tipo de cambio	Guardará los tipos de cambio de la moneda local con las monedas foráneas	EXTERNO
Monto límite	Esta entidad es la implementación de las políticas de gastos e viajes y representación y es donde se definen los montos máximos que puede gastar u usuario dependiendo de su nivel de responsabilidad, de l tipo de concepto de gasto, del tipo de operación y del territorio donde se lleva a cabo la operación.	SISTEMA
Tipo de operación	Es la entidad donde se determina si la operación solicitada es por ejemplo un anticipo, una comprobación de gasto o una compra.	SISTEMA
Usuario	Maneja los datos generales que se necesita tener por cada usuario que accesa al sistema.	EXTERNO

Tabla 2.5. Entidades principales del sistema.

Relaciones entre las entidades.

En la tabla siguiente se detallan las relaciones que deben existir entre las entidades para asegurar la integridad de la información:

Entidad	Se relaciona con	Por medio de
Anticipo	Solicitud	Folio de la solicitud
Área de responsabilidad	Empresa	Identificador de la empresa
Banco	Depósito Usuario Proveedor	Identificador del banco
Centro de costo	Empresa Usuario	Identificador de la empresa Ident. Empresa, Centro de costos
Compra	Solicitud	Folio de la solicitud
Concepto de gasto	Cuenta contable Monto límite	Ident. Empresa, Cuenta cont. Ident. Concepto, Ident. Empresa
Contrato	Solicitud	Folio de la solicitud
Cuenta contable	Empresa Presupuesto Concepto de gasto	Identificador de la empresa Ident. Empresa, Cuenta cont. Ident. Concepto, Ident. Empresa
Deposito	Moneda Banco Usuario Empresa	Ident. Moneda Identificador del banco Ident. Empresa, Num. Empleado Ident. Empresa
Empresa	Flujo Autorización Tipo concepto Depósito Solicitud Usuario Centro de costo Presupuesto Cuenta contable Areade responsabilidad	Ident. Empresa
Flujo de autorización	Empresa	Ident. Empresa
Moneda	Solicitud Tipo de cambio Monto límite	Ident. De moneda, moneda de pago Ident. De moneda Ident. De moneda
Nivel	Usuario Monto límite	Ident. Nivel, Ident. Empresa
Pago	Prveedor Solicitud	Ident. Proveedor Folio de la Solicitud.
Perfil	Usuario	Ident. De perfil, Ident. De empresa

	Empresa	Ident. Empresa
Presupuesto	Cuenta contable	Ident. Empresa, cuenta contable
	Banco	Ident. Banco
Proveedor	Pago	Ident. Proveedor
Reservacion	Solicitud	Folo de la solicitud
	Pago	Folo de la solicitud
	Contrato	Folo de la solicitud
	Compra	Folo de la solicitud
	Anticipo	Folo de la solicitud
	Reservación	Folo de la solicitud
	Moneda	Ident. Moneda, moneda de pago
	Usuario	Ident. Empresa, Num. Empleado
Solicitud	Empresa	Ident. Empresa
Tipo de concepto	Empresa	Ident. Empresa
Tipo de cambio	Moneda	Moneda de origen, moneda destino
	Concepto de gasto	Ident. Concepto, Ident. Empresa
	Nivel	Ident. Nivel, Ident. Empresa
	Moneda	Ident. Moneda
Monto límite	Tipo de operación	Ident. Empresa, tipo operación
Tipo de operación	Monto límite	Ident. Empresa, tipo operación
	Centro de costo	Ident. Empresa, centro costo
	Empresa	Ident. Empresa
	Depósito	Ident. Empresa, Num. Empleado
	Banco	Ident. Banco
	Solicitud	Ident. Empresa, Num. Empleado
	Perfil	Ident. Perfil, Ident. Empresa
Usuario	Nivel	Ident. Nivel, Ident. Empresa

Tabla 2.6. Relaciones entre las entidades del sistema.

Diagrama de entidades

A partir de los datos de las tablas anteriores podemos realizar el siguiente diagrama de las entidades principales que estarán en el sistema:

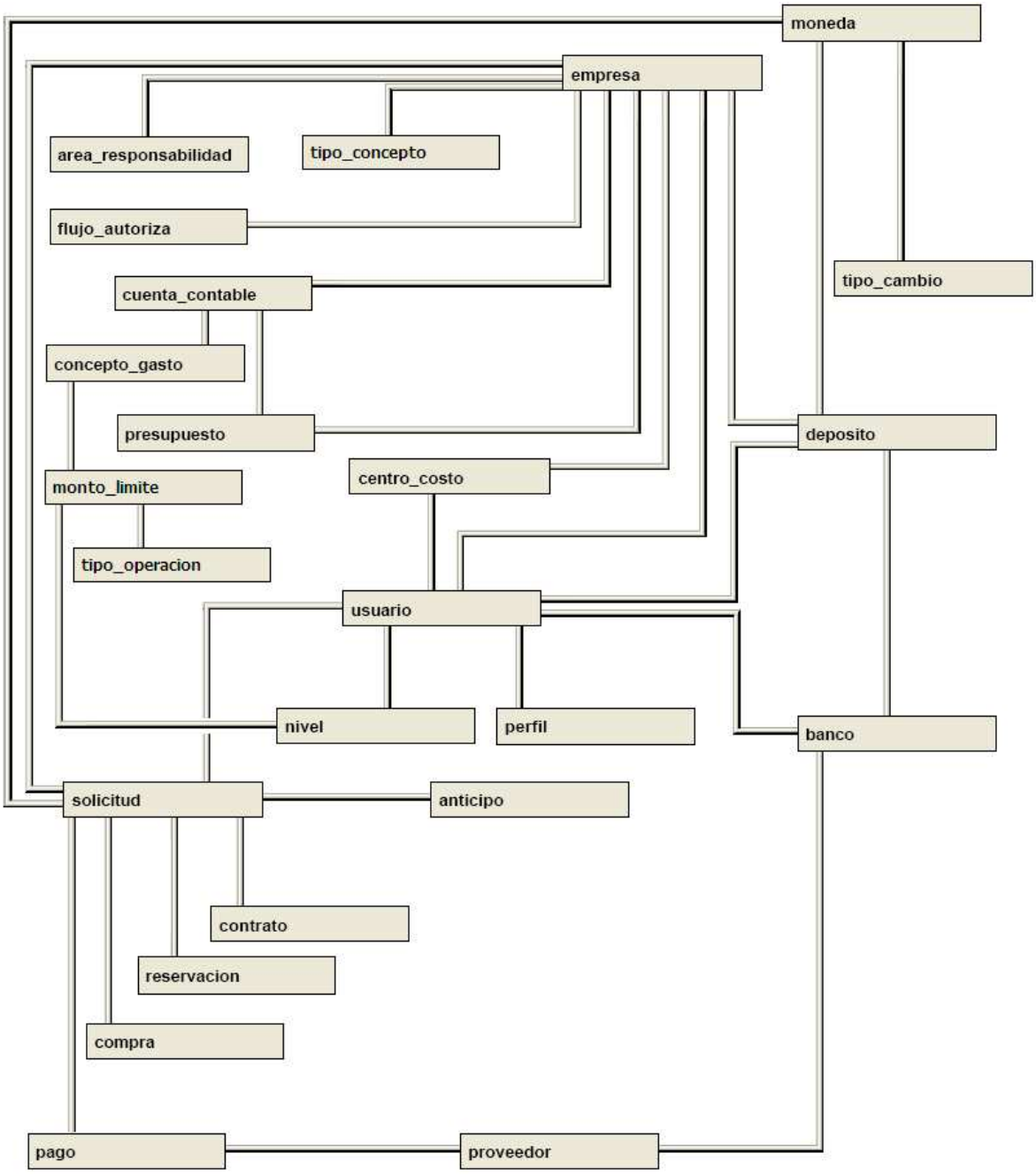


Figura 2.3. Diagrama de entidades

Datos por entidad.

Una vez definidas las entidades principales, y en base al análisis previo, se puede comenzar a definir los atributos mas importantes que se van a manejar en cada una de las entidades:

Anticipo	
Atributo	**Descripción**
Folio	Número de folio de la solicitud de anticipo
Fecha inicial	Fecha a partir de la cual se aplica el anticipo
Fecha final	Fecha final de la vigencia del anticipo
Usuario	identificador del usuario al que se le otorga el anticipo

Área de responsabilidad	
Atributo	**Descripción**
Identificador	Número consecutivo que identifique cada área
Empresa	identificador de la empresa a la que pertenece el área
Descripción	Nombre del área
Estatus	Clave que identifica el estado de dicha área en el sistema
Empleado	Responsable del área

Banco	
Atributo	**Descripción**
Identificador	Número que identifica de manera única cada banco
Nombre	Nombre del banco
Territorio	Identificación de si es nacional o extranjero
Cuenta	Número de la cuenta que tiene asignada para la empresa
Estatus	Clave que identifica su estado dentro del sistema

Centro de Costo	
Atributo	**Descripción**
identificador	Número que identifica de manera única cada centro
Empresa	identificador de la empresa a ala que pertenece
Descripción	Nombre del centro de costos
Autorizador	Número del empleado que autoriza por ese centro
Estatus	Clave que identifica su estado dentro del sistema

Compra	
Atributo	**Descripción**
Folio	Número único de una solicitud de compra
Proveedor	Número del proveedor al que se hace la compra
Departamento	Departamento del solicitante

Concepto de Gasto	
Atributo	**Descripción**
Identificador	Número único de un concepto de gasto
Tipo de concepto	identificador que determina características especiales del concepto de gasto para determinar que datos so relevantes en una solicitud
Empresa	Identificador de la empresa que lo usa
Descripción	Nombre del concepto de gasto
Estatus	Clave que identifica su estado dentro del sistema
Cuenta deducible	Cuenta contable deducible a la que pertenece al concepto
Porcentaje deducible	Porcentaje deducible del concepto
Cuenta no deducible	Cuenta contable no deducible a la que pertenece al concepto
Porcentaje no deducible	Porcentaje no deducible del concepto

Contrato	
Atributo	**Descripción**
Folio	Número del contrato
Estatus	Estado del contrato dentro del sistema
Fecha primer pago	Fecha del primer pago
Fecha último pago	Fecha para el último pago
Número de pagos	Cantidad de pagos involucrados
Periodicidad	Qué tan seguido serán los pagos

Cuenta Contable	
Atributo	**Descripción**
Empresa	Empresa dueña de la cuenta
Cuenta	Clave de la cuenta contable
Nombre	Nombre de la cuenta
Estatus	Estado de la cuenta para el sistema

Depósito	
Atributo	**Descripción**
Folio	Número del depósito en el sistema
Empresa	Número de empresa a favor de la que es el depósito
Empleado	Número del empleado que realiza el depósito
Fecha	Fecha del mismo
Banco	Banco en el que se realizó
Referencia	Referencia que se dio al movimiento
Monto	Cantidad depositada
Moneda	Moneda de la transacción

Empresa	
Atributo	**Descripción**
identificador	Número de la empresa en el sistema
Nombre	Nombre de la empresa
RFC	RFC de la empresa
Dirección	Dirección Completa
Teléfono	Teléfonos, fax
Email	Email de contacto
Logotipo	Imagen del logotipo
Estatus	Estado en el sistema

Flujo de Autorización	
Atributo	**Descripción**
Identificador	Número del flujo de autorización
Empresa	Número de la empresa
tipo de operación	Tipo de la operación a la que se aplica el flujo
Estatus	Estado que guarda el flujo en el momento actual

estatus siguiente	Estado siguiente que tendrá el flujo después del actual

Moneda	
Atributo	**Descripción**
identificador	Clave de la moneda en el sistema
descripción	Nombre de la moneda
Estatus	Estado de la moneda en el sistema

Nivel	
Atributo	**Descripción**
Identificador	Número del nivel de responsabilidad para el sistema
Empresa	Empresa a la que pertenece
Descripción	Nombre del nivel de responsabilidad

Pago	
Atributo	**Descripción**
Folio	Número de la solicitud de pago
Proveedor	Proveedor al que se hace el pago
Factura	Número de la factura del proveedor

Perfil	
Atributo	**Descripción**
Identificador	Clave del perfil de usuario
Empresa	Empresa para la que es aplicable
Descripción	Nombre del perfil

Presupuesto	
Atributo	**Descripción**
Identificador	Número de afectación al presupuesto
Empresa	Empresa afectada
Cuenta contable	Cuenta contable a la que se aplica el presupuesto afectado
Presupuestado	Presupuesto designado
Gastado	Presupuesto gastado
Comprometido	Presupuesto comprometido
Moneda	Moneda en la que se afecta
Centro de costo	Centro de costo afectado por el movimiento
Solicitud	Solicitud que afecta el presupuesto
Monto	Monto que afecta el presupuesto
Tipo	tipo de presupuesto afectado
Fecha	Fecha de afectación

Proveedor	
Atributo	**Descripción**
Identificador	Clave única de Proveedor
Nombre	Nombre
RFC	RFC
Dirección	Dirección completa
Teléfono	Teléfonos, fax
Email	Dirección de correo electrónico de contacto
Banco	Banco que usa el proveedor para cobrar
Cuenta	Cuenta bancaria del proveedor
Tipo de pago	Forma en la que se hace el pago al proveedor

Reservación	
Atributo	**Descripción**
Folio	Número de la solicitud de reservación
Proveedor	Identificador del proveedor
Fecha solicitud	Fecha de la solicitud

Solicitud	
Atributo	**Descripción**
Folio	Número de la solicitud
Tipo de operación	Tipo de solicitud
Empresa	Identificador de la empresa involucrada
Empleado	Identificador del empleado que hace la solicitud
Estatus	Estatus de la solicitud en el sistema
Moneda	Moneda en que se solicita
Moneda de pago	Moneda en la que se requiere el pago
Territorio	Territorio de aplicación
Fecha solicitud	Fecha de la solicitud
Descripción	Texto descriptivo

Tipo de Cambio	
Atributo	**Descripción**
Moneda origen	Moneda desde la que se hace la equivalencia
Moneda destino	Moneda destino de la conversión
Fecha	Fecha del tipo de cambio
tipo de cambio	Relación entre ambas monedas

Usuario	
Atributo	**Descripción**
Empresa	Empresa a la que pertenece el usuario
Identificador	Número que identifica al usuario en el sistema
Tipo de usuario	Tipo de usuario según el sistema
Nivel	Nivel de responsabilidad
Banco	Banco para pagos al usuario
Perfil	Identificador del perfil del usuario
Centro de costos	Centro de costos del usuario
Estatus	Estado para el sistema
Nombre	Nombre completo del usuario
RFC	RFC

Puesto	Puesto e la empresa
Cuenta bancaria	Número de la cuenta bancaria del usuario
Clave de acceso	Clave o password de acceso al sistema
Tarjeta corporativa	Número de la tarjeta corporativa asignada al usuario si la hay
Email	Correo electrónico
Jefe	Número de usuario del jefe inmediato
Tipo de pago	Tipo de la forma de pago preferida por el usuario

Política	
Atributo	**Descripción**
Identificador	Número o clave de la política
Nivel	Nivel de responsabilidad al que aplica
Concepto	Clave del concepto de gasto al que aplica
Tipo Operación	Clave del tipo de la operación
Empresa	Empresa involucrada
Moneda	Moneda del gasto
Territorio	Territorio del gasto
Monto Límite	Monto límite permitido según la política

Tipo de concepto	
Atributo	**Descripción**
Empresa	Empresa a la que corresponde el tipo de concepto
Descripción	Texto descriptivo del tipo de concepto
Boleto	Dice si se solicita la información relativa a un boleto de avión
Kilometraje	Establece si se solicita información de kilometraje
Precio unitario y cantidad	Se solicita información de precio unitario y cantidad?
IVA	Se solicita IVA?
TUA	Se solicita TUA?
Propina	Aplica pedir información de propina?
Días	Se solicita información de días?
Comensales	Se solicita información de comensales?
Depósito	Pedir información de depósito?

Tipo de operación	
Atributo	**Descripción**
Empresa	Número de la empresa relacionada
Tipo de operación	Identificador del tipo de operación
Descripción	Nombre del tipo de operación (para el sistema)
Nombre	Nombre del tipo de operación (para el usuario)
Estatus	Estado del tipo de operación en el sistema

Análisis de la aplicación.

En esta sección vamos a analizar la funcionalidad que se requiere por parte de la capa de aplicación para dar lugar al proceso de diseño, pero antes es necesario hacer un comentario al respecto:

Debido a que la base de datos que vamos a usar se basa en el estándar SQL, y en especial en el SQL Server de Microsoft, vamos a tratar de mantener tanta funcionalidad de la aplicación dentro de la base de datos, es decir, vamos a codificar la mayor cantidad posible de procesamiento de la aplicación como objetos de la base de datos, ya sea en forma de procedimientos almacenados, triggers o funciones; dejando la menor cantidad de proceso posible al lenguaje que servirá de interfase gráfica , que en este caso será ASP. Lo anterior debido a que si en un futuro fuera necesario cambiar la interfase gráfica o adaptar el sistema a un nuevo lenguaje o actualización del mismo, ya solo sería necesario cambiar la parte de la aplicación que corresponde específicamente a la interfase gráfica y todo lo que está en la base de datos no es necesario modificarlo. Aún en el caso de que en algún momento se quisiera migrar a otra base de datos, sería mas fácil en esta forma.

Casos de uso

Los diagramas siguientes ilustran de manera gráfica la generalidad de las operaciones que los diferentes usuarios podrán realizar en el sistema.

Captura de solicitud y envío para autorización

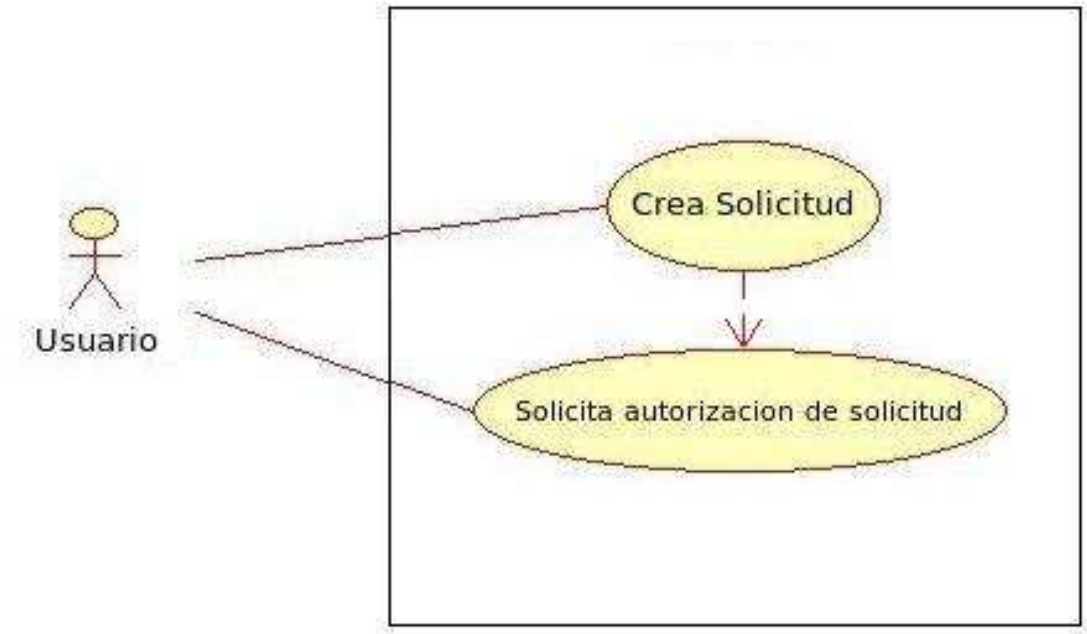


Figura 2.4. Diagrama de caso de uso de captura de solicitud y envío a autorización

Revisa y Autoriza/Rechaza solicitud

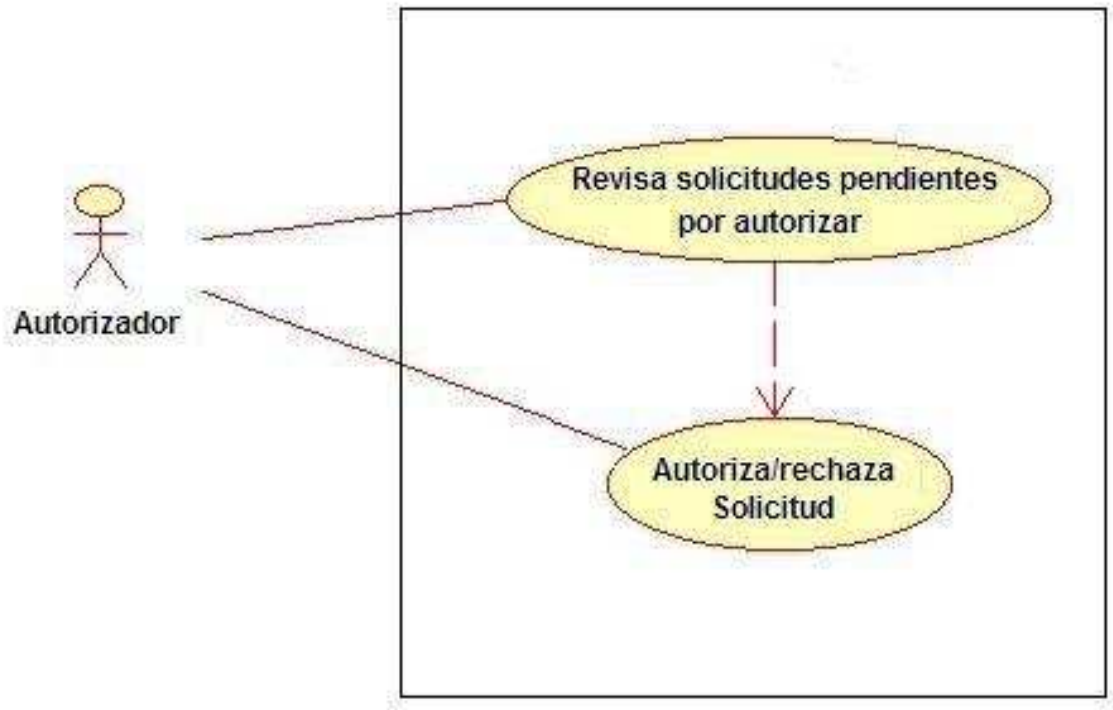


Figura 2.5. Diagrama de caso de uso de autorización/rechazo de solicitud.

Obtiene datos de contabilidad

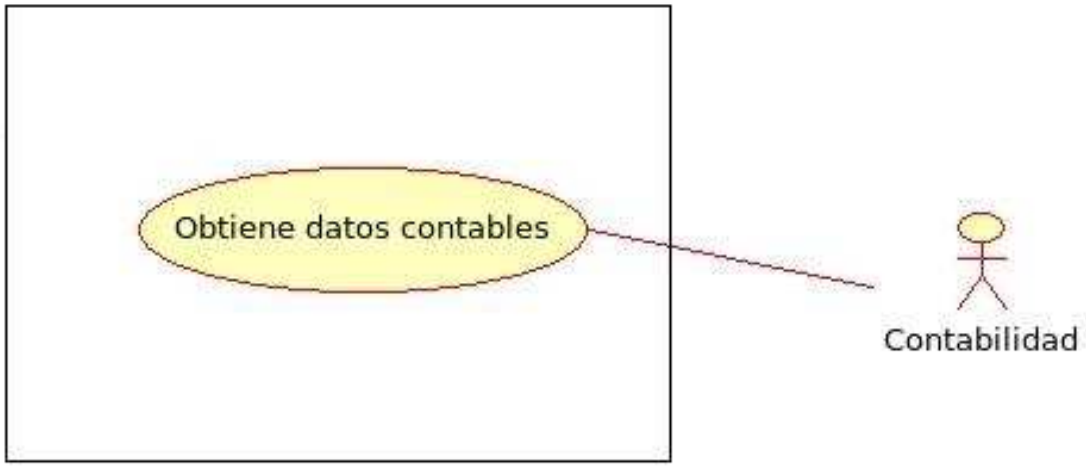


Figura 2.6. Diagrama de caso de uso de la obtención de datos para contabilidad.

Consulta de reportes

Figura 2.7. Diagrama de caso de uso de captura de solicitud.

Consulta Solicitudes

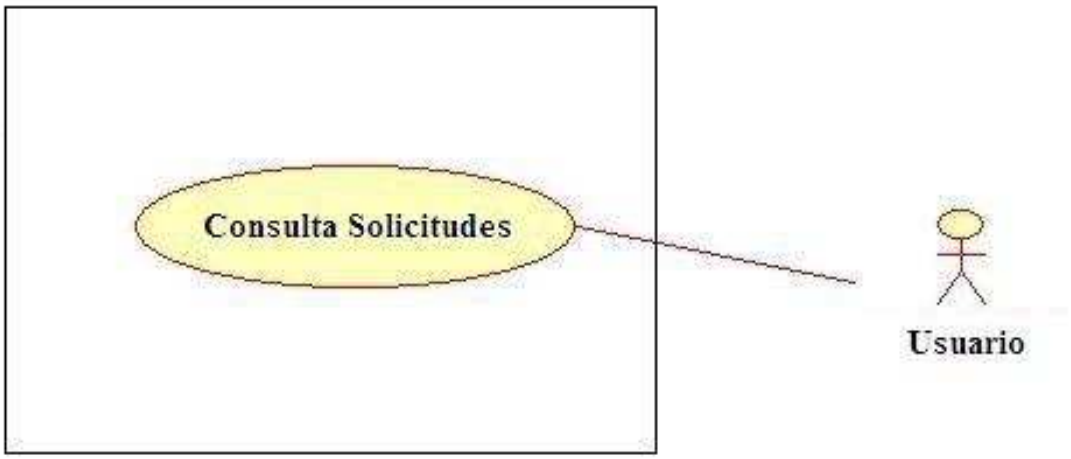


Figura 2.8. Diagrama de caso de uso de captura de solicitud.

Estructura funcional de la aplicación

Para cubrir la funcionalidad requerida por el cliente será necesario tener como ya se dijo con anterioridad, una aplicación que funcione sobre web y que por lo tanto estará formada de “páginas” web que estarán entrelazadas entre si por un sistema de “navegación” que en este caso será un menú interactivo. De tal manera, se ha dividido la funcionalidad de la aplicación en los siguientes módulos que a su vez pueden tener sub-divisiones y tienen cada uno sus respectivas interfases con el usuario que llamaremos, dada su naturaleza pantallas, o mejor dicho, páginas:

Seguridad:

- Página de acceso al sistema
- Página de inicio.

Solicitudes:

- Página de Anticipos
- Página de comprobación de anticipos
- Página de comprobación de gastos

Autorizaciones:

- Página de autorizaciones

Opciones:

- Página de cambio de clave de acceso
- Página de reporte de autorizaciones

A continuación se detalla cada una de ellas.

Módulo de seguridad

Página de acceso al sistema

La página de acceso es la encargada de validar a los usuarios del sistema y permitirles el ingreso a la página de inicio, asi como también brindarles la posibilidad de recuperar su clave de acceso (password) si lo olvidaron. Por lo tanto, las funciones de esta página son:

Validar que los datos de acceso sean correctos.

En caso de que **si** lo sean, enviar al usuario a la pantalla de inicio.

En caso de **no** serlo, debe mostrar un mensaje de advertencia, para que el usuario verifique que los datos que ingreso sean correctos y vuelva a intentar ingresar.

En caso de no tener un usuario valido, el administrador de sistema será el encargado de dar de alta uno, verificando las políticas de acceso al sistema de la empresa.

En caso de que el usuario olvide su clave, enviarla por correo electrónico a la dirección que tiene registrado ese usuario en la base de datos.

Página de Inicio.

La página de inicio es el primer lugar al que llega todo usuario después de haber pasado por el proceso de acceso o "login" al sistema.

En esta página se debe mostrar al usuario el menú con las opciones a las cuales tiene permitido acceder, según su perfil[9] (Usuario estándar, autorizador, administrador) de tal manera que los usuarios no puedan realizar acciones que no les corresponden o para las que no están autorizados.

[9] Como ya se había mencionado, cada usuario en el sistema tendrá un roll específico según las actividades que valla a desarrollar en el mismo por lo que la implementación de esta regla es a base de un sistema de perfiles en los que cada perfil tiene definidas ciertas actividades y el administrador del sistema asigna uno u otro perfil dependiendo de la función que desempeña el usuario en la empresa y específicamente en el sistema.

Módulo de solicitudes

Dado que los diferentes tipos de solicitudes que pueden realizar los usuarios comparten una cosa en común: ser solicitudes, tendrán por lo mismo, datos en común que serán tratados como un bloque común sin importar el tipo específico de solicitud de que se trate en un momento dado, entonces vamos a comenzar por definir la funcionalidad del bloque común de datos que serán alimentados al sistema, siempre que se trate de una solicitud.

Bloque de datos generales (encabezado de las solicitudes)

Para todas las solicitudes se van a requerir los siguientes datos:

- Folio. Número de solicitud asignado por el sistema para control tanto interno como informativo para el usuario.
- Moneda de la solicitud. La moneda en que se está realizando la solicitud.
- Moneda del pago. La moneda en la que el usuario quiere que se le pague el importe de la solicitud.
- Territorio. Lugar donde se realizará el gasto (nacional o extranjero)
- Descripción: Texto descriptivo del gasto

Bloque de detalles generales (Detalle de las solicitudes, parte general)

De igual forma que para el encabezado, hay datos del detalle de las solicitudes que son comunes a todas, a saber:

- Concepto de gasto. Es la categoría de gasto que pretende capturar el usuario, la cual debe seleccionar de la lista de conceptos válidos según el tipo de operación que esté realizando.
- Descripción del gasto. Es un texto descriptivo de este concepto.
- Fecha del gasto. La fecha en la que se realiza el gasto relativo al concepto.

En este bloque debe ser posible capturar tantos conceptos de gasto como lo necesite el usuario, siendo este bloque por lo tanto, iterativo.

Detalles particulares de cada tipo de gasto.

Según el tipo de operación seleccionado en el bloque de datos generales y el tipo de gasto seleccionado en el bloque de detalles generales, el sistema debe solicitar una serie de datos particulares del tipo de gasto en cuestión los cuales deben haber sido previamente configurados por el administrador siguiendo las pautas de las políticas de gastos aplicables de la empresa.

Asi por ejemplo para un boleto de avión debe solicitarse:

- Origen
- Destino
- Aerolínea
- Hora de salida

Para una comida:

- Número de comensales
- Propina

Para otro tipo de gasto:

- Precio unitario
- Unidades

Otros datos que se solicitarán dependiendo del tipo de gasto son:

- Subtotal del concepto
- IVA
- Porcentaje de IVA
- Kilometraje
- Cantidad de días
- TUA
- Total

Los datos que se puedan calcular automáticamente deberán hacerse así, por ejemplo si el usuario captura el subtotal, al momento de capturar el porcentaje de IVA se pueden calcular y desplegar el IVA y el total

Funcionalidad específica por tipo de operación.

Según el tipo de operación, existen algunas diferencias en cuanto a la información que se solicita del usuario. A continuación se detalla para cada tipo de operación, la información requerida en forma general y específica:

Anticipo: En esta página debe solicitarse lo siguiente:

- Datos generales (encabezado)
- Fecha de inicio. Cuándo comienza a aplicarse el anticipo?
- Fecha de fin. Cuándo termina la aplicación del mismo?
- Detalles generales.
- Detalles particulares de los tipos de gasto.

Comprobación de anticipo: La comprobación de anticipo es el siguiente eslabón del anticipo, es decir, cuando el usuario ya realizó el gasto y debe comprobar que parte del mismo realmente ocupó y presentar prueba de lo gastado en forma de comprobantes según las políticas de la empresa.

Al ingresar a la página de comprobación de anticipo, el usuario debe poder ver los anticipos que tiene pendientes de comprobar, es decir, que están pendientes. Esta opción deberá ser en forma de reporte tabular conteniendo la siguiente información:

- Folio del anticipo
- Descripción
- Monto
- Moneda de la operación
- Moneda del pago
- Territorio
- Fecha de la solicitud

Además deberá poder realizar las siguientes acciones:

- Asociar: Mediante esta acción el usuario debe relacionar la comprobación de su anticipo con el anticipo que va a comprobar.
- Ver detalle: Consulta del detalle del anticipo seleccionado. Dentro de esta opción debe poder realizar la impresión del detalle.

Al realizar la asociación con el anticipo, los datos de este deberán llenarse automáticamente en la cabecera de la comprobación que se está capturando, los datos que se copiarán son los que apliquen, por ejemplo, las monedas, el territorio y en la descripción se hará mención al anticipo mediante el folio del mismo.

Es decir, los datos de cabecera del la solicitud de comprobación de anticipo se llenarán a partir de su anticipo relacionado, por lo que esta operación es la primera acción que debe realizarse para continuar la captura.

Una vez lleno el encabezado debe solicitarse al usuario la captura de los detalles generales y particulares de la comprobación.

Comprobación de gastos: Esta solicitud aplica cuando el usuario ha realizado ya los gastos sin un anticipo previo, es decir, de su presupuesto personal, y requiere que se le reembolse el importe de lo gastado.

En esta página se debe solicitar la sección de datos generales y detalles generales y particulares de los tipos de gasto.

Funcionalidad adicional de las páginas de solicitudes.

Adicionalmente a la captura y validación de datos de las solicitudes, las páginas web involucradas en las solicitudes deberán proporcionar funcionalidad de soporte y validación como se detalla a continuación:

Vigilancia de las políticas de la empresa: Una de las funciones medulares del sistema, según se vió en el análisis de requerimientos era precisamente la implementación de las políticas de gastos en viajes y representación que para el caso de las solicitudes se centra en el cumplimiento y apego a una serie de montos máximos en los gastos que se configuran de acuerdo a varios parámetros, a saber:

- El tipo de solicitud
- El nivel de responsabilidad del empleado
- El territorio del gasto
- La moneda de la solicitud
- El concepto de gasto

Estas políticas deben ser evaluadas cada vez que un concepto de gasto es seleccionado.

En el caso de que la política no se cumpla el sistema deberá enviar un mensaje de advertencia para avisar al usuario de que esta rebasando el límite permitido y deberá de tomar la decisión de continuar o cancelar la operación, no impedirá guardar dicha información, ya que es responsabilidad de los autorizadores (en un paso subsecuente) el autorizar o rechazar un incumplimiento.

Distribución a centros de costo: Una vez capturada la información requerida por el sistema, se debe distribuir el gasto a los centros de costo respectivos según la configuración del centro de costos para el usuario.

Confirmación de la captura: Una vez guardada la información de detalle el usuario debe de visualizar su captura, mostrando un total de todos los conceptos y poder hacer correcciones o eliminar dicha captura.

Finalización de la captura: Una vez que el usuario haya ingresado todos los datos de la solicitud, se debe dar la opción de terminar la captura y debe pasar a una página en la que se presentarán las siguientes opciones:

1. Dejar pendiente la solicitud. Esto es, dejarla en un estado de "espera" y tener así la opción de realizar modificaciones en una sesión futura o mandarla a autorizar eventualmente.
2. Regresar al detalle. Si hay alguna corrección por hacer a algún concepto de gasto o la solicitud en general.
3. Ver el resumen de la captura. El usuario debe poder ver un reporte de su captura, es decir, exactamente lo que capturó y cómo lo verán los autorizadores cuando la envíe a autorizar.
4. Ver el flujo de autorización[10]. Aquí el usuario podrá consultar quién o quienes y en qué orden (si es mas de un autorizador) tendrán que dar su visto bueno antes de que pueda ser liberado el pago solicitado.

Si el usuario decidió dejar pendiente la solicitud (en espera), cuando entre nuevamente al sistema en una nueva sesión, se debe mostrarle la solicitud pendiente para que pueda tomarla y continuar con su tratamiento.

El usuario debe de poder elegir esta solicitud y terminarla.

[10] El flujo de autorización es la cadena de uno o varios autorizadores que deben dar su visto bueno en un orden determinado para dar por autorizada una solicitud.

Cada solicitud terminada se puede mandar a autorizar, esto deberá mandar la solicitud al flujo de autorización definido en la configuración hecha por el administrador.

Solicitud de autorización: Si el usuario, al terminar su captura, selecciona la opción de “mandar a autorizar”, la solicitud debe ser liberada al flujo de autorización predefinido para ese tipo de solicitud, es decir, se debe cambiar el estatus de la solicitud de “nueva” a “pendiente de autorizar” y enviarle aviso al primer autorizador en la cadena de autorización.

Historial de la solicitud: En esta sección o página, el usuario podrá consultar los movimientos que ha tenido una solicitud desde que fue creada hasta el estado que tiene en la actualidad.

Para cada solicitud se debe mostrar la siguiente información:

En el encabezado:

- Tipo de solicitud
- Folio
- Solicitante
- Estatus

En el detalle:

- Usuario
- Estatus al que pasó
- Fecha y hora
- Observación anotada

Autorizaciones.

El flujo de autorización es el camino por el que tiene que pasar una solicitud para ser pagada. Así, tenemos que una solicitud de anticipo tiene que pasar por la autorización del jefe inmediato y de contabilidad antes de que se le pague el dinero al empleado. O que una solicitud de comprobación de gastos tiene que ser autorizada por el responsable de presupuesto y el director antes de hacer el reembolso al empleado. Los diferentes niveles de autorización por los que tiene que pasar una solicitud antes de ser pagada es lo que se conoce como flujo de autorización.

Las autorizaciones son, junto con la realización de solicitudes electrónicas, la parte medular del SISTEMA. La pantalla de autorizaciones permite a los autorizadores o a los responsables de todos los niveles de autorización autorizar o rechazar las solicitudes electrónicas de los empleados.

Cuando un se autoriza una solicitud, el SISTEMA envía automáticamente un correo a los autorizadores de los siguientes niveles. Esto se repite hasta que la solicitud ha sido autorizada por el último nivel definido en el flujo.

Cuando una solicitud es rechazada, sin importar el nivel en que esté, se le envía un correo electrónico al solicitante avisándole que su solicitud ha sido truncada en el flujo. Comúnmente, el solicitante modifica esta solicitud y la manda a autorizar nuevamente. Cuando una solicitud es rechazada, ésta comienza a participar en el flujo desde el primer nivel, ya que la solicitud ha sido modificada y ya no corresponde a lo que se autorizó con antelación.

Página de autorización.

En la página de autorización se debe presentar al usuario el reporte de las solicitudes pendientes de autorizar con los siguientes datos generales:

Número de folio de la solicitud
Beneficiario
Monto
Fecha de la solicitud

Para cada solicitud debe ser posible visualizar sus detalles generales, así como a que centro de costos se está cargando el gasto. Así mismo, en esta página se debe dar al usuario la posibilidad de autorizar o rechazar la solicitud en cuestión, así como de poder alimentar un comentario o motivo de rechazo.

Opciones.

Pantalla de cambio de clave de acceso.

En esta pantalla se debe presentar al usuario un formulario sencillo que solicite el número de usuario, la clave actual de acceso para validarla en la tabla de usuario, y la clave nueva a la que quiere cambiar, esta por duplicado para evitar errores de tecleo.

También los botones de “Aceptar”, “Cancelar” y “limpiar forma”

Reporte de autorizaciones.

Esta pantalla debe ser un sencillo reporte que solicite como parámetros, el tipo de operación, moneda de la solicitud, fecha o rango de fechas, y el estatus de las solicitudes que se quieren visualizar.

Los datos a presentar en el reporte son: el folio de la solicitud, fecha, descripción, estatus de tratamiento, estatus de autorización, el monto y la moneda.

Debe dar opción al usuario a consultar el historial, y el flujo de autorización si es que está pendiente alguna, así como poder ver el detalle completo de la solicitud seleccionada.

2.3. Diseño de la base de datos

2.3.1. Diagramas Entidad-Relación

Diagrama del módulo de seguridad

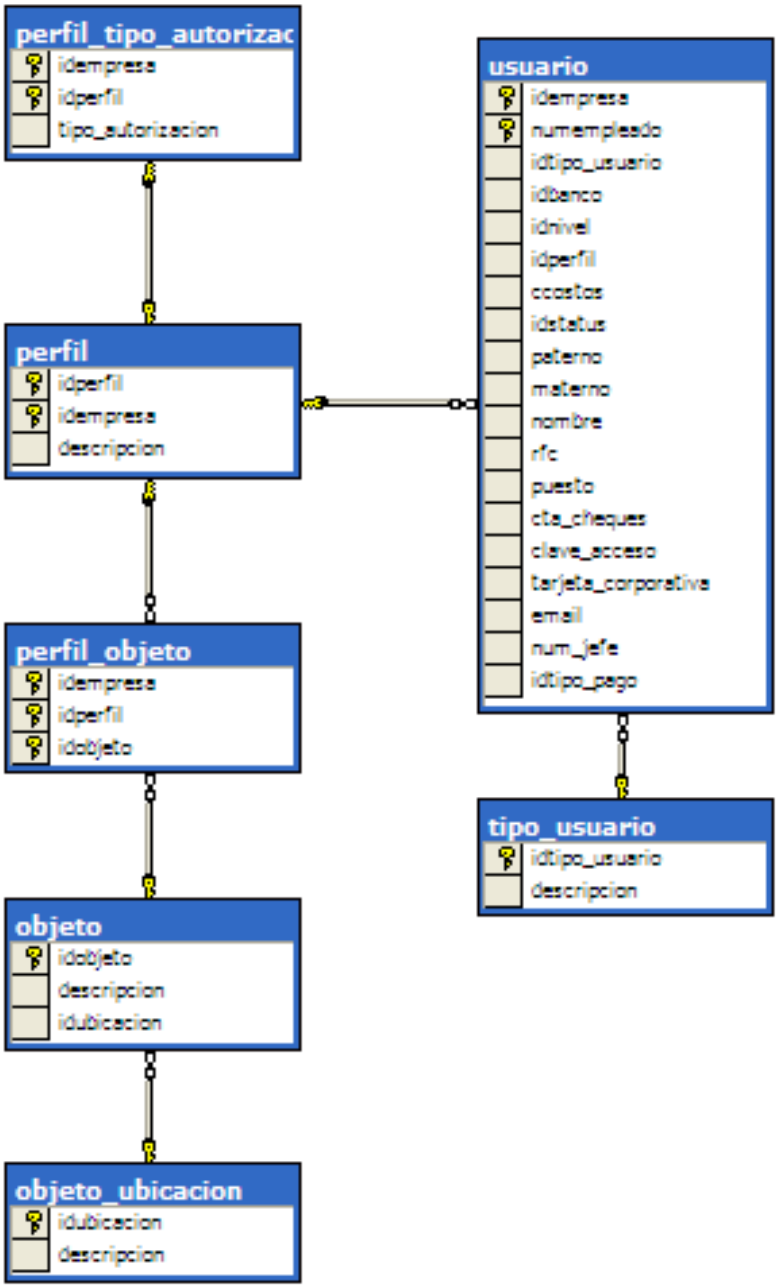


Figura 2.9. Diagrama E-R del módulo de seguridad.

Diagrama E-R del módulo de solicitudes

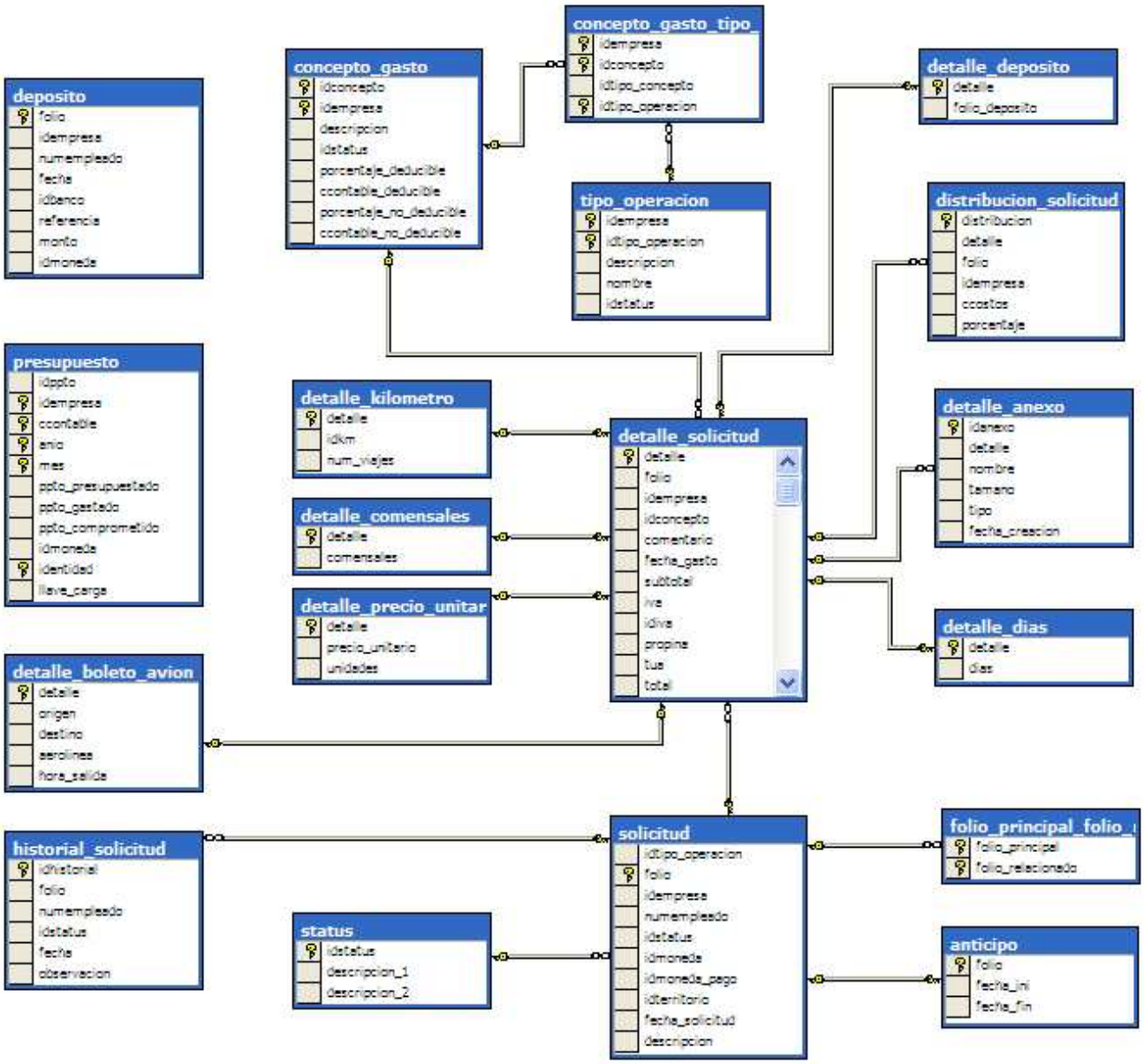


Figura 2.10. Diagrama E-R del módulo de solicitudes

Diagrama Entidad-Relación del módulo de autorización

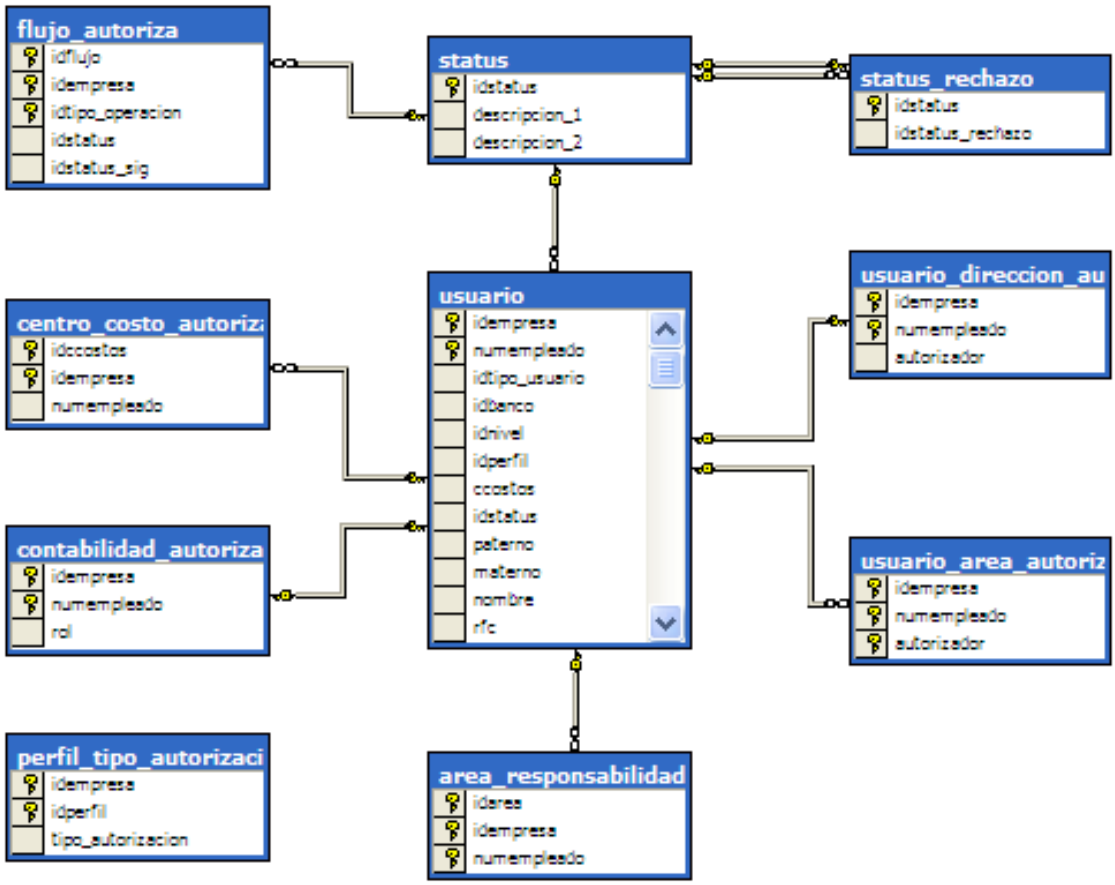


Figura 2.11. Diagrama E-R del módulo de autorización

2.3.2. Diagramas de flujo de datos

Los siguientes diagramas tienen que ver con la forma en que funciona la aplicación y se incluyen aquí porque de ellos dependen objetos de la base de datos como procedimientos almacenados, funciones y triggers.

Pantalla de acceso al sistema

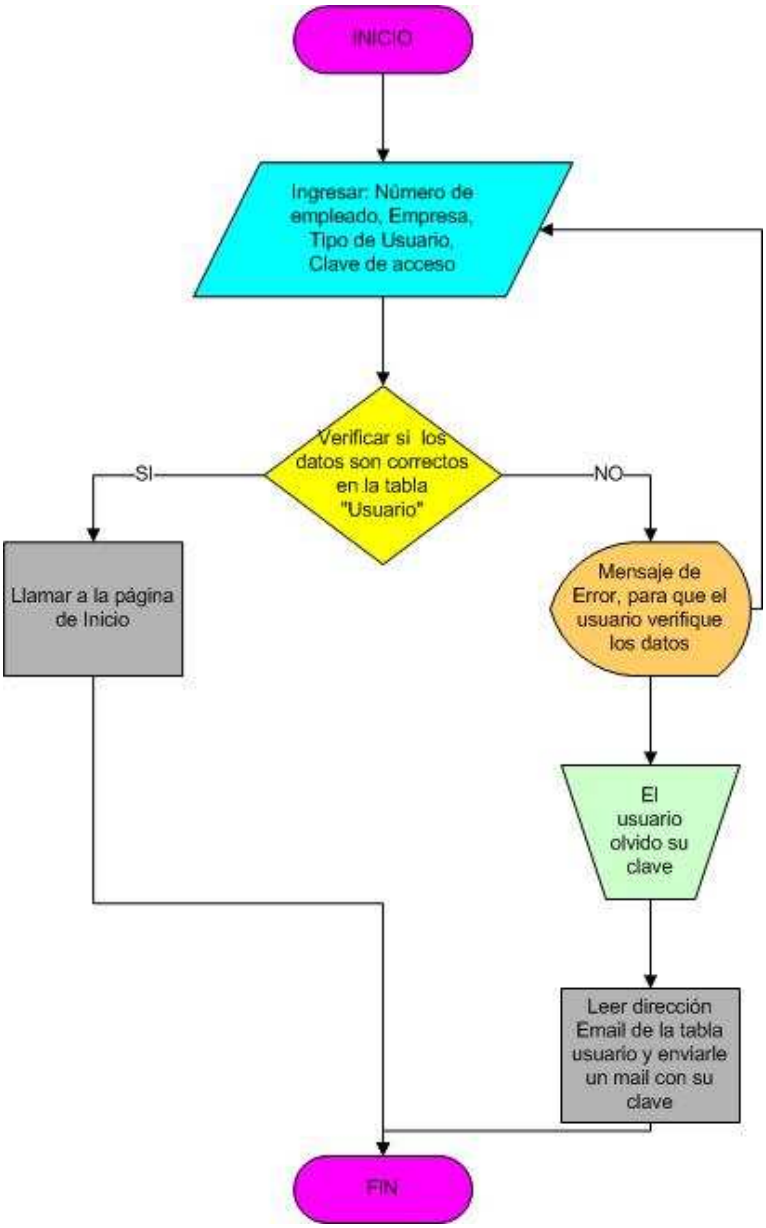


Figura 2.12. Diagrama de flujo de la pantalla de acceso al sistema

Pantalla de inicio

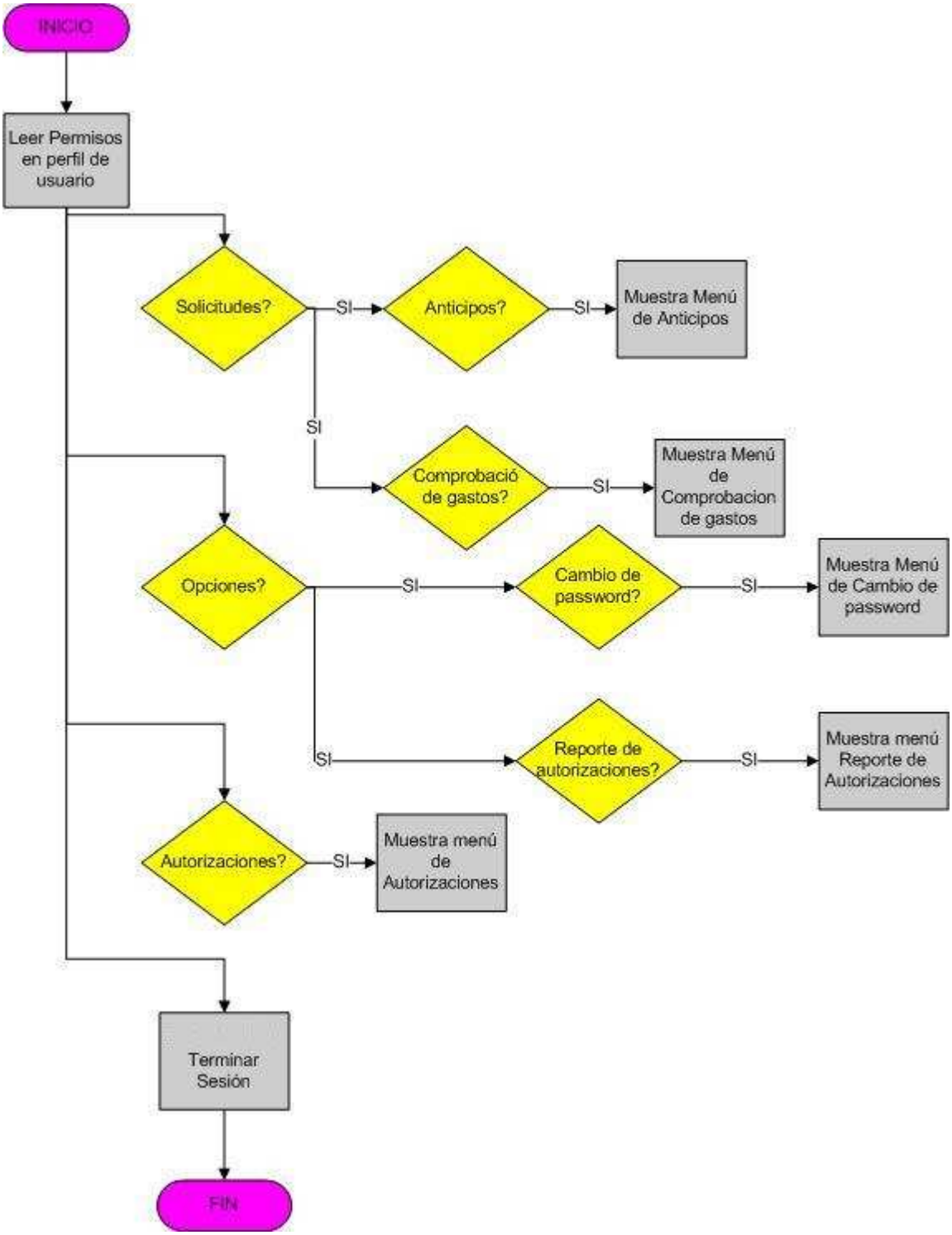


Figura 2.13. Diagrama de flujo de la pantalla de inicio.

Pantalla de solicitud de anticipos.

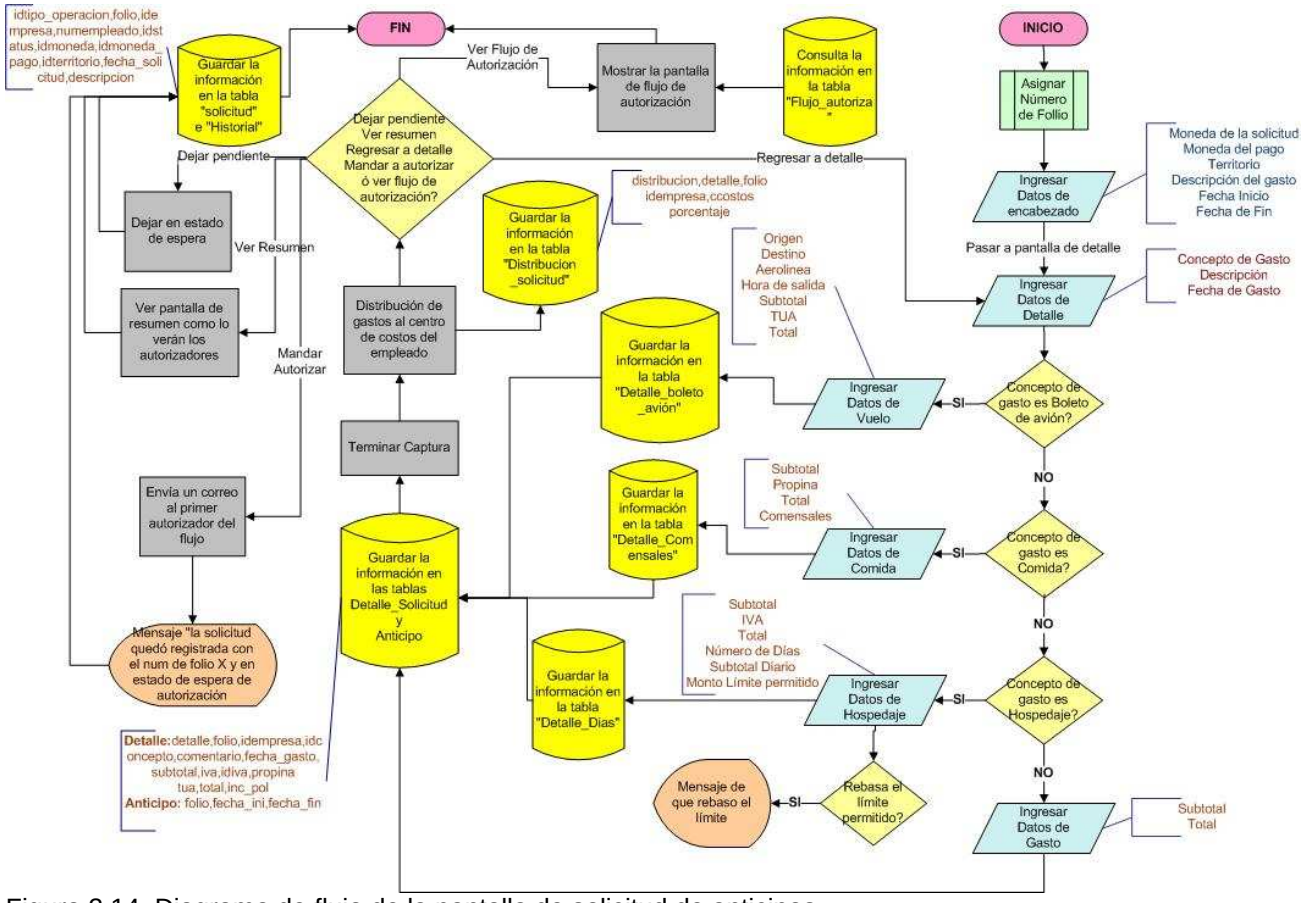


Figura 2.14. Diagrama de flujo de la pantalla de solicitud de anticipos.

Pantalla de comprobación de anticipos (cabecera)

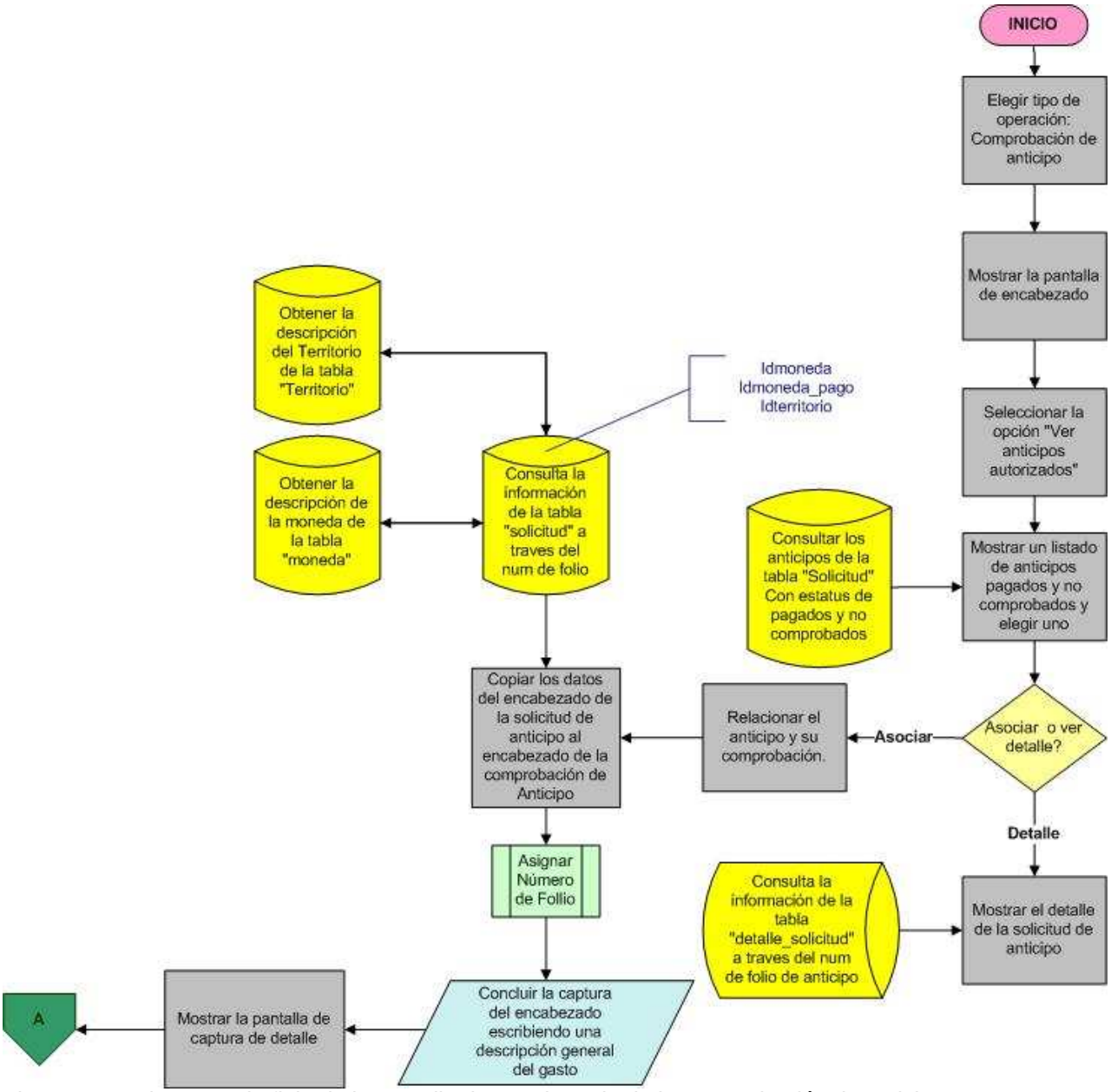


Figura 2.15. Diagrama de flujo de la pantalla de encabezado de la comprobación de anticipos.

Pantalla de comprobación de gastos (cabecera)

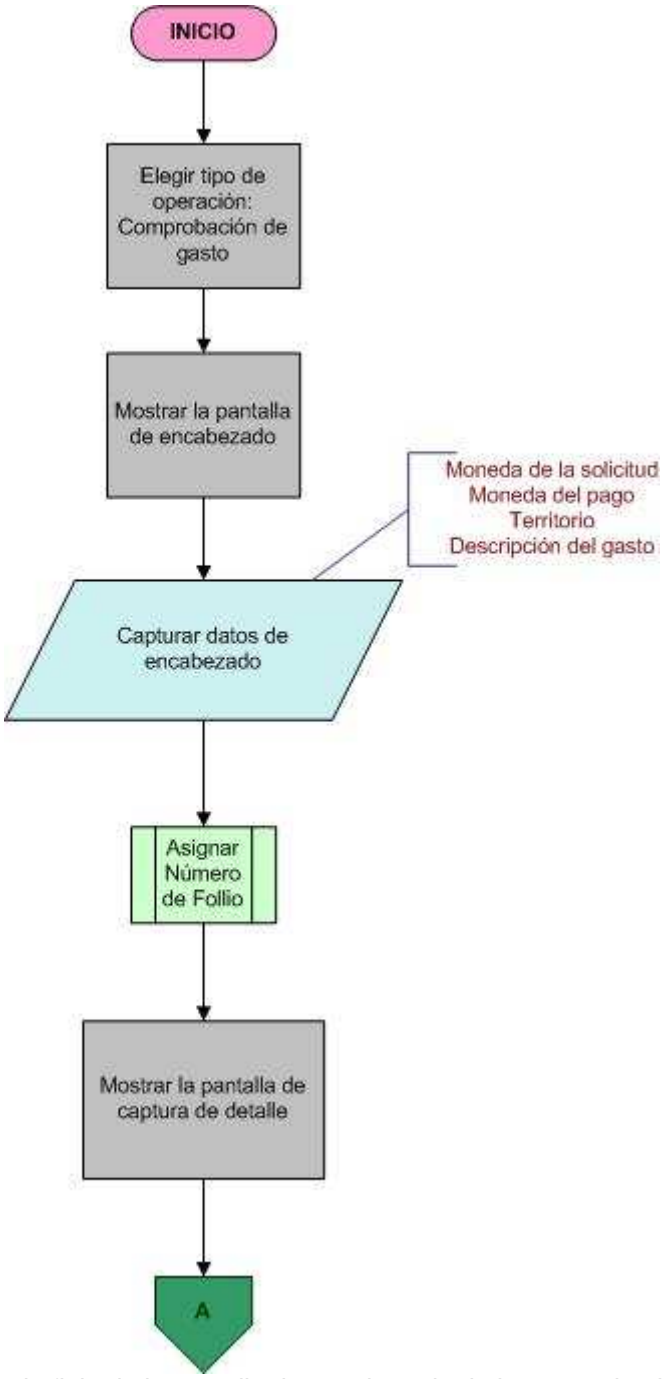


Figura 2.16. Diagrama de flujo de la pantalla de encabezado de la comprobación de gastos.

Pantalla de detalle para comprobación de anticipos y gastos.

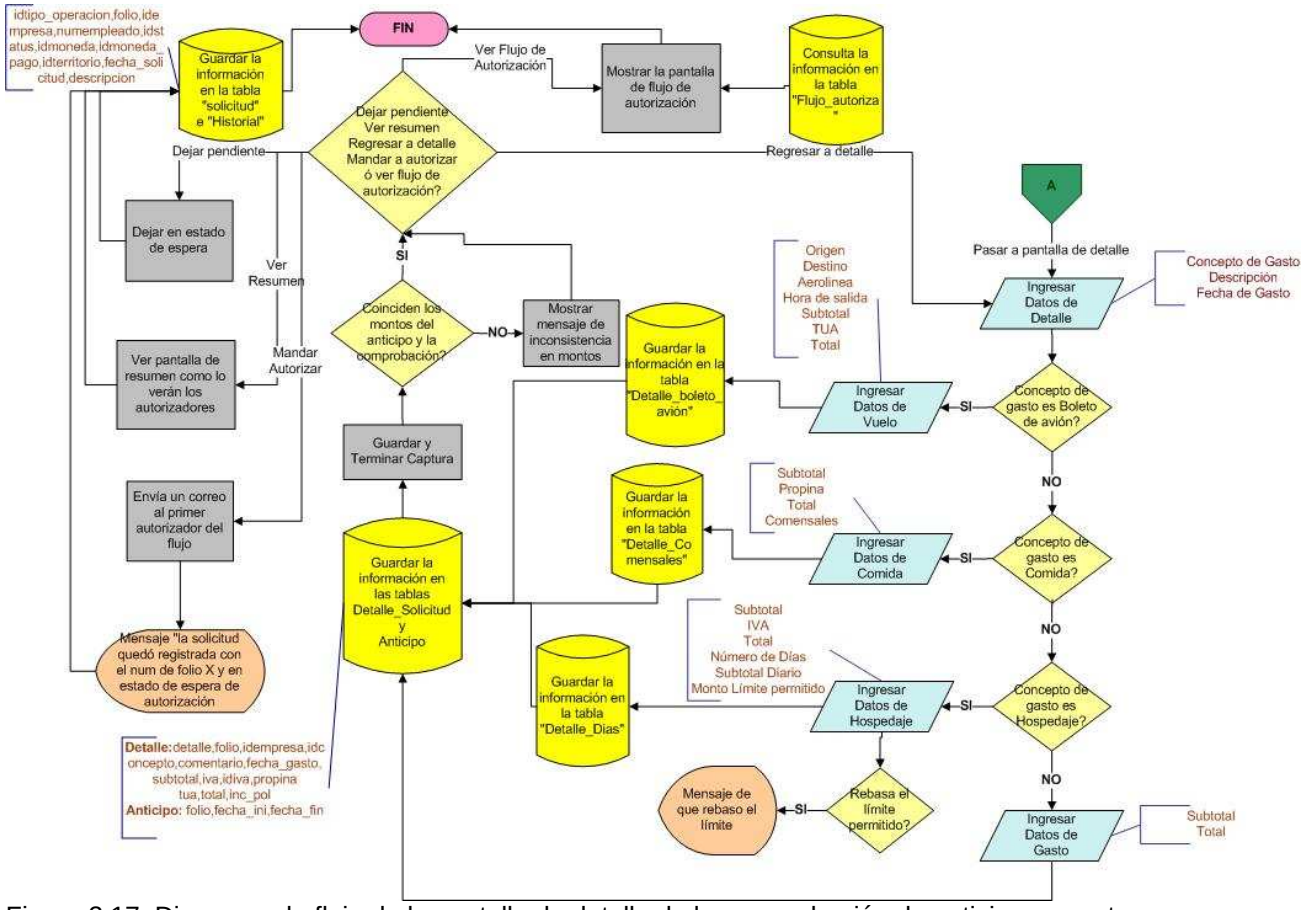


Figura 2.17. Diagrama de flujo de la pantalla de detalle de la comprobación de anticipos y gastos.

Pantalla de historial de solicitudes.

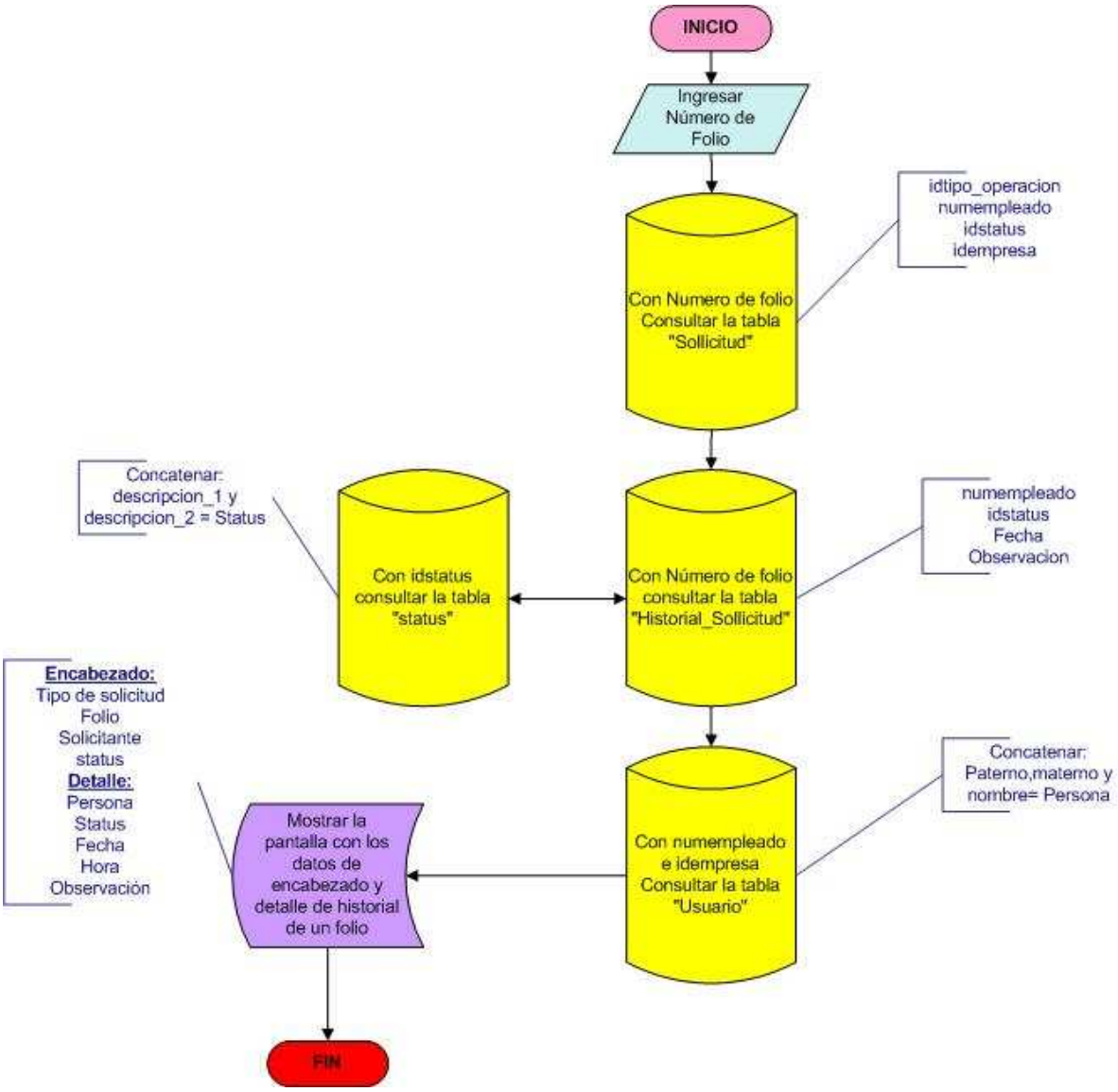


Figura 2.18. Diagrama de flujo de la pantalla de historial de solicitudes.

Capítulo III. Implementación de la base de datos del sistema de control de gastos en viajes y representación.

Una vez completa la fase de diseño se puede construir el Script SQL con el que se definen los objetos de la base de datos.

Cabe mencionar que este capítulo básicamente tendría que contener el código SQL de toda la base de datos, el cual es en si la culminación del análisis y el diseño de la base de datos y el producto principal de este trabajo, sin embargo, ya que se trata de un documento muy extenso, resulta impráctico alojarlo en este trabajo escrito de manera íntegra por lo que se ha optado por incluir sólo algunos ejemplos representativos de los principales objetos de la base de datos y si el lector está interesado en conocer el código completo se puede referir al anexo 1 (script de la base de datos) contenido en el CD que acompaña este reporte.

Acerca de la normalización de la base de datos.

Todas las tablas se han construido normalizándolas hasta la 4ª forma normal, entendiendo por esto que cumplen con lo siguiente:

- Todos los atributos que no forman parte de la llave primaria dependen completa y exclusivamente de la llave primaria (1ª, 2ª y 3ª formas normales)
- No existen múltiples atributos de la misma naturaleza en una misma tupla (4ª forma normal), excepto en los casos en los que la multiplicidad alcanza un número finito, esto último atendiendo a razones de practicidad[1]

[1] Por ejemplo si en los datos de la empresa se manejaran dos teléfonos (telefono1 y telefono2) sería porque solo se van a manejar dos teléfonos como máximo para cualquier empresa, ya que de lo contrario sería necesario crear una tabla de empresa-teléfono y por lo mismo una pantalla de mantenimiento, lo cual resulta impráctico para nuestra aplicación.

Creación de la base de datos

El siguiente fragmento de código crea la base de datos de nombre “Travex” en la ruta C:\Db\TRAVEX\ del disco duro con un tamaño de 20 Mb (es u tamaño para efectos de demostración) con crecimientos del 10%, asi como el log de transacciones con un tamaño de 50 Mb con un crecimiento también del 10%

```
CREATE DATABASE [TRAVEX]  ON (NAME = N'TRAVEX_Data', FILENAME = N'C:\Db\TRAVEX\TRAVEX_Data.MDF' , SIZE = 20, FILEGROWTH = 10%) LOG ON (NAME = N'TRAVEX_Log', FILENAME = N'C:\Db\TRAVEX\TRAVEX_Log.LDF' , SIZE = 50, FILEGROWTH = 10%)
 COLLATE SQL_Latin1_General_CP1_CI_AS
```

Creación de las tablas

Aquí se crea la tabla de solicitudes y abajo su llave primaria:

```
CREATE TABLE [dbo].[solicitud] (
	[idtipo_operacion] [int] NOT NULL ,
	[folio] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[numempleado] [int] NOT NULL ,
	[idstatus] [char] (2) NOT NULL ,
	[idmoneda] [int] NOT NULL ,
	[idmoneda_pago] [int] NOT NULL ,
	[idterritorio] [int] NOT NULL ,
	[fecha_solicitud] [datetime] NOT NULL ,
	[descripcion] [varchar] (250) NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[solicitud] WITH NOCHECK ADD
	CONSTRAINT [PK_comprobacion_gasto] PRIMARY KEY  CLUSTERED
	(
		[folio]
	)  ON [PRIMARY]
GO
```

Este otro fragmento de código crea la tabla de detalle de las solicitudes y abajo su llave primaria:

```
CREATE TABLE [dbo].[detalle_solicitud] (
	[detalle] [int] NOT NULL ,
	[folio] [int] NOT NULL ,
	[idempresa] [int] NULL ,
	[idconcepto] [int] NULL ,
	[comentario] [varchar] (250) NULL ,
	[fecha_gasto] [datetime] NULL ,
	[subtotal] [money] NULL ,
	[iva] [money] NULL ,
	[idiva] [char] (3) NULL ,
	[propina] [money] NULL ,
	[tua] [money] NULL ,
	[total] [money] NULL ,
	[tarjeta_corporativa] [tinyint] NULL ,
	[inc_pol] [tinyint] NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_solicitud] WITH NOCHECK ADD
	CONSTRAINT [PK_detalle_solicitud] PRIMARY KEY  CLUSTERED
	(
		[detalle]
	)  ON [PRIMARY]
GO
```

Integridad referencial

En esta sección se define la manera en que se relacionan las tablas entre si, como ejemplo se incluye la definición de la tabla de solicitudes.

```
ALTER TABLE [dbo].[solicitud] ADD
	CONSTRAINT [FK_solicitud_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	),
	CONSTRAINT [FK_solicitud_moneda] FOREIGN KEY
	(
		[idmoneda]
	) REFERENCES [dbo].[moneda] (
		[idmoneda]
	),
	CONSTRAINT [FK_solicitud_moneda1] FOREIGN KEY
	(
		[idmoneda_pago]
	) REFERENCES [dbo].[moneda] (
		[idmoneda]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_solicitud_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	),
	CONSTRAINT [FK_solicitud_territorio] FOREIGN KEY
	(
		[idterritorio]
	) REFERENCES [dbo].[territorio] (
		[idterritorio]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_solicitud_usuario] FOREIGN KEY
	(
		[idempresa],
		[numempleado]
	) REFERENCES [dbo].[usuario] (
		[idempresa],
		[numempleado]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO
```

Procedimientos almacenados.

En este apartado se ejemplifica la creación de los procedimientos o subrutinas que se se usan en la base de datos para la manipulación de los datos. Para ilustrar esto, se incluye el procedimiento para crear/actualizar/eliminar registros de detalle de solicitudes.

```
/*
Nombre:		sp_detalle_solicitud_inserta_modifica

Funcionalidad:
Inserta, modifica o elimina un detalle de solicitud.

Entradas:
Identificador del registro del detalle de la solicitud
Empresa en la que se trabaja
Concepto de gasto
Comentario acerca del detalle
Fecha de gasto
Subtotal del detalle
IVA
Identificador del IVA utilizado
Propina
TUA
total
tarjeta_corporativa
incumplimiento de politicas
Opcion de accion
	(i = inserta)
	(u = actualiza)
	(d = elimina)
Variable commit

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta, actualiza o elimina

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
Termina
*/


CREATE procedure [dbo].sp_detalle_solicitud_inserta_modifica
(
/* se declaran los parámetros que se le van a pasar al procedimiento */
	@id int,
	@idempresa int,
	@idconcepto int,
	@comentario varchar(250),
	@fecha_gasto char(11),
	@subtotal money,
	@iva money,
	@idiva char(3),
	@propina money,
	@tua money,
	@total money,
	@tarjeta_corporativa tinyint,
	@inc_pol tinyint,

	@proviene char(1),
	@commit tinyint = 1
```

```
)
as
begin
	declare
	@detalle int

	begin transaction trans

	if @proviene = 'i'
	begin
		set @detalle = (select [dbo].fn_detalle_obtiene_folio_consecutivo())

		set dateformat dmy insert into detalle_solicitud
		(
		detalle,
		folio,
		idempresa,
		idconcepto,
		comentario,
		fecha_gasto,
		subtotal,
		iva,
		idiva,
		propina,
		tua,
		total,
		tarjeta_corporativa,
		inc_pol
		)
		values(
		@detalle,
		@id,
		@idempresa,
		@idconcepto,
		upper(left(@comentario,250)),
		[dbo].sp_convierte_texto_a_fecha(@fecha_gasto),
		@subtotal,
		@iva,
		@idiva,
		@propina,
		@tua,
		@total,
		@tarjeta_corporativa,
		@inc_pol
		)

		select @detalle as detalle

	end

	if @proviene = 'u'
	begin
		set dateformat dmy
		update detalle_solicitud
		set
		comentario = upper(left(@comentario,250)),
		fecha_gasto = [dbo].sp_convierte_texto_a_fecha(@fecha_gasto),
		subtotal = @subtotal,
		iva = @iva,
		idiva = @idiva,
		propina = @propina,
		tua = @tua,
		total = @total,
		tarjeta_corporativa = @tarjeta_corporativa,
```

```
                inc_pol = @inc_pol
                where detalle = @id
        end

        if @proviene = 'd'
        begin
                delete from detalle_solicitud where detalle = @id
        end

        if @commit = 1
        begin
                commit transaction trans
                print 'Transaction commited'

        end

        if @commit <> 1
        begin
                rollback transaction trans
                print 'Transaction rolled-back'
        end

end
```

Funciones.

Para ejemplificar las funciones de la base de datos se agrega el código de la función que valida si un usuario que está intentando accesar a la aplicación existe en la base de datos y tiene un password válido.

```
/*
Inicia
Nombre:
fn_autentica_login



Funcionalidad:
Verifica la existencia de un usuario y su password alsolicitar acceso a la aplicación

Entradas:
Número de la empresa a la que el usuario se quiere firmar
Número de empleado que pretende firmarse a la aplicación
Tipo de usuario que quiere entrar
        1 = interno
        2 = externo
Clave de acceso que el usuario provee

Proceso:
Toma los datos de entrada y los busca en la BD para verificar si el usuario en realidad existe

Salida:
1 = el usuario existe
0 = el usuario no existe

Ejemplo:
select [dbo].fn_autentica_login(1,123,1,'123')
```

```
*/

CREATE function [dbo].fn_autentica_login
(
	@idempresa int,
	@numempleado int,
	@idtipo_usuario int,
	@pwd varchar(10)
)
returns tinyint
as
begin
	declare
	@existe tinyint

	/*
	Verifica que los datos de entrada existan en la BD
	*/
	if exists(
		select
		clave_acceso
		from usuario
		where	idempresa = @idempresa and
			numempleado = @numempleado and
			idtipo_usuario = @idtipo_usuario and
			[dbo].user_pwd(rtrim(@pwd),1) = clave_acceso
	)
	/*
	Si existe
	*/
	begin
		set @existe = 1
	end
	else
	/*
	No existe
	*/
	begin
		set @existe = 0
	end

	return(@existe)

end
```

Triggers.

Como ejemplo de uno de los triggers de la base de datos, se incluye el código del trigger que elimina el presupuesto asignadoa un detalle de solicitud cuando este es borrado. En este caso el código se dispara ante el evento “borrar” de la tabla “ppto_detalle” cuando está confirmado el borrado del registro.

```
/*
Nombre:
tr_ppto_detalle_elimina

Funcionalidad:
Cuando un detalle de solicitud se elimina, se elimina el presupuesto gastado o comprometido para ese detalle.

Proceso:
Se actualizan los registros comprometidos o gastados de un detalle.

Termina
*/

CREATE TRIGGER [dbo].tr_ppto_detalle_elimina
ON [dbo].ppto_detalle
after delete
as
	declare
	@monto money,
	@tipo char(1),
	@c_idreg int,
	@c_idppto int,
	@c_detalle int,
	@c_tipo char(1),
	@c_monto money

	declare ppto_detalle cursor for
		select idreg,idppto,detalle,tipo,monto from deleted

		open ppto_detalle

		fetch next from ppto_detalle
			into
				@c_idreg,
				@c_idppto,
				@c_detalle,
				@c_tipo,
				@c_monto

		while @@fetch_status = 0
			begin

				if @c_tipo = 'g'
				begin
					update presupuesto
					set
					ppto_gastado = isnull(ppto_gastado,0) - @c_monto
					where idppto = @c_idppto

				end

				if @c_tipo = 'c'
				begin
```

```
                                        update presupuesto
                                        set
                                        ppto_comprometido = isnull(ppto_comprometido,0) - @c_monto
                                        where idppto = @c_idppto

                                end

                                fetch next from ppto_detalle
                                into
                                        @c_idreg,
                                        @c_idppto,
                                        @c_detalle,
                                        @c_tipo,
                                        @c_monto
                        end

        close           ppto_detalle
        deallocate      ppto_detalle

GO
```

Capítulo IV. Interfase de usuario para gestionar la base de datos de control de gastos en viajes y representación.

4.1 Implementación de la aplicación.

En la sección de diseño de la base de datos se especifica la funcionalidad de cada pantalla de la interfase de usuario por lo que en esta sección vamos a tomar esa información como entrada y a desarrollar el diseño de las pantallas (o páginas web) que van a manejar la navegación del usuario y a llamar a los objetos de la base de datos que son los que van a leer y modificar los datos en las tablas. El código ASP y Java Script que corresponde a dichas pantallas se puede consultar en el anexo 2. Código ASP para gestionar la base de datos (en el CD).

Pantalla de acceso.

Figura 4.1. Pantalla de acceso al sistema

Pantalla de inicio.

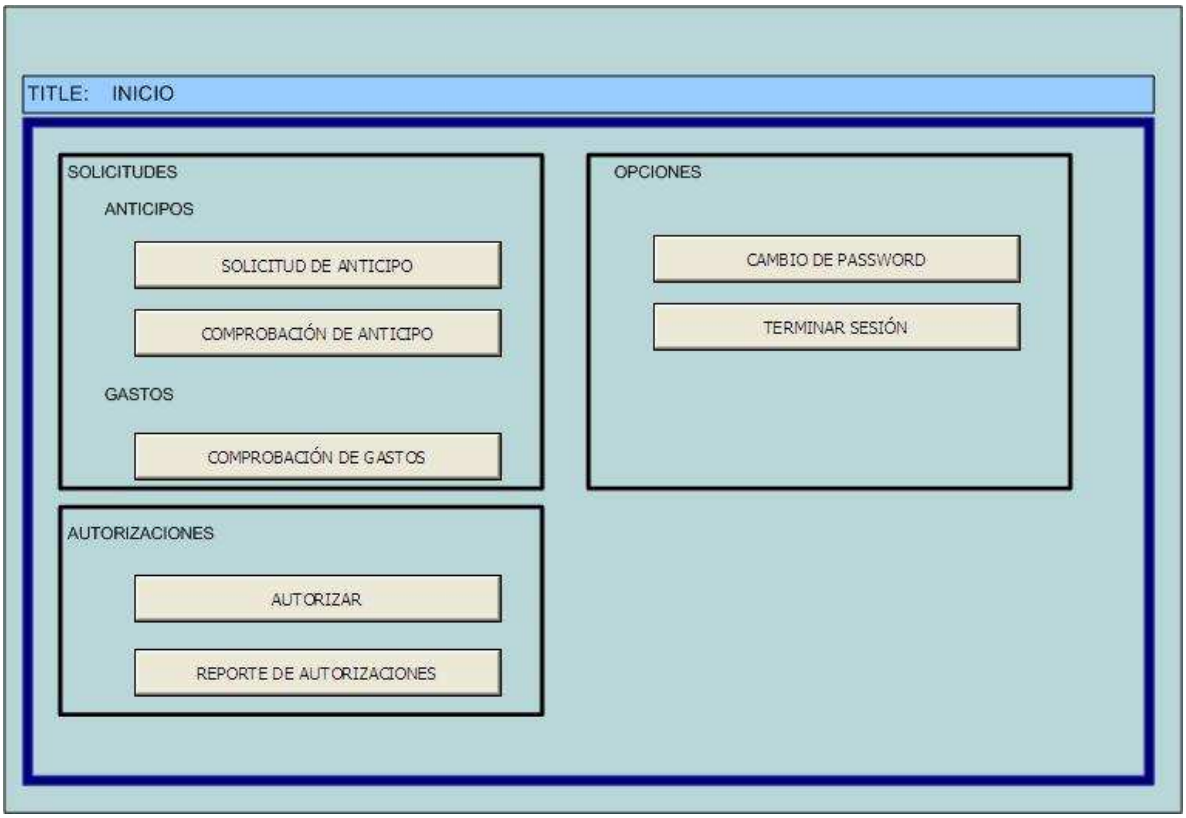


Figura 4.2. Pantalla de inicio

Esta es la vista previa de la pantalla inicial que verán los usuarios al ingresar. Desde aquí se pueden realizar todas las funciones que el usuario tenga autorizadas en su perfil.

Pantallas de Anticipos.

Cabecera.

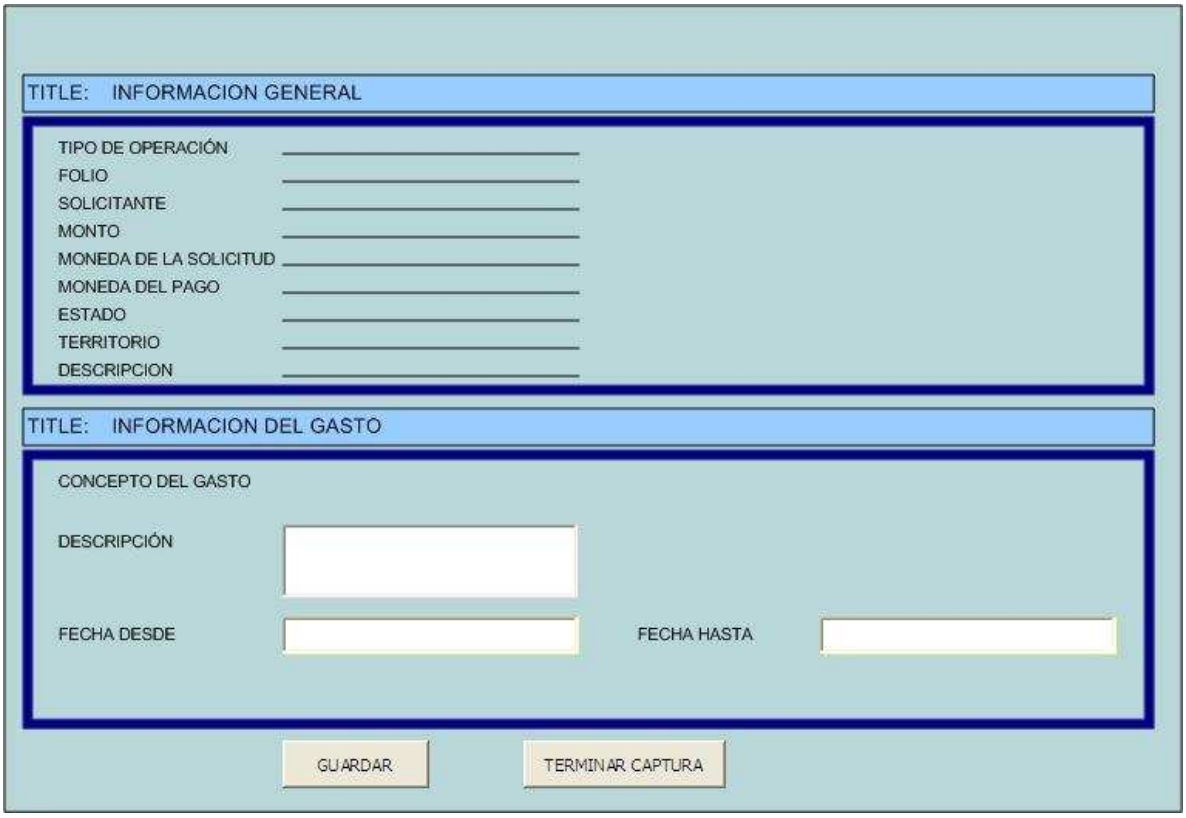


Figura 4.3. Pantalla de anticipos (datos de cabecera)

En esta pantalla se dan de alta los datos de cabecera de las solicitudes de anticipos.

Detalle de gastos de hotel

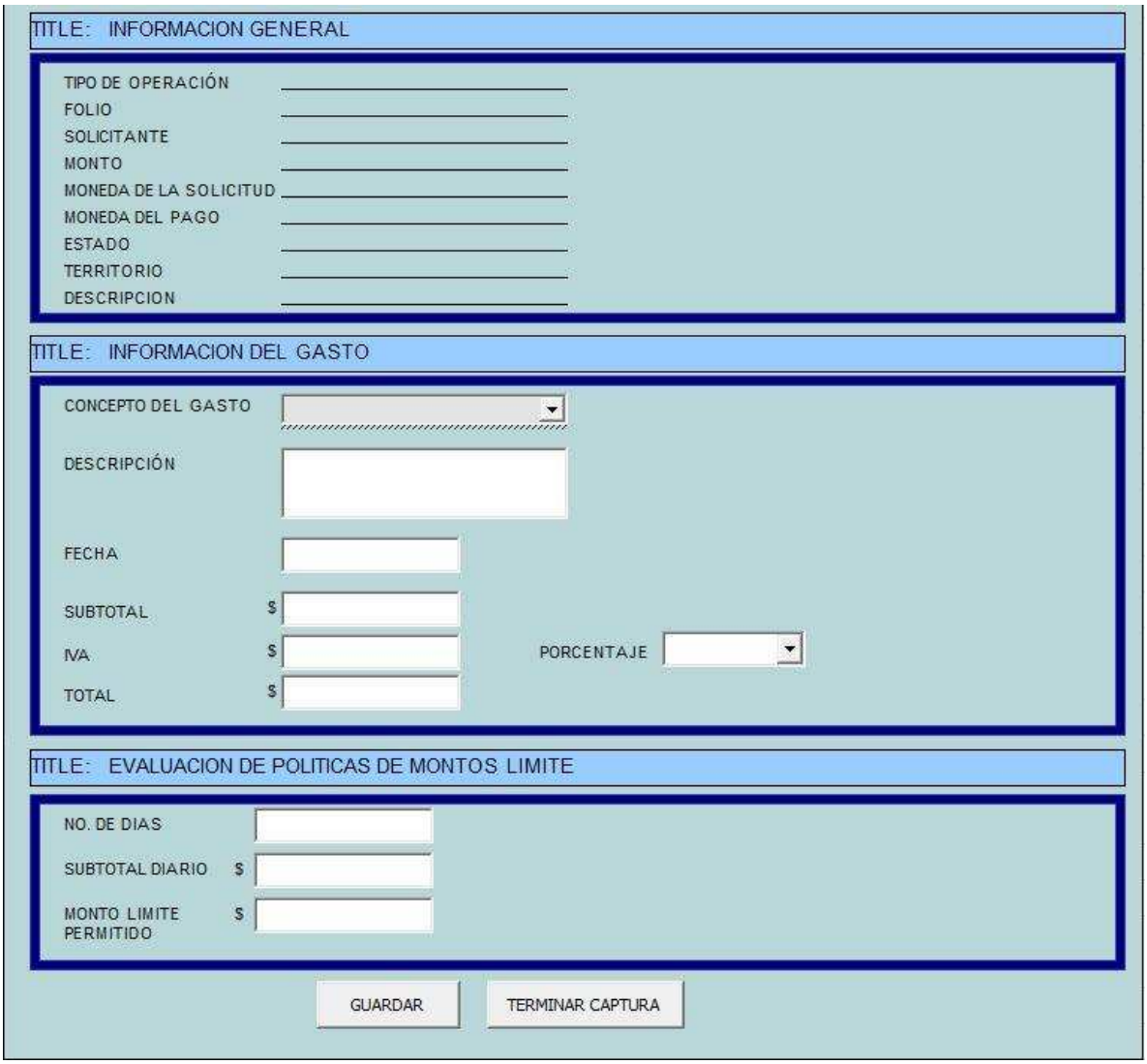


Figura 4.4. Pantalla de anticipos (datos de hotel)

Si la solicitud tiene algun concepto de gasto relacionado con reservaciones de hotel, en esta pantalla es donde se alimentan los datos pertinentes.

Detalle de gastos de comida

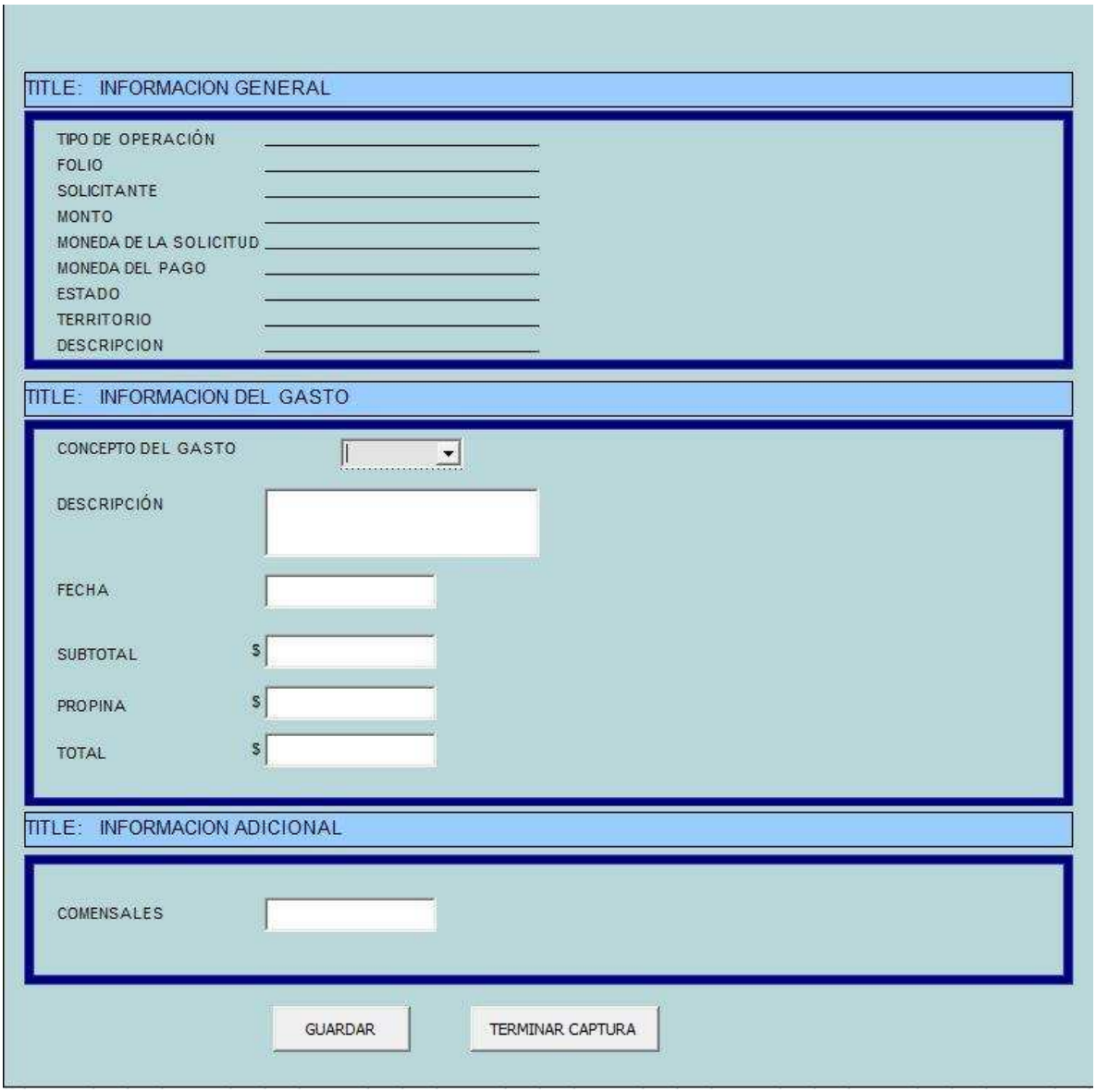


Figura 4.5. Pantalla de anticipos (datos de comidas)

Cuando el tipo de gasto es una comida, en esta pantalla se dan de alta los datos de la misma.

Detalle de gastos de avión

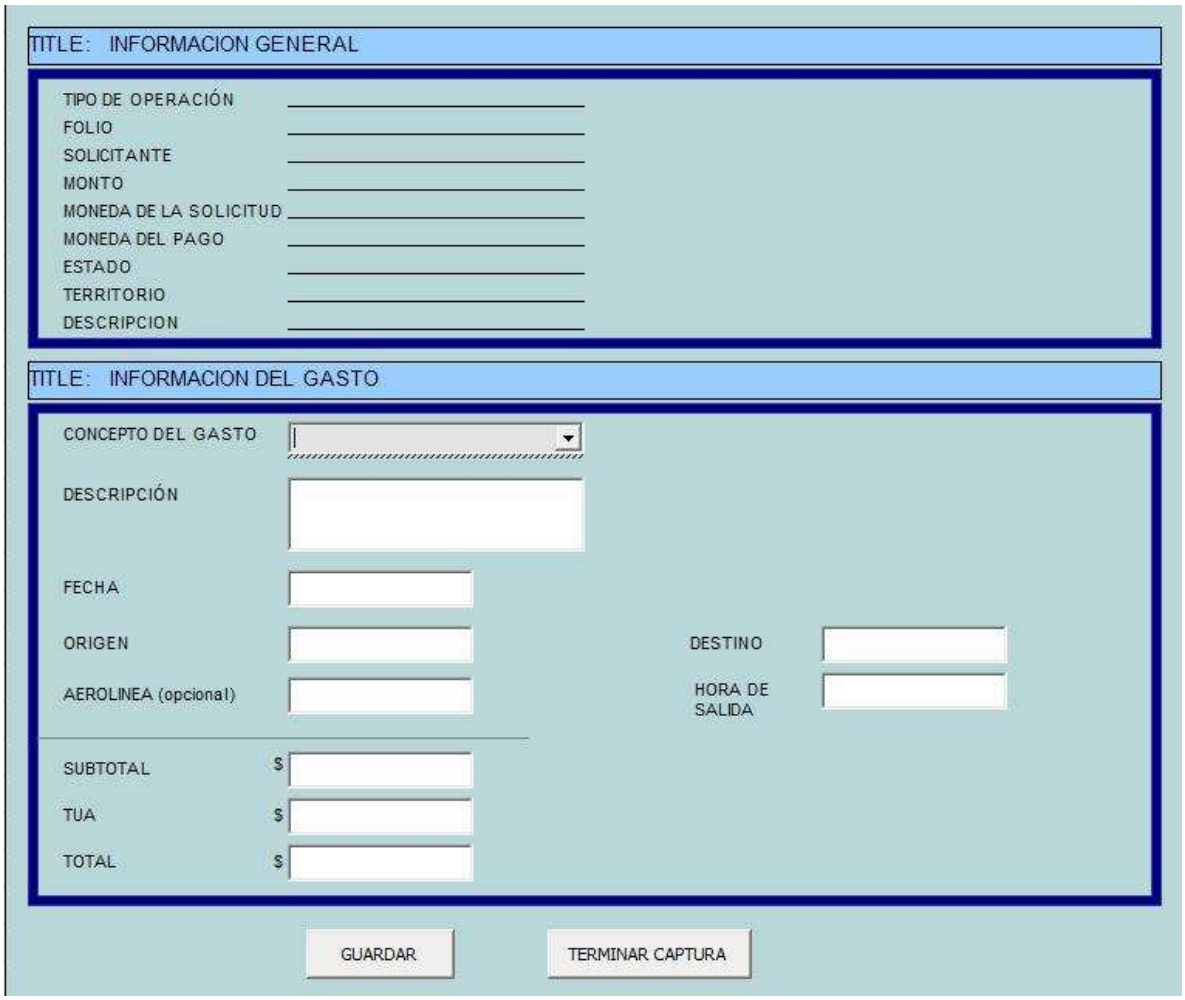


Figura 4.6. Pantalla de anticipos (datos de boletos de avión)

En esta pantalla se dan de alta los datos de los boletos de avión cuando asi se requiera.

Pantallas de Comprobación de anticipos.

Pantalla principal

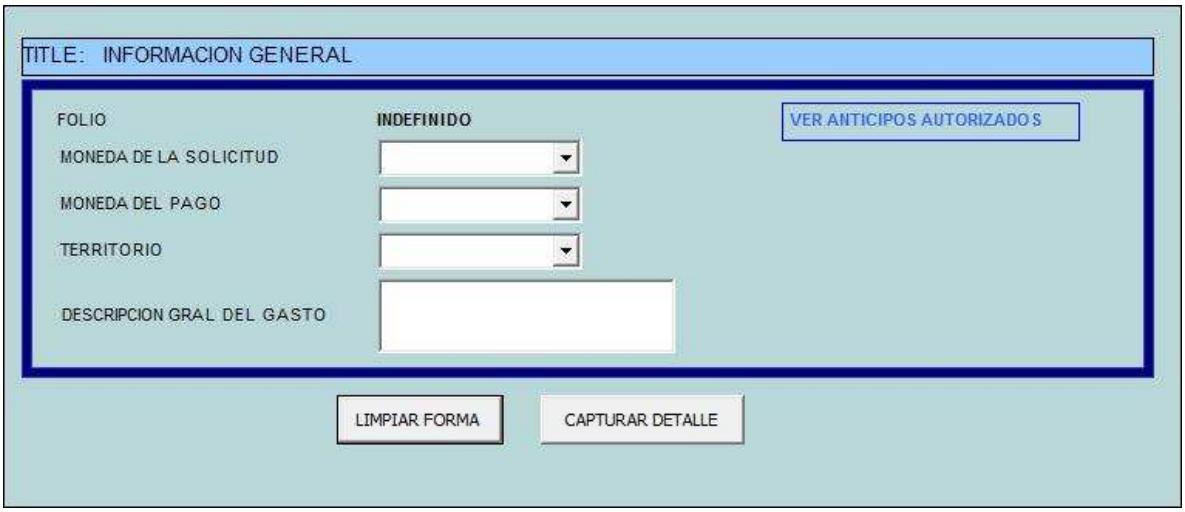


Figura 4.7. Pantalla principal de la comprobación de anticipos.

Reporte – selección de anticipos a comprobar

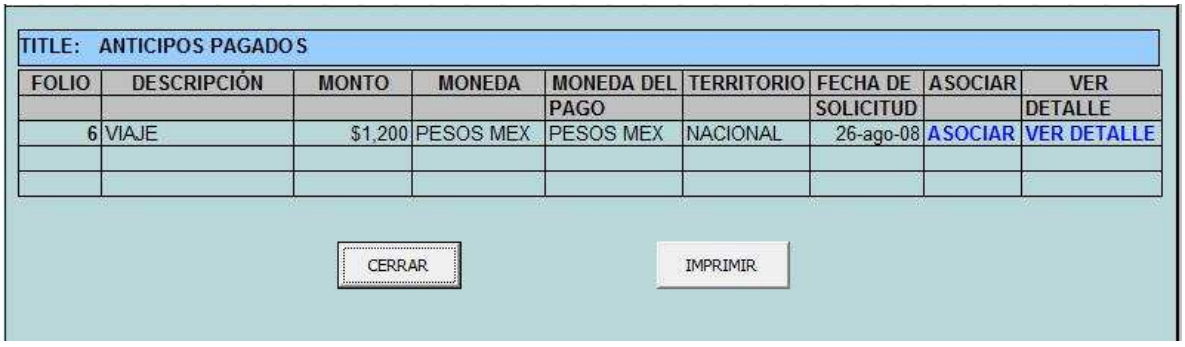


Figura 4.8. Pantalla de selección de anticipos de la comprobación de anticipos.

Pantallas de Comprobación de gastos.

Pantalla principal.

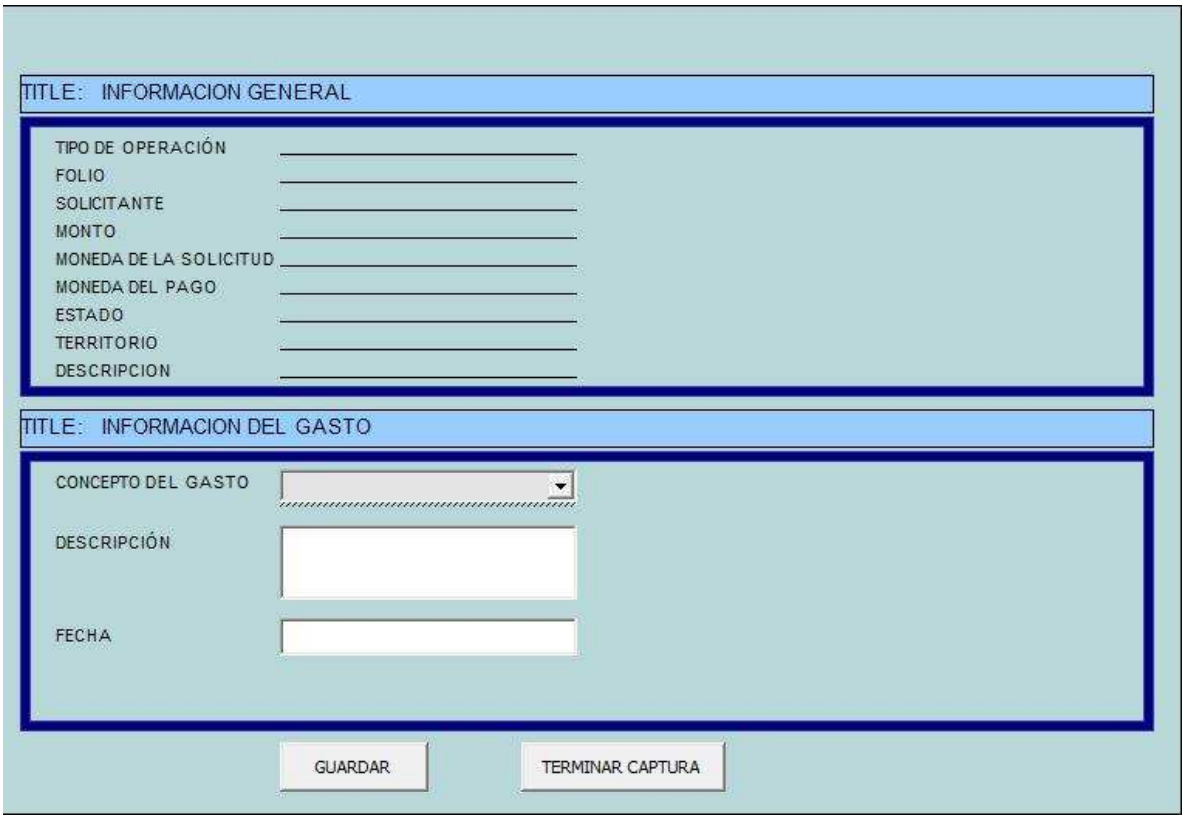


Figura 4.9. Pantalla principal de la comprobación de gastos.

Aquise capturan los datos de cabecera de las comprobaciones de gastos.

Nota: Las pantallas restantes de comprobación de gastos son idénticas a las de anticipos.

Pantalla de resumen de solicitudes.

Figura 4.10. Pantalla del resumen de solicitudes.

Esta pantalla muestra el resumen de una solicitud según se haya seleccionado en lapantalla que la llama.

Pantalla mandar a autorizar solicitud.

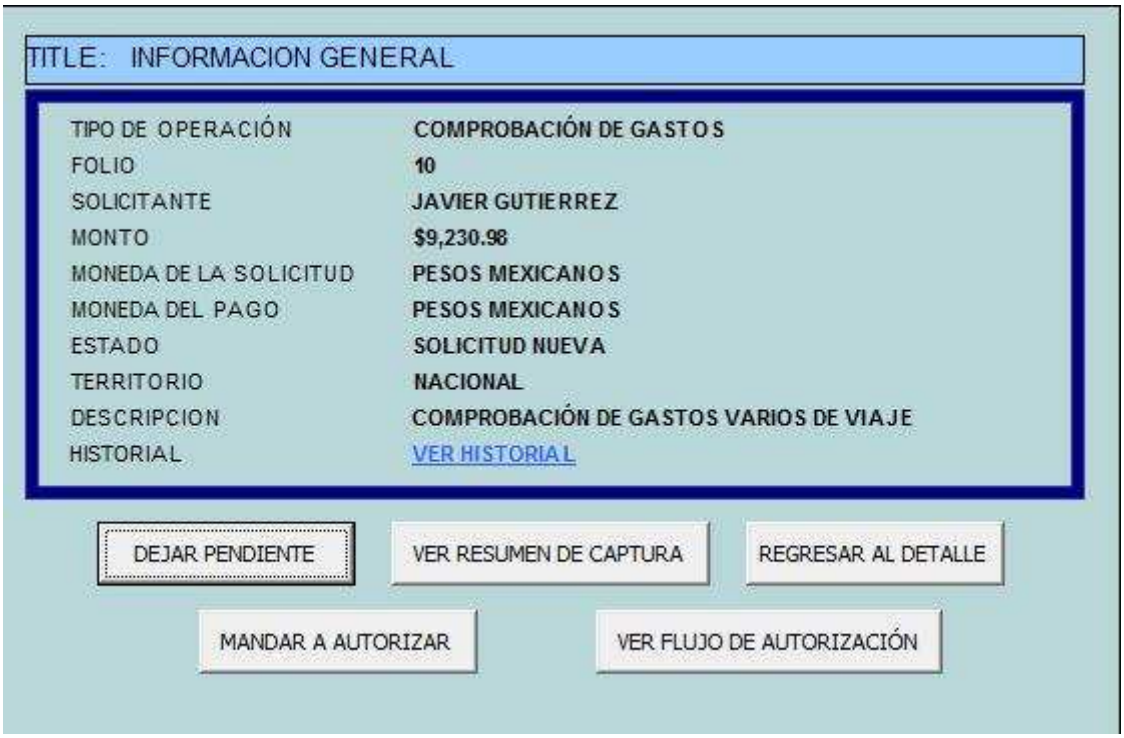


Figura 4.11. Pantalla de la opción “mandar a autorizar”.

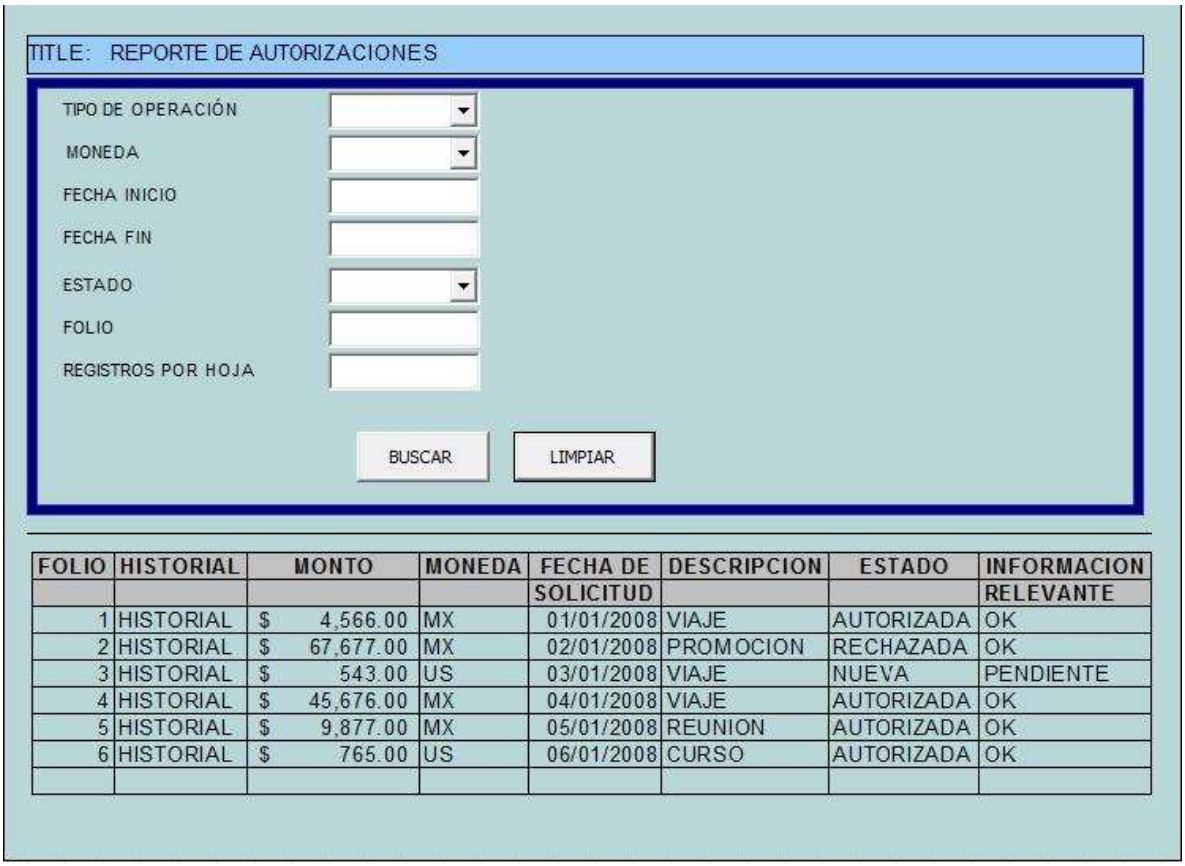
En esta pantalla se le presentan al usuario los datos de la solicitud que acaba de dar por terminada y a la vez se le muestran las opciones que tiene disponibles al momento, entre ellas, dejar la solicitud pendiente para modificarla posteriormente, ver un resumen de los datos capturados, ir a la pantalla de detalle para realizar alguna corrección en el mismo, liberar la solicitud al flujo de autorización, o bien, consultar el flujo de autorización que habrá de seguir la solicitud antes de mandarla a autorizar.

Pantalla de autorizaciones.

Figura 4.12. Pantalla de autorización de solicitudes.

En esta pantalla se le da al autorizador la información de las solicitudes pendientes de autorizar. Desde aquí se puede autorizar la solicitud, rechazarla, imprimir la solicitud o bien, regresar a la pantalla anterior.

Pantalla del reporte de autorizaciones.



FOLIO	HISTORIAL	MONTO	MONEDA	FECHA DE SOLICITUD	DESCRIPCION	ESTADO	INFORMACION RELEVANTE
1	HISTORIAL	$ 4,566.00	MX	01/01/2008	VIAJE	AUTORIZADA	OK
2	HISTORIAL	$ 67,677.00	MX	02/01/2008	PROMOCION	RECHAZADA	OK
3	HISTORIAL	$ 543.00	US	03/01/2008	VIAJE	NUEVA	PENDIENTE
4	HISTORIAL	$ 45,676.00	MX	04/01/2008	VIAJE	AUTORIZADA	OK
5	HISTORIAL	$ 9,877.00	MX	05/01/2008	REUNION	AUTORIZADA	OK
6	HISTORIAL	$ 765.00	US	06/01/2008	CURSO	AUTORIZADA	OK

Figura 4.13. Pantalla del reporte de autorizaciones.

En este reporte se pueden consultar las solicitudes dependiendo de los parámetros alimentados por el usuario. La información que se muestra es una linea por solicitud con los principalesdatos de la misma, entre ellos, el estado que guarda en el sistema.

Pantalla de Historial de las solicitudes.

Figura 4.14. Pantalla del historial de lassolicitudes.

En esta pantalla se pueden consultar todos los movimientos que ha tendo una solicitud desde que se liberó al flujo de autorización hasta la última autorizacióno rechazo y quien realizó dicho movimiento, asi como la fecha y hora del mismo.

Conclusión.

Como se mencionó antes, esta es la primera iteración o versión de lo que será un sistema completo por lo que seguramente hay muchos puntos de mejora y funcionalidades que sería deseable que tuviera, lo cual es algo que se tiene que evaluar y priorizar con todos los interesados, por ejemplo, mientras que el administrador del sistema puede estar esperando que haya un menú de administración para facilitar el mantenimiento y configuración de las tablas, catálogos y parámetros del sistema, por el otro lado puede estar el departamento de contabilidad solicitando que haya una interfase automática que tome los datos de las solicitudes autorizadas para que se generen los cheques correspondientes y las pólizas contables.

Las posibilidades son muchas en un sistema de este tipo ya que a la vez que proporciona control sobre los gastos que normalmente no se tiene sistematizados a este grado, también ayuda a ahorrar recursos en la empresa o institución, permitiendo su utilización en áreas de alta prioridad para la misma. Incluso, como se comentó en la sección de alcances, puede actuar como una sistema B2B[1] con proveedores de bienes y servicios y, al estar disponible en web, no es necesario tener instalado un cliente; basta con tener un explorador como el “internet explorer” o “firefox” por ejemplo.

[1] **B2B** es la abreviatura comercial de la expresión anglosajona ***business to business***: *(comunicaciones de comercio electrónico) de empresa a empresa*, por oposición a las relaciones de comercio entre empresas y consumidores (B2C), o las expresiones menos usadas empresas y gobierno (B2G) o empresas y empleados (B2E)

Esta página se dejó intensionalmente en blanco.

Anexo 1. Creación de la base de datos

```
CREATE DATABASE [base]  ON (NAME = N'base_Data', FILENAME = N'C:\Db\base\base_Data.MDF' , SIZE
= 10, FILEGROWTH = 20%) LOG ON (NAME = N'base_Log', FILENAME = N'C:\Db\base\base_Log.LDF' ,
SIZE = 10, FILEGROWTH = 10%)
 COLLATE SQL_Latin1_General_CP1_CI_AS
GO

exec sp_dboption N'base', N'autoclose', N'false'
GO

exec sp_dboption N'base', N'bulkcopy', N'false'
GO

exec sp_dboption N'base', N'trunc. log', N'false'
GO

exec sp_dboption N'base', N'torn page detection', N'true'
GO

exec sp_dboption N'base', N'read only', N'false'
GO

exec sp_dboption N'base', N'dbo use', N'false'
GO

exec sp_dboption N'base', N'single', N'false'
GO

exec sp_dboption N'base', N'autoshrink', N'false'
GO

exec sp_dboption N'base', N'ANSI null default', N'false'
GO

exec sp_dboption N'base', N'recursive triggers', N'false'
GO

exec sp_dboption N'base', N'ANSI nulls', N'false'
GO

exec sp_dboption N'base', N'concat null yields null', N'false'
GO

exec sp_dboption N'base', N'cursor close on commit', N'false'
GO

exec sp_dboption N'base', N'default to local cursor', N'false'
GO

exec sp_dboption N'base', N'quoted identifier', N'false'
GO

exec sp_dboption N'base', N'ANSI warnings', N'false'
GO

exec sp_dboption N'base', N'auto create statistics', N'true'
GO

exec sp_dboption N'base', N'auto update statistics', N'true'
GO
```

```
if( (@@microsoftversion / power(2, 24) = 8) and (@@microsoftversion & 0xffff >= 724) )
	exec sp_dboption N'base', N'db chaining', N'false'
GO

use [base]
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_autentica_login    Script Date: 02/07/2008 04:36:32 p.m. ******/




/*
Inicia
Nombre:
fn_autentica_login



Funcionalidad:
Verifica la existencia de un usuario y su password alsolicitar acceso a la aplicación

Entradas:
Número de la empresa a la que el usuario se quiere firmar
Número de empleado que pretende firmarse a la aplicación
Tipo de usuario que quiere entrar
	1 = interno
	2 = externo
Clave de acceso que el usuario provee

Proceso:
Toma los datos de entrada y los busca en la BD para verificar si el usuario en realidad existe

Salida:
1 = el usuario existe
0 = el usuario no existe

Ejemplo:
select [dbo].fn_autentica_login(1,123,1,'123')

Termina
*/

CREATE function [dbo].fn_autentica_login
(
	@idempresa int,
	@numempleado int,
	@idtipo_usuario int,
	@pwd varchar(10)
)
returns tinyint
as
```

```
begin
	declare
	@existe tinyint

	/*
	Verifica que los datos de entrada existan en la BD
	*/
	if exists(
		select
		clave_acceso
		from usuario
		where	idempresa = @idempresa and
			numempleado = @numempleado and
			idtipo_usuario = @idtipo_usuario and
			[dbo].user_pwd(rtrim(@pwd),1) = clave_acceso
	)
	/*
	Si existe
	*/
	begin
		set @existe = 1
	end
	else
	/*
	No existe
	*/
	begin
		set @existe = 0
	end

	return(@existe)

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_anexo_obtiene_folio_consecutivo    Script Date:
02/07/2008 04:36:33 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_detalle_anexo_obtiene_folio_consecutivo



Funcionalidad:
Obtiene el identificador del próximo archivo anexo. Se utiliza cuando se va a anexar un archivo nuevo, esta
función obtiene el identificador único de ese archivo anexo

Entradas:
No hay entradas

Proceso:
Obtiene el siguiente identificador consecutivo y se lo asigna al siguiente archivo anexo

Salida:
Número consecutivo

Ejemplo:
select [dbo].fn_detalle_anexo_obtiene_folio_consecutivo

Termina
*/

CREATE function [dbo].fn_detalle_anexo_obtiene_folio_consecutivo
()
returns int
as
begin
        declare
        @consecutivo as int

        set @consecutivo = (select isnull(max(idanexo),0) + 1 from detalle_anexo)

        return(@consecutivo)
end



GO
SET QUOTED_IDENTIFIER OFF
```

```
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_obtiene_aerolinea_boleto_avion    Script Date:
02/07/2008 04:36:33 p.m. ******/




/*
Inicia
Nombre:
fn_detalle_obtiene_aerolinea_boleto_avion



Funcionalidad:
Obtiene la aerolinea del boleto de avión de un detalle específico

Entradas:
Folio del detalle de la solicitud

Proceso:
Obtiene la aerolinea registrada para un boleto de avion de un detalle en particular

Salida:
Nombre de la aerolínea

Ejemplo:
select [dbo].fn_detalle_obtiene_aerolinea_boleto_avion(2)

Termina
*/

CREATE function [dbo].fn_detalle_obtiene_aerolinea_boleto_avion
(
	@detalle int
)
returns varchar(50)
as
begin
	declare
	@aerolinea varchar(50)

	set @aerolinea = (
		select aerolinea
		from detalle_solicitud as de
		left outer join detalle_boleto_avion as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)
```

```
	if @aerolinea is null
	begin
		set @aerolinea = ''
	end

	return(@aerolinea)
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_obtiene_comensales    Script Date: 02/07/2008 04:36:33
p.m. ******/



/*
Inicia
Nombre:
fn_detalle_obtiene_comensales



Funcionalidad:
Obtiene el número de comensales de un detalla específico

Entradas:
Folio del detalle de la solicitud

Proceso:
Obtiene el número de comensales registrados para un detalle en particular. Si no hay comensales registrados,
el default es 1
```

```
Salida:
Número de comensales

Ejemplo:
select [dbo].fn_detalle_obtiene_comensales(2)

Termina
*/

CREATE function [dbo].fn_detalle_obtiene_comensales
(
	@detalle int
)
returns int
as
begin
	declare
	@comensales int

	set @comensales = (
		select comensales
		from detalle_solicitud as de
		left outer join detalle_comensales as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)


	if @comensales is null
	begin
		set @comensales = 1
	end

	return(@comensales)
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_detalle_obtiene_destino_boleto_avion    Script Date: 02/07/2008
04:36:33 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_detalle_obtiene_destino_boleto_avion



Funcionalidad:
Obtiene el destino del boleto de avion de un detalle especifico

Entradas:
Folio del detalle de la solicitud

Proceso:
Obtiene el destino registrado para un boleto de avion de un detalle en particular

Salida:
Destino del boleto de avión

Ejemplo:
select [dbo].fn_detalle_obtiene_destino_boleto_avion(2)
Termina
*/

CREATE function [dbo].fn_detalle_obtiene_destino_boleto_avion
(
	@detalle int
)
returns varchar(50)
as
begin
	declare
	@destino varchar(50)

	set @destino = (
		select destino
		from detalle_solicitud as de
		left outer join detalle_boleto_avion as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)


	if @destino is null
	begin
		set @destino = ''
	end

	return(@destino)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_obtiene_dias    Script Date: 02/07/2008 04:36:34 p.m.
******/




/*
Inicia
Nombre:
fn_detalle_obtiene_dias


Funcionalidad:
Obtiene cuantos dias hay registrados por un detalle especifico

Entradas:
Folio del detalle de la solicitud

Proceso:
Obtiene el numero de dias registrados para un detalle en particular

Salida:
Numero de dias

Ejemplo:
select [dbo].fn_detalle_obtiene_dias(2)

Termina
*/

CREATE function [dbo].fn_detalle_obtiene_dias
(
        @detalle int
)
returns float
as
```

```
begin
	declare
	@dias float

	set @dias = (
		select dias
		from detalle_solicitud as de
		left outer join detalle_dias as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)

	if @dias is null
	begin
		set @dias = 1
	end

	return(round(@dias,2))
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_obtiene_folio_consecutivo    Script Date: 02/07/2008
04:36:33 p.m. ******/




/*
Inicia
Nombre:
fn_detalle_obtiene_folio_consecutivo


Funcionalidad:
Obtiene el siguiente folio de detalle al registrado

Entradas:
```

No hay entradas

Proceso:
Obtiene el numero siguiente del detalle al registrado

Salida:
Numero siguiente del detalle

Ejemplo:
select [dbo].fn_detalle_obtiene_folio_consecutivo()
Termina
*/

```
CREATE function [dbo].fn_detalle_obtiene_folio_consecutivo
()
returns int
as
begin
	declare
	@consecutivo as int

	set @consecutivo = (select isnull(max(detalle),0) + 1 from detalle_solicitud)

	return(@consecutivo)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

/****** Object: User Defined Function dbo.fn_detalle_obtiene_folio_deposito   Script Date: 02/07/2008 04:36:34 p.m. ******/

/*
Inicia
Nombre:
fn_detalle_obtiene_folio_deposito

```
Funcionalidad:
Obtiene el folio de un deposito

Entradas:
Número del detalle del depósito

Proceso:
Obtiene el número de folio del depósito de un detalle de solicitud

Salida:
Folio del depósito

Ejemplo:
select [dbo].fn_detalle_obtiene_folio_deposito(2)
Termina
*/

CREATE function [dbo].fn_detalle_obtiene_folio_deposito
(
	@detalle int
)
returns int
as
begin
	declare
	@folio_deposito int

	set @folio_deposito = (
		select folio_deposito
		from detalle_solicitud as de
		left outer join detalle_deposito as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)


	if @folio_deposito is null
	begin
		set @folio_deposito = 0
	end

	return(@folio_deposito)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
```

```
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_obtiene_hora_salida_boleto_avion    Script Date:
02/07/2008 04:36:34 p.m. ******/




/*
Inicia
Nombre:
fn_detalle_obtiene_hora_salida_boleto_avion



Funcionalidad:
Obtiene la hora de salida del boleto de avion de un detalle especifico

Entradas:
Folio del detalle de la solicitud

Proceso:
Obtiene el hora de salida registrada para un boleto de avion de un detalle en particular

Salida:
Hora de salida de un boleto de avión

Ejemplo:
select [dbo].fn_detalle_obtiene_hora de salida_boleto_avion(2)
Termina
*/

CREATE function [dbo].fn_detalle_obtiene_hora_salida_boleto_avion
(
	@detalle int
)
returns varchar(20)
as
begin
	declare
	@hora_salida varchar(20)

	set @hora_salida = (
		select hora_salida
		from detalle_solicitud as de
		left outer join detalle_boleto_avion as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)


	if @hora_salida is null
	begin
		set @hora_salida = ''
```

```
        end

        return(@hora_salida)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.fn_detalle_obtiene_idkm_por_detalle    Script Date: 02/07/2008
04:36:34 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_detalle_obtiene_idkm_por_detalle


Funcionalidad:
Obtiene el registro de relacion origen/distancia capturado para un detalle

Entradas:
Empresa en la que se esta trabajando
Numero de detalle del que se quiere obtener el la relacion origen/destino

Proceso:
Obtiene la relacion origen/destino para un detalle

Salida:
Identificador del registro

Ejemplo:
```

```
select [dbo].fn_detalle_obtiene_idkm_por_detalle(12,1)
Termina
*/


CREATE function [dbo].fn_detalle_obtiene_idkm_por_detalle
(
	@idempresa int,
	@detalle int
)
returns int
as
begin
	declare
	@idkm int

	set @idkm = (
		select km.idkm
		from solicitud as so
		inner join detalle_solicitud as de on so.folio = de.folio
		left outer join detalle_kilometro as dk on de.detalle = dk.detalle
		left outer join kilometro as km on dk.idkm = km.idkm and so.idempresa = km.idempresa
		where	so.idempresa = @idempresa and
			de.detalle = @detalle
	)

	if @idkm is null
	begin
		set @idkm = 0
	end

	return(@idkm)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.fn_detalle_obtiene_no_viajes_por_detalle    Script Date: 02/07/2008
04:36:34 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_detalle_obtiene_no_viajes_por_detalle



Funcionalidad:
Obtiene el numero de viajes que un detalle tiene para un origen/destino

Entradas:
Empresa en la que se esta trabajando
Numero de detalle del que se quiere obtener el numero de viajes

Proceso:
Obtiene el numero de viajes de un detalle

Salida:
Numero de viajes

Ejemplo:
select [dbo].fn_detalle_obtiene_no_viajes_por_detalle(12,1)
Termina
*/

CREATE function [dbo].fn_detalle_obtiene_no_viajes_por_detalle
(
	@idempresa int,
	@detalle int
)
returns int
as
begin
	declare
	@no_viajes int

	set @no_viajes = (
		select dk.num_viajes
		from solicitud as so
		inner join detalle_solicitud as de on so.folio = de.folio
		left outer join detalle_kilometro as dk on de.detalle = dk.detalle
		left outer join kilometro as km on dk.idkm = km.idkm and so.idempresa = km.idempresa
		where	so.idempresa = @idempresa and
			de.detalle = @detalle
	)


	if @no_viajes is null
	begin
		set @no_viajes = 0
	end
```

```
	return(@no_viajes)
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_obtiene_origen_boleto_avion    Script Date: 02/07/2008
04:36:34 p.m. ******/



/*
Inicia
Nombre:
fn_detalle_obtiene_origen_boleto_avion


Funcionalidad:
Obtiene el origen del boleto de avion de un detalle especifico

Entradas:
Folio del detalle de la solicitud

Proceso:
Obtiene el origen registrado para un boleto de avion de un detalle en particular

Salida:
Origen

Ejemplo:
```

```
select [dbo].fn_detalle_obtiene_origen_boleto_avion(2)
Termina
*/

CREATE function [dbo].fn_detalle_obtiene_origen_boleto_avion
(
	@detalle int
)
returns varchar(50)
as
begin
	declare
	@origen varchar(50)

	set @origen = (
		select origen
		from detalle_solicitud as de
		left outer join detalle_boleto_avion as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)


	if @origen is null
	begin
		set @origen = ''
	end

	return(@origen)
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_detalle_obtiene_precio_unitario    Script Date: 02/07/2008
04:36:34 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_detalle_obtiene_precio_unitario



Funcionalidad:
Obtiene el precio unitario designado para un detalle

Entradas:
Numero de detalle del que se quiere obtener el precio unitario

Proceso:
Obtiene el precio unitario de un detalle

Salida:
Precio unitario

Ejemplo:
select [dbo].fn_detalle_obtiene_precio_unitario(1)
Termina
*/

CREATE function [dbo].fn_detalle_obtiene_precio_unitario
(
	@detalle int
)
returns money
as
begin
	declare
	@precio_unitario money

	set @precio_unitario = (
		select precio_unitario
		from detalle_solicitud as de
		left outer join detalle_precio_unitario_unidades as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)


	if @precio_unitario is null
	begin
		set @precio_unitario = 1
	end

	return(@precio_unitario)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_obtiene_referencia_deposito    Script Date: 02/07/2008
04:36:34 p.m. ******/




/*
Inicia
Nombre:
fn_detalle_obtiene_referencia_deposito


Funcionalidad:
Obtiene la referencia de un deposito hecho por un empleado a favor de la empresa

Entradas:
Folio del detalle de la solicitud

Proceso:
Obtiene la referencia de un deposito específico

Salida:
Referencia de depósito

Ejemplo:
select [dbo].fn_detalle_obtiene_referencia_deposito(2)
Termina
*/


CREATE function [dbo].fn_detalle_obtiene_referencia_deposito
```

```
(
	@detalle int
)
returns varchar(20)
as
begin
	declare
	@referencia_deposito varchar(20)

	set @referencia_deposito = (
		select referencia
		from detalle_solicitud as de
		left outer join detalle_deposito as pu on de.detalle = pu.detalle
		left outer join deposito as dp on pu.folio_deposito = dp.folio
		where	de.detalle = @detalle
	)


	if @referencia_deposito is null
	begin
		set @referencia_deposito = ''
	end

	return(@referencia_deposito)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.fn_detalle_obtiene_total_kilometros    Script Date: 02/07/2008
04:36:34 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_detalle_obtiene_total_kilometros



Funcionalidad:
Obtiene el total de kilometros recorridos para un detalle

Entradas:
Numero de empresa en la que se esta trabajando
Numero de detalle del que se quiere obtener el total de kilometros

Proceso:
Obtiene el total de kilometros de un detalle

Salida:
Total de kilometros

Ejemplo:
select [dbo].fn_detalle_obtiene_total_kilometros(12,1)
Termina
*/

CREATE function [dbo].fn_detalle_obtiene_total_kilometros
(
	@idempresa int,
	@detalle int
)
returns float
as
begin
	declare
	@total_kilometros float

	set @total_kilometros = (
		select km.distancia*dk.num_viajes
		from solicitud as so
		inner join detalle_solicitud as de on so.folio = de.folio
		left outer join detalle_kilometro as dk on de.detalle = dk.detalle
		left outer join kilometro as km on dk.idkm = km.idkm and so.idempresa = km.idempresa
		where	so.idempresa = @idempresa and
			de.detalle = @detalle
	)


	if @total_kilometros is null
	begin
		set @total_kilometros = 0
	end

	return(round(@total_kilometros,2))
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_detalle_obtiene_unidades    Script Date: 02/07/2008 04:36:34
p.m. ******/




/*
Inicia
Nombre:
fn_detalle_obtiene_unidades



Funcionalidad:
Obtiene las unidades para un detalle

Entradas:
Numero de detalle del que se quiere obtener las unidades

Proceso:
Obtiene las unidades de un detalle

Salida:
Unidades

Ejemplo:
select [dbo].fn_detalle_obtiene_unidades(1)
Termina
*/
```

```
CREATE function [dbo].fn_detalle_obtiene_unidades
(
	@detalle int
)
returns float
as
begin
	declare
	@unidades float

	set @unidades = (
		select unidades
		from detalle_solicitud as de
		left outer join detalle_precio_unitario_unidades as pu on de.detalle = pu.detalle
		where	de.detalle = @detalle
	)


	if @unidades is null
	begin
		set @unidades = 1
	end

	return(round(@unidades,2))
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.fn_distribucion_obtiene_folio_consecutivo    Script Date: 02/07/2008
04:36:34 p.m. ******/




/*
Inicia
Nombre:
fn_distribucion_obtiene_folio_consecutivo



Funcionalidad:
Obtiene el siguiente folio de distribucion al registrado

Entradas:
No hay entradas

Proceso:
Obtiene el numero siguiente del distribucion al registrado

Salida:
Numero siguiente de distribucion

Ejemplo:
select [dbo].fn_distribucion_obtiene_folio_consecutivo()
Termina
*/

CREATE function [dbo].fn_distribucion_obtiene_folio_consecutivo
()
returns int
as
begin
	declare
	@consecutivo as int

	set @consecutivo = (select isnull(max(distribucion),0) + 1 from distribucion_solicitud)

	return(@consecutivo)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_flujo_autoriza_obtiene_ultimo_id_rechazo    Script Date:
02/07/2008 04:36:33 p.m. ******/




/*
Inicia
Nombre:
fn_flujo_autoriza_obtiene_ultimo_id_rechazo



Funcionalidad:
Obtiene el id del historial del ultimo rechazo

Entradas:
Folio de la solicitud

Proceso:
Revisa el historial de ese tipo de solicitud y obtiene el ultimo id de rechazo que le corresponde a ese folio

Salida:
Id del ultimo rechazo

Ejemplo:
select [dbo].fn_flujo_autoriza_obtiene_ultimo_id_rechazo(1)
Termina
*/

CREATE function [dbo].fn_flujo_autoriza_obtiene_ultimo_id_rechazo
(
	@folio int
)
returns int
as
begin
	declare
	@ultimo_id int

	set  @ultimo_id = (
		select top 1 isnull(idhistorial,0)
		from historial_solicitud
		where	substring(idstatus,1,1) = 'r' and
			folio = @folio
		order by fecha desc
	)

	if @ultimo_id is null
	begin
		set @ultimo_id = 0
	end
```

```
	return(@ultimo_id)
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_alta_tipo_operacion    Script Date: 02/07/2008 04:36:31
p.m. ******/



/*
Inicia
Nombre:
fn_obtiene_alta_tipo_operacion


Funcionalidad:
Revisa si el tipo de operación dado está activo para esa empresa

Entradas:
Empresa en la que se esta trabajando
Tipo de operacion, cuya actividad o inactividad quiere ser evaluada

Proceso:
Revisa si el tipo de operacion se encuentra activo para la empresa dada

Salida:
1 = Si esta activa
0 = No esta activa
```

```
Ejemplo:
select [dbo].fn_obtiene_alta_tipo_operacion(1,1)
Termina
*/


CREATE function [dbo].fn_obtiene_alta_tipo_operacion
(
	@idempresa int,
	@idtipo_operacion int
)
returns tinyint
as
begin
	declare
	@existe tinyint

	if exists(
		select
		idtipo_operacion
		from tipo_operacion
		where	idempresa = @idempresa and
			idtipo_operacion = @idtipo_operacion and
			idstatus = 'a'
	)
	begin
		set @existe = 1
	end
	else
	begin
		set @existe = 0
	end

	return(@existe)
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.fn_obtiene_autorizacion_perfil    Script Date: 02/07/2008 04:36:32
p.m. ******/



/*
Inicia
Nombre:
fn_obtiene_autorizacion_perfil



Funcionalidad:
Obtiene el tipo de autorización que tiene un usuario o un perfil de usuario.

Entradas:
Empresa en la que se esta trabajando
Identificador del usuario o perfil del que se quiere obtener su tipo de autorización.
Accion a seguir
        p = revisar el tipo de autorizacion para un perfil
        a = revisar el tipo de autorizacion para un usuario

Proceso:
Obtiene el tipo de autorizacion (Detalle por detalle o Masiva) de un usuario o perfil

Salida:
Tipo de autorizacion
        0 = Detalle por detalla
        1 = Masiva

Ejemplo:
select [dbo].fn_obtiene_autorizacion_perfil(1,1,'p')
Termina
*/

CREATE function [dbo].fn_obtiene_autorizacion_perfil
(
        @idempresa int,
        @id int,
        @proviene char(1)
)
returns char(1)
as
begin
        declare
        @tipo_autorizacion char(1),
        @idperfil int

        /*
        Si viene de la pagina de perfiles
        */
        if @proviene = 'p'
        begin
                set @idperfil = @id
                set @tipo_autorizacion = (
                        select
                        isnull(tipo_autorizacion,0)
                        from perfil_tipo_autorizacion
```

```
                        where   idempresa = @idempresa and
                                idperfil = @idperfil
                )

                if @tipo_autorizacion is not null
                begin
                        set @tipo_autorizacion = @tipo_autorizacion
                end
                else
                begin
                        set @tipo_autorizacion = 0
                end
        end

        /*
        Si viene de la pagina de autorizaciones, es decir de la pantalla de autorizaciones
        */
        if @proviene = 'a'
        begin
                set @idperfil = (select idperfil from usuario where idempresa = @idempresa and
numempleado = @id)
                set @tipo_autorizacion = (
                        select
                        case tipo_autorizacion
                                when 'D' then '0'
                                when 'M' then '1'
                        end as tipo_autorizacion
                        from perfil_tipo_autorizacion
                        where   idempresa = @idempresa and
                                idperfil = @idperfil
                )

                if @tipo_autorizacion is not null
                begin
                        set @tipo_autorizacion = @tipo_autorizacion
                end
                else
                begin
                        set @tipo_autorizacion = 0
                end
        end

        return(@tipo_autorizacion)

end




GO
SET QUOTED_IDENTIFIER OFF
```

```
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_ccostos_por_empleado    Script Date: 02/07/2008
04:36:32 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_ccostos_por_empleado



Funcionalidad:
Obtiene el centro de costo por default de un empleado

Entradas:
Empresa en la que se trabaja
Numero de empleado del que se quiere obtener el centro de costo por default

Proceso:
Obtiene el centro de costo por defecto de un empleado

Salida:
Centro de costo

Ejemplo:
select [dbo].fn_obtiene_ccostos_por_empleado(12,123)
Termina
*/
CREATE function [dbo].fn_obtiene_ccostos_por_empleado
(
	@idempresa int,
	@numempleado int
)
returns int
as
begin
	declare
	@ccostos int

	set @ccostos = (
		select isnull(ccostos,0)
		from usuario
		where	idempresa = @idempresa and
			numempleado = @numempleado
	)
```

```
	return(@ccostos)

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_centro_costo    Script Date: 02/07/2008 04:36:32 p.m.
******/




/*
Inicia
Nombre:
fn_obtiene_centro_costo



Funcionalidad:
Obtiene el número y nombre del centro de costo de un empleado dado

Entradas:
Empresa en la que se trabaja
Numero de empleado del que se quiere obtener el centro de costo

Proceso:
Obtiene el centro de costo por defecto de un empleado

Salida:
Centro de costo (número y nombre)

Ejemplo:
select [dbo].fn_obtiene_centro_costo(12,123)
Termina
```

```
*/

CREATE function [dbo].fn_obtiene_centro_costo
(
	@idempresa int,
	@numempleado int
)
returns varchar(250)
as
begin
	return(
		select
		rtrim(isnull(cc.idccostos,'')) + ' ' + rtrim(isnull(cc.descripcion,''))
		from usuario as us
		inner join centro_costo as cc on us.ccostos = cc.idccostos
		where	us.idempresa = @idempresa and
			us.idempresa = cc.idempresa and
			us.numempleado = @numempleado
	)
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_concepto_tipo_operacion    Script Date: 02/07/2008
04:36:32 p.m. ******/







/*
Inicia
Nombre:
fn_obtiene_concepto_tipo_operacion
```

Funcionalidad:
Obtiene si un concepto de gasto está asignado a un tipo de operación. Se utiliza en el catálogo de concepto de gasto para desplegar a qué tipos de operación pertenece ese concepto

Entradas:
Empresa en la que se trabaja
Concepto de gasto
Tipo de operación

Proceso:
Obtiene si el concepto de gasto está asignado al tipo de operación dado

Salida:
1 = Si esta asginado
0 = No esta asignado

Ejemplo:
select [dbo].fn_obtiene_concepto_tipo_operacion(1,1,1)
Termina
*/

```
CREATE function [dbo].fn_obtiene_concepto_tipo_operacion
(
	@idempresa int,
	@idconcepto int,
	@idtipo_operacion int
)
returns tinyint
as
begin
	declare
	@existe tinyint

	if exists(
		select
		idtipo_operacion
		from concepto_gasto_tipo_concepto
		where	idempresa = @idempresa and
			idconcepto = @idconcepto and
			idtipo_operacion = @idtipo_operacion
	)
	begin
		set @existe = 1
	end
	else
	begin
		set @existe = 0
	end

	return(@existe)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_costo_por_kilometro    Script Date: 02/07/2008 04:36:31
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_costo_por_kilometro



Funcionalidad:
Obtiene el costo por kilometro dependiendo de la moneda de la solicitud

Entradas:
Empresa en la que se esta trabajando
Moneda de la solicitud

Proceso:
Obtiene el costo por kilometro para esa moneda

Salida:
Costo por kilometro

Ejemplo:
select [dbo].fn_obtiene_costo_por_kilometro (12,1)
Termina
*/

CREATE function [dbo].fn_obtiene_costo_por_kilometro
(
	@idempresa int,
	@idmoneda int
)
returns money
as
begin
	declare
	@costo_por_kilometro money

	set @costo_por_kilometro = (
		select costo_por_km
```

```
		from costo_por_km as so
		where	so.idempresa = @idempresa and
			so.idmoneda = @idmoneda
	)

	if @costo_por_kilometro is null
	begin
		set @costo_por_kilometro = 0
	end

	return(@costo_por_kilometro)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_descripcion_status    Script Date: 02/07/2008 04:36:31
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_descripcion_status



Funcionalidad:
Obtiene el nombre del status dado

Entradas:
Identificador del status

Proceso:
Obtiene la descripcion o nombre del status provisto

Salida:
Nombre de status
```

```
Ejemplo:
select [dbo].fn_obtiene_descripcion_status ('ep')
Termina
*/

CREATE function [dbo].fn_obtiene_descripcion_status
(
	@idstatus char(2)
)
returns varchar(100)
as
begin
	return(
		select top 1 isnull(st.descripcion_1,'')
		from status as st
		where	idstatus = @idstatus
	)
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_destino_kilometro    Script Date: 02/07/2008 04:36:32
p.m. ******/







/*
Inicia
Nombre:
fn_obtiene_destino_kilometro



Funcionalidad:
Obtiene el destino del viaje de una relacion origen/destino
```

```
Entradas:
Empresa en la que se esta trabajando
Identificador de la relacion origen/destino

Proceso:
Obtiene el nombre del lugar destino

Salida:
Nombre del destino

Ejemplo:
select [dbo].fn_obtiene_destino_kilometro(12,3)
Termina
*/

CREATE function [dbo].fn_obtiene_destino_kilometro
(
	@idempresa int,
	@idkm int
)
returns varchar(50)
as
begin
	declare
	@destino varchar(50)

	set @destino = (select destino from kilometro where idempresa = @idempresa and idkm = @idkm)

	if @destino is null
	begin
		set @destino = 'No definido'
	end

	return(left(@destino,50))

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.fn_obtiene_distancia_kilometro    Script Date: 02/07/2008 04:36:32
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_distancia_kilometro



Funcionalidad:
Obtiene la distancia en kilometros de una relacion origen/destino

Entradas:
Empresa en la que se esta trabajando
Identificador de la relacion origen/destino

Proceso:
Obtiene la distancia en kilometros para una moneda y empresa

Salida:
Distancia entre el origen y el destino

Ejemplo:
select [dbo].fn_obtiene_distancia_kilometro(12,3)
Termina
*/

CREATE function [dbo].fn_obtiene_distancia_kilometro
(
	@idempresa int,
	@idkm int
)
returns float
as
begin
	declare
	@distancia float

	set @distancia = (select distancia from kilometro where idempresa = @idempresa and idkm =
@idkm)

	if @distancia is null
	begin
		set @distancia = 0
	end

	return(round(@distancia,2))

end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_email    Script Date: 02/07/2008 04:36:32 p.m. ******/



/*
Inicia
Nombre:
fn_obtiene_email


Funcionalidad:
Obtiene el correo electrónico de un empleado

Entradas:
Empresa en la que se esta trabajando
Numero de empleado del que se quiere obtener el correo electrónico

Proceso:
Obtiene el correo electrónico de un empleado

Salida:
Correo electrónico

Ejemplo:
select [dbo].fn_obtiene_email(12,3)
Termina
*/

CREATE function [dbo].fn_obtiene_email
(
	@idempresa int,
	@numempleado int
```

```
)
returns varchar(50)
as
begin
	declare
	@email varchar(50)

	set @email = (
		select rtrim(email)
		from usuario
		where	idempresa = @idempresa and
			numempleado = @numempleado
	)

	if @email is null
	begin
		set @email = ''
	end

	return(rtrim(@email))
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_fecha_hoy    Script Date: 02/07/2008 04:36:31 p.m.
******/
/*
Inicia
Nombre:
fn_obtiene_fecha_hoy



Funcionalidad:
Obtiene la fecha actual del sistema y la convierte en formato mdy sin hora

Entradas:
La fecha actual

Proceso:
Convierte la fecha actual a formato mdy y le quita la hora
```

Salida:
Fecha actual

Ejemplo:
select [dbo].fn_obtiene_fecha_hoy(getdate())
Termina
*/

CREATE function [dbo].fn_obtiene_fecha_hoy
(
	@fecha_hoy datetime
)
returns datetime
as
begin
	declare
	@fecha datetime

	set @fecha = convert(datetime,convert(char(10),@fecha_hoy,101))

	return(@fecha)
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

/****** Object: User Defined Function dbo.fn_obtiene_iva_por_detalle   Script Date: 02/07/2008 04:36:33 p.m.
******/

/*
Inicia
Nombre:
fn_obtiene_iva_por_detalle

Funcionalidad:
Obtiene el IVA capturado para un detalle

Entradas:
Numero de detalle del que se quiere obtener el IVA

```
Proceso:
Obtiene el valor del IVA para un detalle

Salida:
Valor de IVA

Ejemplo:
select [dbo].fn_obtiene_iva_por_detalle (1)
Termina
*/

CREATE function [dbo].fn_obtiene_iva_por_detalle
(
	@detalle int
)
returns money
as
begin
	declare
	@iva_por_detalle money

	set @iva_por_detalle = (
		select iva
		from detalle_solicitud
		where	detalle = @detalle
	)

	if @iva_por_detalle is null
	begin
		set @iva_por_detalle = 0
	end

	return(round(@iva_por_detalle,2))
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_iva_valor_por_detalle    Script Date: 02/07/2008
04:36:33 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_obtiene_iva_valor_por_detalle



Funcionalidad:
Obtiene el codigo del IVA capturado para un detalle

Entradas:
Numero de detalle del que se quiere obtener el codigo del IVA

Proceso:
Obtiene el codigo del IVA para un detalle

Salida:
Codigo de IVA

Ejemplo:
select [dbo].fn_obtiene_iva_valor_por_detalle (1)
Termina
*/

CREATE function [dbo].fn_obtiene_iva_valor_por_detalle
(
	@detalle int
)
returns char(3)
as
begin
	declare
	@iva_valor_por_detalle char(3)

	set @iva_valor_por_detalle = (
		select idiva
		from detalle_solicitud
		where	detalle = @detalle
	)

	if @iva_valor_por_detalle is null
	begin
		set @iva_valor_por_detalle = '0'
	end

	return(rtrim(@iva_valor_por_detalle))
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_logo    Script Date: 02/07/2008 04:36:31 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_logo



Funcionalidad:
Obtiene el nombre del archivo logotipo de una empresa

Entradas:
Numero de la empresa de la que se quiere obtener el logotipo

Proceso:
Obtiene el nombre del archivo logotipo de una empresa

Salida:
Nombre de archivo

Ejemplo:
select [dbo].fn_obtiene_logo (1)
Termina
*/

CREATE function [dbo].fn_obtiene_logo
(
	@idempresa int
)
returns varchar(50)
as
begin
	return(
		select rtrim(isnull(logo,''))
		from empresa
		where idempresa = @idempresa
	)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_monto_limite    Script Date: 02/07/2008 04:36:33 p.m.
******/




/*
Inicia
Nombre:
fn_obtiene_monto_limite



Funcionalidad:
Obtiene el monto limite permitido para un usuario

Entradas:
Nivel del usuario
Concepto de gasto
Empresa en la que se trabaja
Moneda de la solicitud
Territorio del gasto
Tipo de operación

Proceso:
Se obtiene un monto limite por nivel, concepto de gasto, moneda, territorio y tipo de operación

Salida:
Monto limite permitido o -1 (No hay monto l[imite definido)

Ejemplo:
select [dbo].fn_obtiene_monto_limite (1,1,1,1,1,1)
Termina
*/

CREATE function [dbo].fn_obtiene_monto_limite
(
        @idnivel int,
```

```
	@idconcepto int,
	@idempresa int,
	@idmoneda int,
	@idterritorio int,
	@idtipo_operacion int
)
returns money
as
begin
	declare
	@monto_limite money

	set @monto_limite = (
		select monto_limite
		from monto_limite
		where	idnivel = @idnivel and
			idconcepto = @idconcepto and
			idempresa = @idempresa and
			idmoneda = @idmoneda and
			idterritorio = @idterritorio and
			idtipo_operacion = @idtipo_operacion
	)


	if @monto_limite is null
	begin
		set @monto_limite = -1
	end
	else
	begin
		set @monto_limite = @monto_limite
	end

	/*
	Si se define a @monto_limite = -1 quiere decir que no hay monto limite registrado para esa
combinacion
	de parametros
	*/
	return(@monto_limite)
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
```

```
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_nombre    Script Date: 02/07/2008 04:36:32 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_nombre



Funcionalidad:
Obtiene el nombre de un usuario

Entradas:
Empresa en la que se trabaja
Numero de empleado del que se quiere obtener el nombre

Proceso:
Obtiene el nombre de un empleado

Salida:
Nombre de un empleado

Ejemplo:
select [dbo].fn_obtiene_nombre (1,1234)
Termina
*/

CREATE function [dbo].fn_obtiene_nombre
(
	@idempresa int,
	@numempleado int
)
returns varchar(250)
as
begin
	return(
		select rtrim(isnull(nombre,'')) + ' ' + rtrim(isnull(paterno,'')) + ' ' + rtrim(isnull(materno,''))
		from usuario
		where	idempresa = @idempresa and
			numempleado = @numempleado
	)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_nombre_proveedor    Script Date: 02/07/2008 04:36:32
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_nombre_proveedor



Funcionalidad:
Obtiene el nombre de un proveedor

Entradas:
Identificador del proveedor del que se quiere obtener el nombre

Proceso:
Obtiene el nombre de un proveedor

Salida:
Nombre de un proveedor

Ejemplo:
select [dbo].fn_obtiene_nombre_proveedor (15)
Termina
*/

CREATE function [dbo].fn_obtiene_nombre_proveedor
(
        @idproveedor int
)
returns varchar(50)
as
begin
        declare
        @nombre varchar(50)

        set @nombre = (
                select
```

```
                nombre
                from proveedor
                where idproveedor = @idproveedor
        )
        if @nombre is null
        begin
                set @nombre = ''
        end

        return(@nombre)

end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_numero_solicitudes_por_status    Script Date:
02/07/2008 04:36:33 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_numero_solicitudes_por_status



Funcionalidad:
   ###############################################################################################

Entradas:
Empresa en la que se configura el flujo
Identificador del tipo de operación
Estado

Proceso:
Cuenta el numero de solicitudes en el estado dado
```

```
Salida:
Numero de solicitudes

Ejemplo:
select [dbo].fn_obtiene_numero_solicitudes_por_status (1,1,'ep')
Termina
*/

CREATE function [dbo].fn_obtiene_numero_solicitudes_por_status
(
	@idempresa int,
	@idtipo_operacion int,
	@idstatus char(2)
)
returns int
as
begin
	return(
		select count(folio)
		from solicitud
		where	idempresa = @idempresa and
			idstatus = @idstatus and
			idtipo_operacion = @idtipo_operacion
	)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_operacion_perfil    Script Date: 02/07/2008 04:36:32
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_operacion_perfil
```

Funcionalidad:
Obtiene cuántas solicitudes en proceso puede tener un empleado antes de solicitar una nueva. Entiéndase por solicitudes en proceso aquellas que están siendo partícipes en el flujo de autorización. No se pueden pedir solicitudes cuando haya N solicitudes en proceso. Esto es parte de hacer cumplir la política empresarial.

Entradas:
Empresa en la que se configura el flujo
Perfil del empleado
Identificador del objeto del menú (Numero del objeto de la solicitud en cuestión)

Proceso:
Cuenta el numero de solicitudes en el estado dado

Salida:
Numero de solicitudes

Ejemplo:
select [dbo].fn_obtiene_operacion_perfil (1,1,1)
Termina
*/

```
CREATE function [dbo].fn_obtiene_operacion_perfil
(
	@idempresa int,
	@idperfil int,
	@idobjeto int
)
returns int
as
begin
	declare
	@tipo_operacion varchar(50),
	@idtipo_operacion int,
	@operacion_proceso int

	set @tipo_operacion = (
		select rtrim(tp.descripcion)
		from objeto as ob
		inner join tipo_operacion as tp on lower(rtrim(ob.descripcion)) = lower(rtrim(tp.nombre))
		where	ob.idobjeto = @idobjeto and
			tp.idempresa = @idempresa
	)

	set @idtipo_operacion = (select [dbo].fn_obtiene_tipo_operacion(@idempresa,@tipo_operacion))

	set @operacion_proceso = (
		select
		isnull(op_proceso,0)
		from operacion_proceso
		where	idempresa = @idempresa and
			idperfil = @idperfil and
			idtipo_operacion = @idtipo_operacion
	)

	if @operacion_proceso is not null
	begin
		set @operacion_proceso = @operacion_proceso
	end
	else
	begin
		set @operacion_proceso = 0
```

```
        end

        return(@operacion_proceso)

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_origen_kilometro    Script Date: 02/07/2008 04:36:32
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_origen_kilometro



Funcionalidad:
Obtiene el origen del viaje de una relacion origen/destino

Entradas:
Empresa en la que se esta trabajando
Identificador de la relacion origen/destino

Proceso:
Obtiene el nombre del lugar origen

Salida:
Nombre del origen

Ejemplo:
select [dbo].fn_obtiene_origen_kilometro(12,3)
Termina
*/
```

```
CREATE function [dbo].fn_obtiene_origen_kilometro
(
	@idempresa int,
	@idkm int
)
returns varchar(50)
as
begin
	declare
	@origen varchar(50)

	set @origen = (select origen from kilometro where idempresa = @idempresa and idkm = @idkm)

	if @origen is null
	begin
		set @origen = 'No definido'
	end

	return(left(@origen,50))

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_porcentaje_iva_por_detalle    Script Date: 02/07/2008
04:36:33 p.m. ******/




/*
Inicia
Nombre:
```

fn_obtiene_porcentaje_iva_por_detalle

Funcionalidad:
Obtiene el porcentaje del IVA capturado para un detalle

Entradas:
Numero de detalle del que se quiere obtener el porcentaje del IVA

Proceso:
Obtiene el porcentaje del IVA para un detalle

Salida:
Porcentaje de IVA

Ejemplo:
select [dbo].fn_obtiene_porcentaje_iva_por_detalle (1)
Termina
*/

```
CREATE function [dbo].fn_obtiene_porcentaje_iva_por_detalle
(
	@detalle int
)
returns float
as
begin
	declare
	@iva_valor_por_detalle char(3),
	@porcentaje float

	set @iva_valor_por_detalle = (select idiva from detalle_solicitud where detalle = @detalle)

	set @porcentaje = (
		select porcentaje
		from iva
		where	rtrim(identificador) = rtrim(@iva_valor_por_detalle)
	)

	if @porcentaje is null
	begin
		set @porcentaje = 0
	end

	return(round(@porcentaje,2))
end
```

GO

```
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_ppto_comprometido    Script Date: 02/07/2008 04:36:32
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_ppto_comprometido



Funcionalidad:
Obtiene el presupuesto comprometido para una empresa, cuenta contable, fecha y centro de costo. El
presupuesto comprometido es aquél que representan las solicitudes que todavía se encuentran en el flujo de
autorización, cuyo monto está en proceso de ser erogado.

Entradas:
Empresa en la que se esta trabajando
Cuenta contable de la que se quiere obtener el presupuesto comprometido
Fecha del ejercicio del ppto
Centro de costo

Proceso:
Obtiene cuanto presupuesto se ha comprometido

Salida:
Monto comprometido

Ejemplo:
select [dbo].fn_obtiene_ppto_comprometido (1,'567612',getdate(),1)
Termina
*/

CREATE function [dbo].fn_obtiene_ppto_comprometido
(
        @idempresa int,
        @ccontable char(6),
        @fecha_gasto datetime,
        @ccostos int
)
returns money
as
begin
        declare
```

```
	@ppto_comprometido money

	set @ppto_comprometido = (
		select isnull(ppto_comprometido,0)
		from presupuesto
		where	idempresa = @idempresa and
		rtrim(ccontable) = rtrim(@ccontable) and
		anio = year(@fecha_gasto) and
		mes = month(@fecha_gasto) and
		identidad = @ccostos
	)

	if @ppto_comprometido is null
	begin
		set @ppto_comprometido = 0
	end

	return(@ppto_comprometido)
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_ppto_gastado    Script Date: 02/07/2008 04:36:32 p.m.
******/









/*
Inicia
Nombre:
fn_obtiene_ppto_gastado



Funcionalidad:
```

Obtiene el presupuesto comprometido para una empresa, cuenta contable, fecha y centro de costo. El presupuesto gastado es qu

Entradas:
Empresa en la que se esta trabajando
Cuenta contable de la que se quiere obtener el presupuesto gastado
Fecha del ejercicio del ppto
Centro de costo

Proceso:
Obtiene cuanto presupuesto se ha gastado

Salida:
Monto comprometido

Ejemplo:
select [dbo].fn_obtiene_ppto_gastado (1,'567612',getdate(),1)
Termina
*/

```
CREATE function [dbo].fn_obtiene_ppto_gastado
(
	@idempresa int,
	@ccontable char(6),
	@fecha_gasto datetime,
	@ccostos int
)
returns money
as
begin
	declare
	@ppto_gastado money

	set @ppto_gastado = (
		select isnull(ppto_gastado,0)
		from presupuesto
		where	idempresa = @idempresa and
		rtrim(ccontable) = rtrim(@ccontable) and
		anio = year(@fecha_gasto) and
		mes = month(@fecha_gasto) and
		identidad = @ccostos
	)

	if @ppto_gastado is null
	begin
		set @ppto_gastado = 0
	end

	return(@ppto_gastado)
end




GO
SET QUOTED_IDENTIFIER OFF
```

```
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_ppto_idmoneda    Script Date: 02/07/2008 04:36:32 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_ppto_idmoneda



Funcionalidad:
Revisa si hay presupuesto registrado en la moneda dada

Entradas:
Empresa en la que se esta trabajando
Cuenta contable
Fecha del ejercicio del ppto
Centro de costo

Proceso:
Revisa si hay presupuesto registrado en la moneda dada. Si si hay prespuesto, se actualiza; si no hay prespuesto entonces se

Salida:
0 = No hay presupuesto registrado
0 <> Si hay presupuesto registrado

Ejemplo:
select [dbo].fn_obtiene_ppto_idmoneda (1,'567612',getdate(),1)
Termina
*/

CREATE function [dbo].fn_obtiene_ppto_idmoneda
(
	@idempresa int,
	@ccontable char(6),
	@fecha_gasto datetime,
	@ccostos int
)
returns int
as
begin
	declare
	@idmoneda int
```

```
	set @idmoneda = (
		select isnull(idmoneda,0)
		from presupuesto
		where	idempresa = @idempresa and
		rtrim(ccontable) = rtrim(@ccontable) and
		anio = year(@fecha_gasto) and
		mes = month(@fecha_gasto) and
		identidad = @ccostos
	)

	if @idmoneda is null
	begin
		set @idmoneda = 0
	end

	return(@idmoneda)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_ppto_presupuestado    Script Date: 02/07/2008 04:36:32
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_ppto_presupuestado


Funcionalidad:
Obtiene el presupuesto presupuestado para una empresa, cuenta contable, fecha y centro de costo. El
presupuesto presupuestad

Entradas:
Empresa en la que se esta trabajando
```

```
Cuenta contable de la que se quiere obtener el presupuesto presupuestado
Fecha del ejercicio del ppto
Centro de costo

Proceso:
Obtiene cuanto presupuesto se ha destinado

Salida:
Monto comprometido

Ejemplo:
select [dbo].fn_obtiene_ppto_presupuestado (1,'567612',getdate(),1)
Termina
*/

CREATE function [dbo].fn_obtiene_ppto_presupuestado
(
	@idempresa int,
	@ccontable char(6),
	@fecha_gasto datetime,
	@ccostos int
)
returns money
as
begin
	declare
	@ppto_presupuestado money

	set @ppto_presupuestado = (
		select isnull(ppto_presupuestado,0)
		from presupuesto
		where	idempresa = @idempresa and
		rtrim(ccontable) = rtrim(@ccontable) and
		anio = year(@fecha_gasto) and
		mes = month(@fecha_gasto) and
		identidad = @ccostos
	)

	if @ppto_presupuestado is null
	begin
		set @ppto_presupuestado = 0
	end

	return(@ppto_presupuestado)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
```

```
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_propina_por_detalle    Script Date: 02/07/2008 04:36:33
p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_propina_por_detalle



Funcionalidad:
Obtiene la propina capturada para un detalle

Entradas:
Numero de detalle del que se quiere obtener la propina

Proceso:
Obtiene la propina para un detalle

Salida:
Valor de propina

Ejemplo:
select [dbo].fn_obtiene_propina_por_detalle (1)
Termina
*/

CREATE function [dbo].fn_obtiene_propina_por_detalle
(
	@detalle int
)
returns money
as
begin
	declare
	@propina_por_detalle money

	set @propina_por_detalle = (
		select propina
		from detalle_solicitud
		where    detalle = @detalle
	)

	if @propina_por_detalle is null
	begin
		set @propina_por_detalle = 0
	end

	return(round(@propina_por_detalle,2))
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_puesto    Script Date: 02/07/2008 04:36:32 p.m. ******/



/*
Inicia
Nombre:
fn_obtiene_puesto


Funcionalidad:
Obtiene el puesto de un empleado

Entradas:
Empresa en la que se esta trabajando
Numero de empleado del que se quiere obtener el puesto

Proceso:
Obtiene el puesto de un empleado

Salida:
Nombre del puesto
```

```
Ejemplo:
select [dbo].fn_obtiene_puesto (1,1235)
Termina
*/

CREATE function [dbo].fn_obtiene_puesto
(
	@idempresa int,
	@numempleado int
)
returns varchar(250)
as
begin
	return(
		select
		rtrim(isnull(us.puesto,''))
		from usuario as us
		where	us.idempresa = @idempresa and
			us.numempleado = @numempleado
	)
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_status_por_folio    Script Date: 02/07/2008 04:36:33 p.m.
******/




/*
Inicia
Nombre:
```

fn_obtiene_status_por_folio

Funcionalidad:
Obtiene el estado de una solicitud

Entradas:
Folio de la solicitud de la que se quiere saber el folio

Proceso:
Obtiene el estado de una solicitud

Salida:
Estado

Ejemplo:
select [dbo].fn_obtiene_status_por_folio (12)
Termina
*/

```
CREATE function [dbo].fn_obtiene_status_por_folio
(
	@folio int
)
returns char(2)
as
begin
	return(select isnull(idstatus,'') from solicitud where folio = @folio)
end
```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

/****** Object: User Defined Function dbo.fn_obtiene_subtotal_por_detalle   Script Date: 02/07/2008 04:36:33 p.m. ******/

```
/*
Inicia
Nombre:
fn_obtiene_subtotal_por_detalle



Funcionalidad:
Obtiene el subtotal para un detalle

Entradas:
Numero de detalle del que se quiere obtener el subtotal

Proceso:
Obtiene el subtotal para un detalle

Salida:
Subtotal

Ejemplo:
select [dbo].fn_obtiene_subtotal_por_detalle (1)
Termina
*/

CREATE function [dbo].fn_obtiene_subtotal_por_detalle
(
	@detalle int
)
returns money
as
begin
	declare
	@subtotal_por_detalle money

	set @subtotal_por_detalle = (
		select subtotal
		from detalle_solicitud
		where	detalle = @detalle
	)

	if @subtotal_por_detalle is null
	begin
		set @subtotal_por_detalle = 0
	end

	return(round(@subtotal_por_detalle,2))
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_tipo_cambio    Script Date: 02/07/2008 04:36:32 p.m.
******/



/*
Inicia
Nombre:
fn_obtiene_tipo_cambio



Funcionalidad:
Convierte un monto en una moneda a otra moneda.

Entradas:
Moneda de origen (moneda del monto original)
Moneda destino (a la que se quiere convertir)
Fecha en la que se tomará el tipo de cambio
Monto a convertir

Proceso:
Busca en la tabla de tipo de cambio el tipo de cambio de la moneda origen con relación a la moneda destino.
Si las monedas son diferentes, se multiplica el monto por la relacion tipo de cambio. Si las monedas son
iguales entonces el tipo de cambio se considera como la unidad.

Salida:
Monto convertido

Ejemplo:
Termina
*/

CREATE function [dbo].fn_obtiene_tipo_cambio
(
        @idmoneda_origen int,
        @idmoneda_destino int,
        @fecha datetime,
        @monto_a_convertir money
)
returns money
as
begin
        declare
        @tipo_cambio as money,
        @ultima_fecha as datetime,
        @monto_resultado as money
```

```
	/*
	Si las monedas son diferentes
	*/
	if @idmoneda_origen <> @idmoneda_destino
		begin
			/*
			Se busca el tipo de cambio a la fecha proporcionada
			*/
			set @tipo_cambio = (
					select	tipo_cambio
					from	tipo_cambio
					where	idmoneda_origen = @idmoneda_origen AND
						idmoneda_destino = @idmoneda_destino AND
						fecha = @fecha
				)

			/*
			Si no existe tipo de cambio para la fecha proporcionada, se busca el ultimo
			tipo de cambio dado de alta
			*/
			if @tipo_cambio is null
			begin
				set @ultima_fecha = (
						select max(fecha)
						from	tipo_cambio
						where	idmoneda_origen = @idmoneda_origen
AND
							idmoneda_destino =
@idmoneda_destino
					)

				set @tipo_cambio = (
						select	isnull(tipo_cambio,0)
						from	tipo_cambio
						where	idmoneda_origen = @idmoneda_origen
AND
							idmoneda_destino =
@idmoneda_destino AND
							fecha = @ultima_fecha
					)
			end
		end
	/*
	Si las monedas son iguales, el tipo de cambio no existe y se asigna un valor = 1
	*/
	else
		begin

			set @tipo_cambio = 1
		end

	if @tipo_cambio is null
		begin

			set @tipo_cambio = 0
		end

	/*
	Se calcula el valor final convertido, es que el regresa la funcion
	*/
	set @monto_resultado = @monto_a_convertir * @tipo_cambio
```

```
	return(round(cast(@monto_resultado as money),2))
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_tipo_operacion    Script Date: 02/07/2008 04:36:31 p.m. ******/


/*
Inicia
Nombre:
fn_obtiene_tipo_operacion


Funcionalidad:
Obtiene el identificador del tipo de operacion por empresa

Entradas:
Empresa en la que se esta trabajando
Tipo de operacion (anticipo,pago,compra,etc)
```

```
Proceso:
Obtiene el identificador de ese tipo de operacion

Salida:
Id del tipo de operacion

Ejemplo:
select [dbo].fn_obtiene_tipo_operacion (12,'pago')
Termina
*/

CREATE function [dbo].fn_obtiene_tipo_operacion
(
	@idempresa int,
	@tipo_operacion varchar(50)
)
returns int
as
begin
	declare
	@idtipo_operacion int

	set @idtipo_operacion = (
		select idtipo_operacion
		from tipo_operacion
		where	idempresa = @idempresa and
			rtrim(descripcion) = rtrim(ltrim(@tipo_operacion))
	)

	if @idtipo_operacion is null
	begin
		set @idtipo_operacion = 0
	end

	return(@idtipo_operacion)

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.fn_obtiene_total_iva_por_detalle_pago    Script Date: 02/07/2008
04:36:34 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_total_iva_por_detalle_pago



Funcionalidad:
Obtiene cuánto se ha pagado de IVA para un detalle que esta relacionado con otro.

Entradas:
Folio del detalle principal

Proceso:
   ##########################################################################################

Salida:
Suma de IVAs pagados

Ejemplo:
select [dbo].fn_obtiene_total_iva_por_detalle_pago (12)
Termina
*/

CREATE function [dbo].fn_obtiene_total_iva_por_detalle_pago
(
	@detalle_principal int
)
returns money
as
begin
	declare
	@total_iva_por_detalle_pago money

	set @total_iva_por_detalle_pago = (
		select sum([dbo].fn_obtiene_iva_por_detalle(dc.detalle_relacionado))
		from detalle_principal_detalle_relacionado as dc
		where dc.detalle_relacionado in (
			select detalle_relacionado
			from detalle_principal_detalle_relacionado
			where detalle_principal = @detalle_principal
			)
	)

	return(round(@total_iva_por_detalle_pago,2))
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_total_no_viajes_por_detalle_pago    Script Date:
02/07/2008 04:36:35 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_total_no_viajes_por_detalle_pago



Funcionalidad:
Obtiene cuántos viajes de un detalle principal no se han relacionado.

Entradas:
Folio del detalle principal
Numero de empresa en la que se esta trabajando

Proceso:
   ##########################################################################################

Salida:
Número de viajes relacionados

Ejemplo:
select [dbo].fn_obtiene_total_no_viajes_por_detalle_pago (5,1)
Termina
*/

CREATE function [dbo].fn_obtiene_total_no_viajes_por_detalle_pago
(
        @detalle_principal int,
        @idempresa int
```

```
)
returns float
as
begin
	declare
	@total_no_viajes_por_detalle_pago money

	set @total_no_viajes_por_detalle_pago = (
		select
sum([dbo].fn_detalle_obtiene_no_viajes_por_detalle(@idempresa,dc.detalle_relacionado))
		from detalle_principal_detalle_relacionado as dc
		where dc.detalle_relacionado in (
			select detalle_relacionado
			from detalle_principal_detalle_relacionado
			where detalle_principal = @detalle_principal
			)
	)

	return(round(@total_no_viajes_por_detalle_pago,2))
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_total_por_detalle    Script Date: 02/07/2008 04:36:33
p.m. ******/









/*
Inicia
Nombre:
fn_obtiene_total_por_detalle
```

```
Funcionalidad:
Obtiene el total capturado para un detalle

Entradas:
Numero de detalle del que se quiere obtener el total

Proceso:
Obtiene el total para un detalle

Salida:
Total

Ejemplo:
select [dbo].fn_obtiene_total_por_detalle (1)
Termina
*/

CREATE function [dbo].fn_obtiene_total_por_detalle
(
	@detalle int
)
returns money
as
begin
	declare
	@total_por_detalle money

	set @total_por_detalle = (
		select total
		from detalle_solicitud
		where	detalle = @detalle
	)

	if @total_por_detalle is null
	begin
		set @total_por_detalle = 0
	end

	return(round(@total_por_detalle,2))
end




GO
SET QUOTED_IDENTIFIER OFF
GO
```

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_total_por_folio    Script Date: 02/07/2008 04:36:33 p.m.
******/




/*
Inicia
Nombre:
fn_obtiene_total_por_folio



Funcionalidad:
Obtiene el total capturado para un folio

Entradas:
Numero de folio del que se quiere obtener el total

Proceso:
Obtiene el total para un folio

Salida:
Suma Total

Ejemplo:
select [dbo].fn_obtiene_total_por_folio (1)
Termina
*/

CREATE function [dbo].fn_obtiene_total_por_folio
(
	@folio int
)
returns money
as
begin
	declare
	@total_por_folio money

	set @total_por_folio = (
		select sum(total)
		from detalle_solicitud
		where	folio = @folio
	)

	if @total_por_folio is null
	begin
		set @total_por_folio = 0
	end
```

```
        return(round(@total_por_folio,2))
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_total_propina_por_detalle_pago    Script Date:
02/07/2008 04:36:34 p.m. ******/



/*
Inicia
Nombre:
fn_obtiene_total_propina_por_detalle_pago


Funcionalidad:
Obtiene cuánto se ha pagado de propina para un detalle que esta relacionado con otro.

Entradas:
Folio del detalle principal

Proceso:
   ##########################################################################################

Salida:
Suma de propinas pagadas

Ejemplo:
```

```
select [dbo].fn_obtiene_total_propina_por_detalle_pago (12)
Termina
*/

CREATE function [dbo].fn_obtiene_total_propina_por_detalle_pago
(
	@detalle_principal int
)
returns money
as
begin
	declare
	@total_propina_por_detalle_pago money

	set @total_propina_por_detalle_pago = (
		select sum([dbo].fn_obtiene_propina_por_detalle(dc.detalle_relacionado))
		from detalle_principal_detalle_relacionado as dc
		where dc.detalle_relacionado in (
			select detalle_relacionado
			from detalle_principal_detalle_relacionado
			where detalle_principal = @detalle_principal
			)
	)

	return(round(@total_propina_por_detalle_pago,2))
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_total_subtotal_por_detalle_pago    Script Date:
02/07/2008 04:36:34 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_obtiene_total_subtotal_por_detalle_pago



Funcionalidad:
Obtiene cuánto se ha pagado de subtotal para un detalle que esta relacionado con otro.

Entradas:
Folio del detalle principal

Proceso:
   ##########################################################################################

Salida:
Suma de subtotales pagados

Ejemplo:
select [dbo].fn_obtiene_total_subtotal_por_detalle_pago (12)
Termina
*/

CREATE function [dbo].fn_obtiene_total_subtotal_por_detalle_pago
(
	@detalle_principal int
)
returns money
as
begin
	declare
	@total_subtotal_por_detalle_pago money

	set @total_subtotal_por_detalle_pago = (
		select sum([dbo].fn_obtiene_subtotal_por_detalle(dc.detalle_relacionado))
		from detalle_principal_detalle_relacionado as dc
		where dc.detalle_relacionado in (
			select detalle_relacionado
			from detalle_principal_detalle_relacionado
			where detalle_principal = @detalle_principal
			)
	)

	return(round(@total_subtotal_por_detalle_pago,2))
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_total_total_por_detalle_pago    Script Date: 02/07/2008
04:36:34 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_total_total_por_detalle_pago



Funcionalidad:
Obtiene cuánto se ha pagado en total para un detalle que esta relacionado con otro.

Entradas:
Folio del detalle principal

Proceso:
   ###########################################################################################

Salida:
Suma de totales pagados

Ejemplo:
select [dbo].fn_obtiene_total_total_por_detalle_pago (12)
Termina
*/
CREATE function [dbo].fn_obtiene_total_total_por_detalle_pago
(
        @detalle_principal int
)
returns money
as
begin
        declare
        @total_total_por_detalle_pago money

        set @total_total_por_detalle_pago = (
                select sum([dbo].fn_obtiene_total_por_detalle(dc.detalle_relacionado))
                from detalle_principal_detalle_relacionado as dc
                where dc.detalle_relacionado in (
                        select detalle_relacionado
                        from detalle_principal_detalle_relacionado
                        where detalle_principal = @detalle_principal
```

```
                        )
        )

        return(round(@total_total_por_detalle_pago,2))
end









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_total_tua_por_detalle_pago    Script Date: 02/07/2008
04:36:34 p.m. ******/




/*
Inicia
Nombre:
fn_obtiene_total_tua_por_detalle_pago



Funcionalidad:
Obtiene cuánto se ha pagado en TUA para un detalle que esta relacionado con otro.

Entradas:
Folio del detalle principal

Proceso:
   ##########################################################################################

Salida:
Suma de TUAs pagados

Ejemplo:
```

```
select [dbo].fn_obtiene_total_tua_por_detalle_pago (12)
Termina
*/

CREATE function [dbo].fn_obtiene_total_tua_por_detalle_pago
(
	@detalle_principal int
)
returns money
as
begin
	declare
	@total_tua_por_detalle_pago money

	set @total_tua_por_detalle_pago = (
		select sum([dbo].fn_obtiene_tua_por_detalle(dc.detalle_relacionado))
		from detalle_principal_detalle_relacionado as dc
		where dc.detalle_relacionado in (
			select detalle_relacionado
			from detalle_principal_detalle_relacionado
			where detalle_principal = @detalle_principal
			)
	)

	return(round(@total_tua_por_detalle_pago,2))
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_obtiene_tua_por_detalle    Script Date: 02/07/2008 04:36:33 p.m.
******/
```

```
/*
Inicia
Nombre:
fn_obtiene_tua_por_detalle



Funcionalidad:
Obtiene el TUA capturado para un detalle

Entradas:
Numero de detalle del que se quiere obtener el TUA

Proceso:
Obtiene el TUA para un detalle

Salida:
TUA

Ejemplo:
select [dbo].fn_obtiene_tua_por_detalle (1)
Termina
*/

CREATE  function [dbo].fn_obtiene_tua_por_detalle
(
	@detalle int
)
returns money
as
begin
	declare
	@tua_por_detalle money

	set @tua_por_detalle = (
		select tua
		from detalle_solicitud
		where	detalle = @detalle
	)

	if @tua_por_detalle is null
	begin
		set @tua_por_detalle = 0
	end

	return(round(@tua_por_detalle,2))
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_obtiene_ultimo_status_anterior    Script Date: 02/07/2008
04:36:32 p.m. ******/



/*
Inicia
Nombre:
fn_obtiene_ultimo_status_anterior



Funcionalidad:
Obtiene cuál es el status anterior a un status en el flujo de autorizacion.

Entradas:
Empresa en la que se trabaja
Identificador del tipo de operación
Status del que se quiere obtener el status anterior

Proceso:
Para el flujo de autorización de la empresa en cuestión, obtiene el status anterior al status dado.

Salida:
Status anterior al status dado, si se provee del primer status, entonces la salida es el primer status, ya que
este no tiene status anterior.

Ejemplo:
select [dbo].fn_obtiene_ultimo_status_anterior (1,1,'ep')
Termina
*/

CREATE function [dbo].fn_obtiene_ultimo_status_anterior
(
        @idempresa int,
        @idtipo_operacion int,
        @idstatus char(2)
)
returns char(2)
as
begin
        declare
        @idstatus_anterior char(2)

        set @idstatus_anterior = (
```

```
                select  idstatus
                from    flujo_autoriza
                where   idempresa = @idempresa and
                        idstatus_sig = @idstatus and
                        idtipo_operacion = @idtipo_operacion
        )

        if @idstatus_anterior is null or @idstatus_anterior = ''
        begin
                set @idstatus_anterior = 'nu'
        end

        return(@idstatus_anterior)

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_ppto_obtiene_folio_en_ppto_detalle    Script Date: 02/07/2008
04:36:34 p.m. ******/




/*
Inicia
Nombre:
fn_ppto_obtiene_folio_en_ppto_detalle



Funcionalidad:
Revisa si ya hay prespuesto comprometido o pagado para una solicitud.

Entradas:
Folio de la solicitud de la que se quiere saber si ya hay presupuesto comprometido o pagado

Proceso:
Busca en la tabla donde se almacenan todos los detalles de las solicitudes cuyo monto ya se calculó sobre el
presupuesto (ppto_detalle) y se verifica si algun detalle de la solicitud existe en dicha tabla.
```

```
Salida:
0 = No se ha comprometido ppto para esta solicitud
1 = Se ha comprometido ppto para esta solicitud

Ejemplo:
select [dbo].fn_ppto_obtiene_folio_en_ppto_detalle (1)
Termina
*/

CREATE function [dbo].fn_ppto_obtiene_folio_en_ppto_detalle
(
	@folio int
)
returns tinyint
as
begin
	declare
	@existe_en_ppto tinyint

	set @existe_en_ppto = (
		select top 1 '1'
		from detalle_solicitud as de
		inner join ppto_detalle as pp on de.detalle = pp.detalle
		where de.folio = @folio
	)

	if @existe_en_ppto is null
	begin
		set @existe_en_ppto = 0
	end
	else
	begin
		set @existe_en_ppto = 1
	end

	return(@existe_en_ppto)

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.fn_revisa_ccostos_autoriza    Script Date: 02/07/2008 04:36:33 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_revisa_ccostos_autoriza


Funcionalidad:
Revisa si un centro de costo tiene dueño (autorizador de ppto)

Entradas:
Empresa en la que se esta trabajando
Centro de costo

Proceso:
Se revisa si el centro de costo al que se pretende distribuir la solicitud tiene o no centro de costo

Salida:
Caracter de verificacion
	(1 = si tiene dueno)
	(0 = no tiene dueno)

Ejemplo:
select [dbo].fn_revisa_ccostos_autoriza(1,2)

Termina
*/

CREATE function [dbo].fn_revisa_ccostos_autoriza
(
	@idempresa int,
	@ccostos int
)
returns tinyint
begin
	declare
	@tiene_dueno int

	set @tiene_dueno = (
		select isnull(numempleado,0)
		from centro_costo_autoriza
		where	idccostos = @ccostos and
			idempresa = @idempresa
	)

	if @tiene_dueno is not null
	begin
		set @tiene_dueno = 1
	end
	else
	begin
		set @tiene_dueno = 0
	end
```

```
	return(@tiene_dueno)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_revisa_excede_distribucion_por_detalle    Script Date:
02/07/2008 04:36:34 p.m. ******/




/*
Inicia
Nombre:
fn_revisa_excede_distribucion_por_detalle


Funcionalidad:
Revisa si un nuevo porcentaje de ditribucion no hace que el porcentaje rebase el 100%

Entradas:
Numero del detalle que se vaya a distribuir
```

Porcentaje a distribuir

Proceso:
Obtiene si el nuevo porcentaje no hace que la distribucion exceda el 100%

Salida:
Caracter de verificacion
	(1 = si excede)
	(0 = no excede)

Ejemplo:
select [dbo].fn_revisa_excede_distribucion_por_detalle(1,123)
Termina
*/

```
CREATE function [dbo].fn_revisa_excede_distribucion_por_detalle
(
	@detalle int,
	@porcentaje float
)
returns tinyint
begin
	declare
	@existe float,
	@total_distribucion float,
	@resultado tinyint

	set @existe = (
		select
		isnull(sum(di.porcentaje),0)
		from distribucion_solicitud as di
		where di.detalle = @detalle
	)

	set @total_distribucion = @porcentaje + @existe

	if @total_distribucion > 100.00
	begin
		set @resultado = 1
	end
	else
	begin
		set @resultado = 0
	end


	return(@resultado)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.fn_revisa_saldos_en_contra    Script Date: 02/07/2008 04:36:35
p.m. ******/



/*
Inicia
Nombre:
fn_revisa_saldos_en_contra


Funcionalidad:
Obtiene el saldo negativo de una relación de dos solicitudes.

Entradas:
El folio de la solicitud principal (pe. Anticipo)
El folio de la solicitud relacionada (pe. Comprobación de anticipo)
El tipo de operación de la solicitud relacionada. Por el momento no se utiliza este parámetro, pero se dejó en
caso de que se quieran evaluar saldos negativos de otros tipos de operación en la posteridad. Ahorita solo se
utiliza para los anticipos y su comprobación.

Proceso:
Obtiene el monto de la solicitud principal, el monto de la solicitud relacionada y luego realiza una operación
matemática pa

Salida:
Saldo o diferencia entre solicitudes

Ejemplo:
select [dbo].fn_revisa_saldos_en_contra(1,2,'')
Termina
*/

CREATE function [dbo].fn_revisa_saldos_en_contra
```

```
(
	@folio_principal int,
	@folio_relacionado int,
	@tipo_operacion varchar(50)
)
returns money
as
begin
	declare
	@saldo_en_contra money,
	@total_folio_principal money,
	@total_folio_relacionado money

	set @total_folio_principal = (
		select
		[dbo].fn_obtiene_total_por_folio(@folio_principal)
	)

	set @total_folio_relacionado = (
		select
		[dbo].fn_obtiene_total_por_folio(@folio_relacionado)
	)

	set @saldo_en_contra = (
		@total_folio_relacionado
		-
		@total_folio_principal
	)

	return(round(@saldo_en_contra,2))


end







GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  User Defined Function dbo.fn_solicitud_obtiene_folio_consecutivo    Script Date: 02/07/2008
04:36:33 p.m. ******/
```

```
/*
Inicia
Nombre:
fn_solicitud_obtiene_folio_consecutivo



Funcionalidad:
Obtiene el siguiente folio de solicitud

Entradas:
No hay entradas

Proceso:
Obtiene el numero siguiente de la solicitud

Salida:
Numero siguiente de la solicitud

Ejemplo:
select [dbo].fn_solicitud_obtiene_folio_consecutivo()
Termina
*/

CREATE function [dbo].fn_solicitud_obtiene_folio_consecutivo
()
returns int
as
begin
	declare
	@consecutivo as int

	set @consecutivo = (
		select top 1 folio
		from
			(
			select isnull(max(folio),0) + 1 as folio from solicitud
			union
			select isnull(max(folio),0) + 1 as folio from contrato
			)
		temporal
		order by folio desc
	)

	return(@consecutivo)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.sp_convierte_fecha_a_texto    Script Date: 02/07/2008 04:36:31
p.m. ******/




/*
Inicia
Nombre:
sp_convierte_fecha_a_texto



Funcionalidad:
Convierte una fecha con formato '02/02/1979' a formato '02/feb/1979'

Entradas:
Fecha a convertir

Proceso:
Obtiene la fecha en diferente formato

Salida:
Cadena de texto de fecha

Ejemplo:
select [dbo].sp_convierte_fecha_a_texto('02/02/1979')

Termina
*/

CREATE function [dbo].sp_convierte_fecha_a_texto
(
	@fecha_texto varchar(11)
)
returns varchar(11)
as
begin
	declare
	@fecha_nueva varchar(11),
	@empieza_dia int,
	@termina_dia int,
	@empieza_mes int,
	@termina_mes int,
	@empieza_ano int,
```

```
        @termina_ano int,
        @valor_dia char(2),
        @valor_mes char(3),
        @valor_nuevo_mes char(3),
        @valor_ano char(4)

        set @empieza_dia = 1
        set @termina_dia = 2
        set @empieza_mes = 4
        set @termina_mes = 2
        set @empieza_ano = 7
        set @termina_ano = 4


        set @valor_dia = substring(@fecha_texto,@empieza_dia,@termina_dia)
        set @valor_mes = substring(@fecha_texto,@empieza_mes,@termina_mes)
        set @valor_ano = substring(@fecha_texto,@empieza_ano,@termina_ano)

        if @valor_mes = '01' set @valor_nuevo_mes = 'ene'
        if @valor_mes = '02' set @valor_nuevo_mes = 'feb'
        if @valor_mes = '03' set @valor_nuevo_mes = 'mar'
        if @valor_mes = '04' set @valor_nuevo_mes = 'abr'
        if @valor_mes = '05' set @valor_nuevo_mes = 'may'
        if @valor_mes = '06' set @valor_nuevo_mes = 'jun'
        if @valor_mes = '07' set @valor_nuevo_mes = 'jul'
        if @valor_mes = '08' set @valor_nuevo_mes = 'ago'
        if @valor_mes = '09' set @valor_nuevo_mes = 'sep'
        if @valor_mes = '10' set @valor_nuevo_mes = 'oct'
        if @valor_mes = '11' set @valor_nuevo_mes = 'nov'
        if @valor_mes = '12' set @valor_nuevo_mes = 'dic'

        set @fecha_nueva = rtrim(@valor_dia) +'/'+@valor_nuevo_mes+'/'+@valor_ano

        return(@fecha_nueva)
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  User Defined Function dbo.sp_convierte_texto_a_fecha    Script Date: 02/07/2008 04:36:31
p.m. ******/




/*
Inicia
Nombre:
sp_convierte_texto_a_fecha



Funcionalidad:
Convierte una fecha con formato '02/feb/1979' a formato '02/02/1979'

Entradas:
Fecha a convertir

Proceso:
Obtiene la fecha en diferente formato

Salida:
Cadena de texto de fecha

Ejemplo:
select [dbo].sp_convierte_texto_a_fecha('7/dic/1979')
Termina
*/

CREATE function [dbo].sp_convierte_texto_a_fecha
(
	@fecha_texto varchar(11)
)
returns varchar(11)
as
begin
	declare
	@fecha_nueva varchar(11),
	@empieza_dia int,
	@termina_dia int,
	@empieza_mes int,
	@termina_mes int,
	@empieza_ano int,
	@termina_ano int,
	@valor_dia char(2),
	@valor_mes char(3),
	@valor_nuevo_mes char(2),
	@valor_ano char(4)

	if len(@fecha_texto) = 11
	begin
		set @empieza_dia = 1
		set @termina_dia = 2
		set @empieza_mes = 4
		set @termina_mes = 3
		set @empieza_ano = 8
		set @termina_ano = 4
	end
```

```
	if len(@fecha_texto) = 10
	begin
		set @empieza_dia = 1
		set @termina_dia = 1
		set @empieza_mes = 3
		set @termina_mes = 4
		set @empieza_ano = 7
		set @termina_ano = 4
	end

	set @valor_dia = substring(@fecha_texto,@empieza_dia,@termina_dia)
	set @valor_mes = substring(@fecha_texto,@empieza_mes,@termina_mes)
	set @valor_ano = substring(@fecha_texto,@empieza_ano,@termina_ano)

	if @valor_mes = 'ene' set @valor_nuevo_mes = '01'
	if @valor_mes = 'feb' set @valor_nuevo_mes = '02'
	if @valor_mes = 'mar' set @valor_nuevo_mes = '03'
	if @valor_mes = 'abr' set @valor_nuevo_mes = '04'
	if @valor_mes = 'may' set @valor_nuevo_mes = '05'
	if @valor_mes = 'jun' set @valor_nuevo_mes = '06'
	if @valor_mes = 'jul' set @valor_nuevo_mes = '07'
	if @valor_mes = 'ago' set @valor_nuevo_mes = '08'
	if @valor_mes = 'sep' set @valor_nuevo_mes = '09'
	if @valor_mes = 'oct' set @valor_nuevo_mes = '10'
	if @valor_mes = 'nov' set @valor_nuevo_mes = '11'
	if @valor_mes = 'dic' set @valor_nuevo_mes = '12'



	if @valor_dia < 10 and len(@fecha_texto) = 11
	begin
		set @valor_dia = @valor_dia
	end

	if @valor_dia < 10 and len(@fecha_texto) = 10
	begin
		set @valor_dia = '0' + @valor_dia
	end

	set @fecha_nueva = rtrim(@valor_dia) +'/'+@valor_nuevo_mes+'/'+@valor_ano

	if @fecha_nueva is null
	begin
		set @fecha_nueva = @fecha_texto
	end

	return(@fecha_nueva)
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  User Defined Function dbo.user_pwd    Script Date: 02/07/2008 04:36:31 p.m. ******/



/*
Inicia
Nombre:
user_pwd


Funcionalidad:
Encripta o desencripta una cadena de 10 caracteres. Esta función se utiliza para almacenar y modificar la
clave de acceso de

Entradas:
Cadena de caracteres
Accion a seguir
        1 = Encripta la cadena
        2 = Desencripta la cadena

Proceso:
Obtiene la cadena de caracteres dada y la encripta o desencripta según se especifique.

Salida:
Cadena de caracteres encriptada o desencriptada

Ejemplo:
Para encriptar select [dbo].[user_pwd]('cevall1S',1)
Para desencriptar select [dbo].[user_pwd]('-ºÂª"1/2†œur',2)
Termina
*/


/*
select [dbo].[user_pwd]('cevall1S',1)
select [dbo].[user_pwd]('-ºÂª"1/2†œur',2)
*/

CREATE function [dbo].[user_pwd]
(
        @cadena nvarchar(10),
```

```
	@accion int
)
returns nvarchar(10)
as
begin
	declare
	@c1 char(1),
	@c2 char(1),
	@c3 char(1),
	@c4 char(1),
	@c5 char(1),
	@c6 char(1),
	@c7 char(1),
	@c8 char(1),
	@c9 char(1),
	@c10 char(1),
	@clave char(10),
	@resultado nvarchar(10)

	Set @clave =
char(74)+char(85)+char(76)+char(73)+char(79)+char(81)+char(85)+char(73)+char(82)+char(79)

	/*
	Esta accion encripta
	*/
	If (@accion=1)
	Begin
		If (LEN(@cadena)<10)
		Begin
			While LEN(@cadena) <= 10
			Begin
				Set @cadena = @cadena + char(35)
				If LEN(@cadena) = 10
				Break
				Else
					Continue
			End
		End

		Set @c1 = char(ASCII(SUBSTRING(@cadena,1,1)) + ASCII(SUBSTRING(@clave,1,1)))
		Set @c2 = char(ASCII(SUBSTRING(@cadena,2,1)) + ASCII(SUBSTRING(@clave,2,1)))
		Set @c3 = char(ASCII(SUBSTRING(@cadena,3,1)) + ASCII(SUBSTRING(@clave,3,1)))
		Set @c4 = char(ASCII(SUBSTRING(@cadena,4,1)) + ASCII(SUBSTRING(@clave,4,1)))
		Set @c5 = char(ASCII(SUBSTRING(@cadena,5,1)) + ASCII(SUBSTRING(@clave,5,1)))
		Set @c6 = char(ASCII(SUBSTRING(@cadena,6,1)) + ASCII(SUBSTRING(@clave,6,1)))
		Set @c7 = char(ASCII(SUBSTRING(@cadena,7,1)) + ASCII(SUBSTRING(@clave,7,1)))
		Set @c8 = char(ASCII(SUBSTRING(@cadena,8,1)) + ASCII(SUBSTRING(@clave,8,1)))
		Set @c9 = char(ASCII(SUBSTRING(@cadena,9,1)) + ASCII(SUBSTRING(@clave,9,1)))
		Set @c10 = char(ASCII(SUBSTRING(@cadena,10,1)) + ASCII(SUBSTRING(@clave,10,1)))
		Set @resultado = @c1 + @c2 + @c3 + @c4 + @c5 + @c6 + @c7 + @c8 + @c9 + @c10
	End

	/*
	Esta accion desencripta
	*/
	If (@accion=2)
	Begin
		SET @c1 = char(ASCII(SUBSTRING(@cadena,1,1)) - ASCII(SUBSTRING(@clave,1,1)))
		SET @c2 = char(ASCII(SUBSTRING(@cadena,2,1)) - ASCII(SUBSTRING(@clave,2,1)))
		SET @c3 = char(ASCII(SUBSTRING(@cadena,3,1)) - ASCII(SUBSTRING(@clave,3,1)))
		SET @c4 = char(ASCII(SUBSTRING(@cadena,4,1)) - ASCII(SUBSTRING(@clave,4,1)))
		SET @c5 = char(ASCII(SUBSTRING(@cadena,5,1)) - ASCII(SUBSTRING(@clave,5,1)))
```

```
                SET @c6 = char(ASCII(SUBSTRING(@cadena,6,1)) - ASCII(SUBSTRING(@clave,6,1)))
                SET @c7 = char(ASCII(SUBSTRING(@cadena,7,1)) - ASCII(SUBSTRING(@clave,7,1)))
                SET @c8 = char(ASCII(SUBSTRING(@cadena,8,1)) - ASCII(SUBSTRING(@clave,8,1)))
                SET @c9 = char(ASCII(SUBSTRING(@cadena,9,1)) - ASCII(SUBSTRING(@clave,9,1)))
                SET @c10 = char(ASCII(SUBSTRING(@cadena,10,1)) - ASCII(SUBSTRING(@clave,10,1)))
                SET @resultado = @c1 + @c2 + @c3 + @c4 + @c5 + @c6 + @c7 + @c8 + @c9 + @c10

                Set @resultado = REPLACE(@resultado,char(35),'')
        End

        Return(@resultado)

End
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

CREATE TABLE [dbo].[anticipo] (
        [folio] [int] NOT NULL ,
        [fecha_ini] [datetime] NOT NULL ,
        [fecha_fin] [datetime] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[area_responsabilidad] (
        [idarea] [int] NOT NULL ,
        [idempresa] [int] NOT NULL ,
        [descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
        [idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[area_responsabilidad_autoriza] (
        [idarea] [int] NOT NULL ,
        [idempresa] [int] NOT NULL ,
        [numempleado] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[banco] (
```

```
	[idbanco] [int] NOT NULL ,
	[nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[nac_ext] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cuenta_nac] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cuenta_ext] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[identificador] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[banco_importar] (
	[idbanco] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[nac_ext] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cuenta_nac] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cuenta_ext] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[status] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[carga] (
	[llave_carga] [int] NOT NULL ,
	[date_time_started] [datetime] NULL ,
	[date_time_ended] [datetime] NULL ,
	[load_description] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[status] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[indicador_carg_estr] [tinyint] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[cdosysmail_failures] (
	[Date of Failure] [datetime] NULL ,
	[Spid] [int] NULL ,
	[From] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[To] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[Subject] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[Body] [varchar] (4000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[iMsg] [int] NULL ,
	[Hr] [int] NULL ,
	[Source of Failure] [varchar] (255) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[Description of Failure] [varchar] (500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[Output from Failure] [varchar] (1000) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[Comment about Failure] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[centro_costo] (
	[idccostos] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[identificador] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[llave_carga] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[centro_costo_autoriza] (
	[idccostos] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
```

```
	[numempleado] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[centro_costo_importar] (
	[ccostos] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[empresa] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[status] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[concepto_gasto] (
	[idconcepto] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[porcentaje_deducible] [numeric](8, 4) NULL ,
	[ccontable_deducible] [char] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[porcentaje_no_deducible] [numeric](8, 4) NULL ,
	[ccontable_no_deducible] [char] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[concepto_gasto_area] (
	[idempresa] [int] NOT NULL ,
	[idconcepto] [int] NOT NULL ,
	[idarea] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[concepto_gasto_asiento_contable] (
	[idempresa] [int] NOT NULL ,
	[idconcepto] [int] NOT NULL ,
	[idasiento] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[concepto_gasto_tipo_concepto] (
	[idempresa] [int] NOT NULL ,
	[idconcepto] [int] NOT NULL ,
	[idtipo_concepto] [int] NOT NULL ,
	[idtipo_operacion] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[contabilidad_autoriza] (
	[idempresa] [int] NOT NULL ,
	[numempleado] [int] NOT NULL ,
	[rol] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[costo_por_km] (
	[idempresa] [int] NOT NULL ,
	[idmoneda] [int] NOT NULL ,
	[costo_por_km] [money] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[cuenta_contable] (
	[idempresa] [int] NOT NULL ,
	[ccontable] [char] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[cuenta_contable_importar] (
	[ccontable] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[empresa] [char] (4) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[status] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[deposito] (
	[folio] [int] NOT NULL ,
	[idempresa] [int] NULL ,
	[numempleado] [int] NULL ,
	[fecha] [datetime] NULL ,
	[idbanco] [int] NULL ,
	[referencia] [char] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[monto] [money] NULL ,
	[idmoneda] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[deposito_importar] (
	[folio] [int] NOT NULL ,
	[no_empresa] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[empleado] [int] NULL ,
	[fecha] [datetime] NULL ,
	[idbanco] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[referencia] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[monto] [money] NULL ,
	[idmoneda] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_anexo] (
	[idanexo] [int] NOT NULL ,
	[detalle] [int] NOT NULL ,
	[nombre] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[tamano] [decimal](15, 2) NOT NULL ,
	[tipo] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[fecha_creacion] [datetime] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_boleto_avion] (
	[detalle] [int] NOT NULL ,
	[origen] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[destino] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[aerolinea] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
```

```
	[hora_salida] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_comensales] (
	[detalle] [int] NOT NULL ,
	[comensales] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_deposito] (
	[detalle] [int] NOT NULL ,
	[folio_deposito] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_dias] (
	[detalle] [int] NOT NULL ,
	[dias] [float] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_kilometro] (
	[detalle] [int] NOT NULL ,
	[idkm] [int] NOT NULL ,
	[num_viajes] [float] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_precio_unitario_unidades] (
	[detalle] [int] NOT NULL ,
	[precio_unitario] [money] NULL ,
	[unidades] [float] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_principal_detalle_relacionado] (
	[detalle_principal] [int] NOT NULL ,
	[detalle_relacionado] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[detalle_solicitud] (
	[detalle] [int] NOT NULL ,
	[folio] [int] NOT NULL ,
	[idempresa] [int] NULL ,
	[idconcepto] [int] NULL ,
	[comentario] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_gasto] [datetime] NULL ,
	[subtotal] [money] NULL ,
	[iva] [money] NULL ,
	[idiva] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[propina] [money] NULL ,
	[tua] [money] NULL ,
	[total] [money] NULL ,
	[tarjeta_corporativa] [tinyint] NULL ,
	[inc_pol] [tinyint] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[distribucion_solicitud] (
	[distribucion] [int] NOT NULL ,
```

```
	[detalle] [int] NOT NULL ,
	[folio] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[ccostos] [int] NOT NULL ,
	[porcentaje] [float] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[empresa] (
	[idempresa] [int] NOT NULL ,
	[Nombre] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[RFC] [varchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[direccion] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[colonia] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[delegacionmunicip] [varchar] (40) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[estado] [varchar] (40) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[pais] [varchar] (40) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cp] [varchar] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[telefono] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fax] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[mail] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[logo] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[no_empresa] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[empresa_importar] (
	[no_empresa] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[Nombre] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[RFC] [varchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[direccion] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[colonia] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[delegacionmunicip] [varchar] (40) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[estado] [varchar] (40) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[pais] [varchar] (40) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cp] [varchar] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[telefono] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fax] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[mail] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idstatus] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[flujo_autoriza] (
	[idflujo] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[idtipo_operacion] [int] NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[idstatus_sig] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[flujo_autoriza_opcion] (
	[idempresa] [int] NOT NULL ,
	[flujo_masivo] [tinyint] NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[folio_principal_folio_relacionado] (
	[folio_principal] [int] NOT NULL ,
	[folio_relacionado] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[historial_solicitud] (
	[idhistorial] [int] NOT NULL ,
	[folio] [int] NOT NULL ,
	[numempleado] [int] NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[fecha] [datetime] NOT NULL ,
	[observacion] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[iva] (
	[identificador] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[porcentaje] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[iva_importar] (
	[identificador] [char] (4) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[porcentaje] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[kilometro] (
	[idkm] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[origen] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[destino] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[distancia] [int] NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[mensaje_status] (
	[idempresa] [int] NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[mensaje] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[moneda] (
	[idmoneda] [int] NOT NULL ,
	[identificador] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[moneda_importar] (
	[identificador] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
```

```
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[monto_limite] (
	[idmonto_limite] [int] NOT NULL ,
	[idnivel] [int] NOT NULL ,
	[idconcepto] [int] NOT NULL ,
	[idtipo_operacion] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[idmoneda] [int] NOT NULL ,
	[idterritorio] [int] NOT NULL ,
	[monto_limite] [money] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[nivel] (
	[idnivel] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[object_control] (
	[object_key] [int] NOT NULL ,
	[object_name] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[last_key_assigned] [int] NULL ,
	[locked_status_flag] [tinyint] NULL ,
	[last_locked_by] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[last_locked_at] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[objeto] (
	[idobjeto] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[idubicacion] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[objeto_ubicacion] (
	[idubicacion] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[operacion_proceso] (
	[idempresa] [int] NOT NULL ,
	[idperfil] [int] NOT NULL ,
	[idtipo_operacion] [int] NOT NULL ,
	[op_proceso] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[pago] (
	[folio] [int] NOT NULL ,
	[idproveedor] [int] NULL ,
	[no_factura] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
```

```
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[perfil] (
	[idperfil] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[perfil_objeto] (
	[idempresa] [int] NOT NULL ,
	[idperfil] [int] NOT NULL ,
	[idobjeto] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[perfil_tipo_autorizacion] (
	[idempresa] [int] NOT NULL ,
	[idperfil] [int] NOT NULL ,
	[tipo_autorizacion] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[ppto_detalle] (
	[idreg] [int] NOT NULL ,
	[idppto] [int] NOT NULL ,
	[detalle] [int] NOT NULL ,
	[monto] [money] NOT NULL ,
	[tipo] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[fecha] [datetime] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[presupuesto] (
	[idppto] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[ccontable] [char] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[anio] [int] NOT NULL ,
	[mes] [int] NOT NULL ,
	[ppto_presupuestado] [money] NULL ,
	[ppto_gastado] [money] NULL ,
	[ppto_comprometido] [money] NULL ,
	[idmoneda] [int] NOT NULL ,
	[identidad] [int] NOT NULL ,
	[llave_carga] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[presupuesto_importar] (
	[idregistro] [int] NOT NULL ,
	[empresa] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[ccontable] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[anio] [int] NULL ,
	[mes] [int] NULL ,
	[ppto_presupuestado] [money] NULL ,
	[ppto_gastado] [money] NULL ,
	[ppto_comprometido] [money] NULL ,
	[idmoneda] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[centro_costo] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
```

```
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[proveedor] (
	[idproveedor] [int] NOT NULL ,
	[nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[rfc] [varchar] (13) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[calle] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[numero] [int] NULL ,
	[interior] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[colonia] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[delegacion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cp] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[estado] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[pais] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[telefono] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fax] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[mail] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[serv_producto] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[observaciones] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idbanco] [int] NULL ,
	[no_cuenta] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idtipo_pago] [int] NULL ,
	[identificador] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[proveedor_concepto] (
	[idempresa] [int] NOT NULL ,
	[idproveedor] [int] NOT NULL ,
	[idconcepto] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[proveedor_empresa] (
	[idproveedor] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[proveedor_empresa_importar] (
	[no_empresa] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[proveedor] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[status] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[proveedor_importar] (
	[identificador] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[rfc] [varchar] (13) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[calle] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[numero] [int] NULL ,
	[interior] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[colonia] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[delegacion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cp] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
```

```
	[estado] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[pais] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[telefono] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fax] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[mail] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[serv_producto] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[observaciones] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[banco] [varchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[no_cuenta] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[tipo_pago] [int] NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[solicitud] (
	[idtipo_operacion] [int] NOT NULL ,
	[folio] [int] NOT NULL ,
	[idempresa] [int] NOT NULL ,
	[numempleado] [int] NOT NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[idmoneda] [int] NOT NULL ,
	[idmoneda_pago] [int] NOT NULL ,
	[idterritorio] [int] NOT NULL ,
	[fecha_solicitud] [datetime] NOT NULL ,
	[descripcion] [varchar] (250) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[status] (
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[descripcion_1] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[descripcion_2] [varchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[status_rechazo] (
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[idstatus_rechazo] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[territorio] (
	[idterritorio] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tipo_cambio] (
	[idmoneda_origen] [int] NOT NULL ,
	[idmoneda_destino] [int] NOT NULL ,
	[fecha] [datetime] NOT NULL ,
	[tipo_cambio] [money] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tipo_cambio_importar] (
	[idmoneda_origen] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[idmoneda_destino] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[fecha] [datetime] NOT NULL ,
	[tipo_cambio] [money] NOT NULL ,
```

```
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tipo_concepto] (
	[idempresa] [int] NOT NULL ,
	[idtipo_concepto] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[boleto_avion] [tinyint] NULL ,
	[kilometraje] [tinyint] NULL ,
	[precio_unitario_cantidad] [int] NULL ,
	[iva] [tinyint] NULL ,
	[tua] [tinyint] NULL ,
	[propina] [tinyint] NULL ,
	[dias] [tinyint] NULL ,
	[comensales] [tinyint] NULL ,
	[deposito] [tinyint] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tipo_operacion] (
	[idempresa] [int] NOT NULL ,
	[idtipo_operacion] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tipo_pago] (
	[idtipo_pago] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tipo_pago_importar] (
	[idtipopag] [int] NOT NULL ,
	[nombre] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tipo_usuario] (
	[idtipo_usuario] [int] NOT NULL ,
	[descripcion] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[usuario] (
	[idempresa] [int] NOT NULL ,
	[numempleado] [int] NOT NULL ,
	[idtipo_usuario] [int] NULL ,
	[idbanco] [int] NULL ,
	[idnivel] [int] NULL ,
	[idperfil] [int] NULL ,
	[ccostos] [int] NULL ,
	[idstatus] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[paterno] [varchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
```

```
	[materno] [varchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[nombre] [varchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[rfc] [varchar] (13) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[puesto] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[cta_cheques] [varchar] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[clave_acceso] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[tarjeta_corporativa] [tinyint] NULL ,
	[email] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[num_jefe] [int] NULL ,
	[idtipo_pago] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[usuario_area_autoriza] (
	[idempresa] [int] NOT NULL ,
	[numempleado] [int] NOT NULL ,
	[autorizador] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[usuario_area_autoriza_importar] (
	[no_empresa] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[numempleado] [int] NULL ,
	[autorizador] [int] NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[usuario_direccion_autoriza] (
	[idempresa] [int] NOT NULL ,
	[numempleado] [int] NOT NULL ,
	[autorizador] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[usuario_direccion_autoriza_importar] (
	[no_empresa] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[numempleado] [int] NULL ,
	[autorizador] [int] NULL ,
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[usuario_importar] (
	[no_empresa] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[numempleado] [int] NOT NULL ,
	[ccostos] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[nombre_puesto] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[nivel] [varchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[paterno] [varchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[materno] [varchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[nombre] [varchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[rfc] [varchar] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[banco] [char] (15) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[no_cuenta] [char] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[status] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
	[email] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[idjefe] [int] NULL ,
```

```
	[llave_carga] [int] NULL ,
	[reject_message] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
	[fecha_ult_act] [datetime] NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[anticipo] ADD
	CONSTRAINT [PK_anticipo] PRIMARY KEY  CLUSTERED
	(
		[folio]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[area_responsabilidad] ADD
	CONSTRAINT [PK_area_responsabilidad] PRIMARY KEY  CLUSTERED
	(
		[idarea],
		[idempresa]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[area_responsabilidad_autoriza] ADD
	CONSTRAINT [PK_area_autoriza] PRIMARY KEY  CLUSTERED
	(
		[idarea],
		[idempresa],
		[numempleado]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[banco] ADD
	CONSTRAINT [PK_banco] PRIMARY KEY  CLUSTERED
	(
		[idbanco]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[banco_importar] ADD
	CONSTRAINT [PK_banco_importar] PRIMARY KEY  CLUSTERED
	(
		[idbanco]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[carga] ADD
	CONSTRAINT [PK_carga] PRIMARY KEY  CLUSTERED
	(
		[llave_carga]
	) WITH  FILLFACTOR = 90  ON [PRIMARY]
GO

ALTER TABLE [dbo].[centro_costo] ADD
	CONSTRAINT [PK_centro_costos] PRIMARY KEY  CLUSTERED
	(
		[idccostos],
		[idempresa]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[centro_costo_autoriza] ADD
	CONSTRAINT [PK_ccostos_autoriza] PRIMARY KEY  CLUSTERED
	(
```

```
		[idccostos],
		[idempresa]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[centro_costo_importar] ADD
	CONSTRAINT [PK_centro_costo_importar] PRIMARY KEY  CLUSTERED
	(
		[ccostos],
		[empresa]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[concepto_gasto] ADD
	CONSTRAINT [PK_concepto_gasto] PRIMARY KEY  CLUSTERED
	(
		[idconcepto],
		[idempresa]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[concepto_gasto_area] ADD
	CONSTRAINT [PK_concepto_gasto_area] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idconcepto],
		[idarea]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[concepto_gasto_asiento_contable] ADD
	CONSTRAINT [PK_concepto_gasto_asiento_contable] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idconcepto],
		[idasiento]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[concepto_gasto_tipo_concepto] ADD
	CONSTRAINT [PK_concepto_gasto_tipo_concepto] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idconcepto],
		[idtipo_operacion]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[contabilidad_autoriza] ADD
	CONSTRAINT [PK_contabilidad_autoriza] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[numempleado]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[costo_por_km] ADD
	CONSTRAINT [PK_costo_por_km] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idmoneda]
	) ON [PRIMARY]
```

```
GO

ALTER TABLE [dbo].[cuenta_contable] ADD
	CONSTRAINT [PK_cuentas_contables] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[ccontable]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[deposito] ADD
	CONSTRAINT [PK_deposito] PRIMARY KEY  CLUSTERED
	(
		[folio]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[deposito_importar] ADD
	CONSTRAINT [PK_deposito_importar] PRIMARY KEY  CLUSTERED
	(
		[folio]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_anexo] ADD
	CONSTRAINT [PK_detalle_anticipo_anexo] PRIMARY KEY  CLUSTERED
	(
		[idanexo]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_boleto_avion] ADD
	CONSTRAINT [PK_detalle_boleto_avion] PRIMARY KEY  CLUSTERED
	(
		[detalle]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_comensales] ADD
	CONSTRAINT [PK_detalle_comensales] PRIMARY KEY  CLUSTERED
	(
		[detalle]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_deposito] ADD
	CONSTRAINT [PK_detalle_deposito] PRIMARY KEY  CLUSTERED
	(
		[detalle]
	)  ON [PRIMARY] ,
	CONSTRAINT [IX_detalle_deposito] UNIQUE  NONCLUSTERED
	(
		[folio_deposito]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_dias] ADD
	CONSTRAINT [PK_detalle_dias] PRIMARY KEY  CLUSTERED
	(
		[detalle]
	)  ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[detalle_kilometro] ADD
	CONSTRAINT [PK_detalle_kilometro] PRIMARY KEY  CLUSTERED
	(
		[detalle]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_precio_unitario_unidades] ADD
	CONSTRAINT [PK_detalle_precio_unitario_unidades] PRIMARY KEY  CLUSTERED
	(
		[detalle]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_principal_detalle_relacionado] ADD
	CONSTRAINT [PK_detalle_compra_detalle_pago] PRIMARY KEY  CLUSTERED
	(
		[detalle_principal],
		[detalle_relacionado]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[detalle_solicitud] ADD
	CONSTRAINT [PK_detalle_solicitud] PRIMARY KEY  CLUSTERED
	(
		[detalle]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[distribucion_solicitud] ADD
	CONSTRAINT [PK_distribucion_anticipo] PRIMARY KEY  CLUSTERED
	(
		[distribucion]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[empresa] ADD
	CONSTRAINT [PK_empresa] PRIMARY KEY  CLUSTERED
	(
		[idempresa]
	) WITH  FILLFACTOR = 90  ON [PRIMARY] ,
	CONSTRAINT [UC_empresa1] UNIQUE  NONCLUSTERED
	(
		[no_empresa]
	) WITH  FILLFACTOR = 90  ON [PRIMARY]
GO

ALTER TABLE [dbo].[empresa_importar] ADD
	CONSTRAINT [DF__empresa_i__llave__0A943F91] DEFAULT (0) FOR [llave_carga],
	CONSTRAINT [PK_empresa_importar] PRIMARY KEY  CLUSTERED
	(
		[no_empresa]
	) WITH  FILLFACTOR = 90  ON [PRIMARY]
GO

ALTER TABLE [dbo].[flujo_autoriza] ADD
	CONSTRAINT [PK_flujo_autoriza_anticipo] PRIMARY KEY  CLUSTERED
	(
		[idflujo],
		[idempresa],
		[idtipo_operacion]
```

```
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[flujo_autoriza_opcion] ADD
	CONSTRAINT [PK_flujo_autoriza_opcion] PRIMARY KEY CLUSTERED
	(
		[idempresa]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[folio_principal_folio_relacionado] ADD
	CONSTRAINT [PK_folio_relacionado] PRIMARY KEY CLUSTERED
	(
		[folio_principal],
		[folio_relacionado]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[historial_solicitud] ADD
	CONSTRAINT [PK_historial_anticipo] PRIMARY KEY CLUSTERED
	(
		[idhistorial]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[iva] ADD
	CONSTRAINT [PK_iva] PRIMARY KEY CLUSTERED
	(
		[identificador]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[iva_importar] ADD
	CONSTRAINT [PK_iva_importar] PRIMARY KEY CLUSTERED
	(
		[identificador]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[kilometro] ADD
	CONSTRAINT [PK_kilometro] PRIMARY KEY CLUSTERED
	(
		[idkm],
		[idempresa]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[mensaje_status] ADD
	CONSTRAINT [PK_mensaje_status] PRIMARY KEY CLUSTERED
	(
		[idempresa],
		[idstatus]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[moneda] ADD
	CONSTRAINT [PK_moneda] PRIMARY KEY CLUSTERED
	(
		[idmoneda]
	) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[monto_limite] ADD
	CONSTRAINT [PK_montos_limite] PRIMARY KEY  CLUSTERED
	(
		[idnivel],
		[idconcepto],
		[idtipo_operacion],
		[idempresa],
		[idmoneda],
		[idterritorio]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[nivel] ADD
	CONSTRAINT [PK_nivel] PRIMARY KEY  CLUSTERED
	(
		[idnivel],
		[idempresa]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[object_control] ADD
	CONSTRAINT [PK_object_control] PRIMARY KEY  CLUSTERED
	(
		[object_key]
	) WITH  FILLFACTOR = 90  ON [PRIMARY]
GO

ALTER TABLE [dbo].[objeto] ADD
	CONSTRAINT [PK_objeto] PRIMARY KEY  CLUSTERED
	(
		[idobjeto]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[objeto_ubicacion] ADD
	CONSTRAINT [PK_objeto_ubicacion] PRIMARY KEY  CLUSTERED
	(
		[idubicacion]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[operacion_proceso] ADD
	CONSTRAINT [PK_operaciones_proceso] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idperfil],
		[idtipo_operacion]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[pago] ADD
	CONSTRAINT [PK_pago] PRIMARY KEY  CLUSTERED
	(
		[folio]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[perfil] ADD
	CONSTRAINT [PK_perfil] PRIMARY KEY  CLUSTERED
	(
		[idperfil],
		[idempresa]
```

```
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[perfil_objeto] ADD
	CONSTRAINT [PK_perfil_objeto] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idperfil],
		[idobjeto]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[perfil_tipo_autorizacion] ADD
	CONSTRAINT [PK_perfil_tipo_autorizacion] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idperfil]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[ppto_detalle] ADD
	CONSTRAINT [PK_ppto_detalle] PRIMARY KEY  CLUSTERED
	(
		[idppto],
		[detalle]
	) ON [PRIMARY] ,
	CONSTRAINT [IX_ppto_detalle] UNIQUE  NONCLUSTERED
	(
		[idreg]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[presupuesto] ADD
	CONSTRAINT [PK_presupuesto] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[ccontable],
		[anio],
		[mes],
		[identidad]
	) ON [PRIMARY] ,
	CONSTRAINT [IX_presupuesto] UNIQUE  NONCLUSTERED
	(
		[idppto]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[presupuesto_importar] ADD
	CONSTRAINT [PK_presupuesto_importar] PRIMARY KEY  CLUSTERED
	(
		[idregistro]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[proveedor] ADD
	CONSTRAINT [PK_proveedor] PRIMARY KEY  CLUSTERED
	(
		[idproveedor]
	) ON [PRIMARY]
GO

ALTER TABLE [dbo].[proveedor_concepto] ADD
```

```
	CONSTRAINT [PK_proveedor_concepto] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idproveedor],
		[idconcepto]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[proveedor_empresa] ADD
	CONSTRAINT [PK_proveedor_empresa] PRIMARY KEY  CLUSTERED
	(
		[idproveedor],
		[idempresa]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[proveedor_importar] ADD
	CONSTRAINT [PK_proveedor_importar] PRIMARY KEY  CLUSTERED
	(
		[identificador]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[solicitud] ADD
	CONSTRAINT [PK_comprobacion_gasto] PRIMARY KEY  CLUSTERED
	(
		[folio]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[status] ADD
	CONSTRAINT [Status_PK] PRIMARY KEY  CLUSTERED
	(
		[idstatus]
	) WITH  FILLFACTOR = 90  ON [PRIMARY]
GO

ALTER TABLE [dbo].[status_rechazo] ADD
	CONSTRAINT [PK_status_rechazo] PRIMARY KEY  CLUSTERED
	(
		[idstatus]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[territorio] ADD
	CONSTRAINT [PK_territorio] PRIMARY KEY  CLUSTERED
	(
		[idterritorio]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[tipo_cambio] ADD
	CONSTRAINT [PK_tipo_cambio] PRIMARY KEY  CLUSTERED
	(
		[idmoneda_origen],
		[idmoneda_destino],
		[fecha]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[tipo_cambio_importar] ADD
	CONSTRAINT [PK_tipo_cambio_importar] PRIMARY KEY  CLUSTERED
```

```
	(
		[idmoneda_origen],
		[idmoneda_destino],
		[fecha]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[tipo_concepto] ADD
	CONSTRAINT [PK_tipo_concepto] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idtipo_concepto]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[tipo_operacion] ADD
	CONSTRAINT [PK_tipo_operacion] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[idtipo_operacion]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[tipo_pago] ADD
	CONSTRAINT [PK_tipo_pago] PRIMARY KEY  CLUSTERED
	(
		[idtipo_pago]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[tipo_usuario] ADD
	CONSTRAINT [PK_tipo_usuario] PRIMARY KEY  CLUSTERED
	(
		[idtipo_usuario]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[usuario] ADD
	CONSTRAINT [PK_usuarios] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[numempleado]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[usuario_area_autoriza] ADD
	CONSTRAINT [PK_usuario_area_autoriza] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[numempleado],
		[autorizador]
	)  ON [PRIMARY]
GO

ALTER TABLE [dbo].[usuario_direccion_autoriza] ADD
	CONSTRAINT [PK_usuario_direccion_autoriza] PRIMARY KEY  CLUSTERED
	(
		[idempresa],
		[numempleado]
	)  ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[anticipo] ADD
	CONSTRAINT [FK_anticipo_solicitud] FOREIGN KEY
	(
		[folio]
	) REFERENCES [dbo].[solicitud] (
		[folio]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[area_responsabilidad] ADD
	CONSTRAINT [FK_area_responsabilidad_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	)
GO

ALTER TABLE [dbo].[area_responsabilidad_autoriza] ADD
	CONSTRAINT [FK_area_responsabilidad_autoriza_area_responsabilidad] FOREIGN KEY
	(
		[idarea],
		[idempresa]
	) REFERENCES [dbo].[area_responsabilidad] (
		[idarea],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_area_responsabilidad_autoriza_usuario] FOREIGN KEY
	(
		[idempresa],
		[numempleado]
	) REFERENCES [dbo].[usuario] (
		[idempresa],
		[numempleado]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[banco] ADD
	CONSTRAINT [FK_banco_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	)
GO

ALTER TABLE [dbo].[banco_importar] ADD
	CONSTRAINT [FK_banco_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[centro_costo] ADD
	CONSTRAINT [FK_centro_costo_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
```

```
	CONSTRAINT [FK_centro_costo_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	)
GO

ALTER TABLE [dbo].[centro_costo_autoriza] ADD
	CONSTRAINT [FK_centro_costo_autoriza_centro_costo] FOREIGN KEY
	(
		[idccostos],
		[idempresa]
	) REFERENCES [dbo].[centro_costo] (
		[idccostos],
		[idempresa]
	),
	CONSTRAINT [FK_centro_costo_autoriza_usuario] FOREIGN KEY
	(
		[idempresa],
		[numempleado]
	) REFERENCES [dbo].[usuario] (
		[idempresa],
		[numempleado]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[centro_costo_importar] ADD
	CONSTRAINT [FK_centro_costo_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[concepto_gasto] ADD
	CONSTRAINT [FK_concepto_gasto_cuenta_contable] FOREIGN KEY
	(
		[idempresa],
		[ccontable_deducible]
	) REFERENCES [dbo].[cuenta_contable] (
		[idempresa],
		[ccontable]
	),
	CONSTRAINT [FK_concepto_gasto_cuenta_contable1] FOREIGN KEY
	(
		[idempresa],
		[ccontable_no_deducible]
	) REFERENCES [dbo].[cuenta_contable] (
		[idempresa],
		[ccontable]
	)
GO

ALTER TABLE [dbo].[concepto_gasto_area] ADD
	CONSTRAINT [FK_concepto_gasto_area_area_responsabilidad] FOREIGN KEY
	(
		[idarea],
		[idempresa]
	) REFERENCES [dbo].[area_responsabilidad] (
		[idarea],
```

```
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_concepto_gasto_area_concepto_gasto] FOREIGN KEY
	(
		[idconcepto],
		[idempresa]
	) REFERENCES [dbo].[concepto_gasto] (
		[idconcepto],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[concepto_gasto_tipo_concepto] ADD
	CONSTRAINT [FK_concepto_gasto_tipo_concepto_concepto_gasto] FOREIGN KEY
	(
		[idconcepto],
		[idempresa]
	) REFERENCES [dbo].[concepto_gasto] (
		[idconcepto],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_concepto_gasto_tipo_concepto_tipo_concepto] FOREIGN KEY
	(
		[idempresa],
		[idtipo_concepto]
	) REFERENCES [dbo].[tipo_concepto] (
		[idempresa],
		[idtipo_concepto]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_concepto_gasto_tipo_concepto_tipo_operacion1] FOREIGN KEY
	(
		[idempresa],
		[idtipo_operacion]
	) REFERENCES [dbo].[tipo_operacion] (
		[idempresa],
		[idtipo_operacion]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[contabilidad_autoriza] ADD
	CONSTRAINT [FK_contabilidad_autoriza_usuario] FOREIGN KEY
	(
		[idempresa],
		[numempleado]
	) REFERENCES [dbo].[usuario] (
		[idempresa],
		[numempleado]
	) ON DELETE CASCADE
GO

ALTER TABLE [dbo].[costo_por_km] ADD
	CONSTRAINT [FK_costo_por_km_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[cuenta_contable] ADD
	CONSTRAINT [FK_cuenta_contable_empresa] FOREIGN KEY
	(
```

```
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_cuenta_contable_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	) ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[cuenta_contable_importar] ADD
	CONSTRAINT [FK_cuenta_contable_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[deposito] ADD
	CONSTRAINT [FK_deposito_banco] FOREIGN KEY
	(
		[idbanco]
	) REFERENCES [dbo].[banco] (
		[idbanco]
	),
	CONSTRAINT [FK_deposito_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	),
	CONSTRAINT [FK_deposito_moneda] FOREIGN KEY
	(
		[idmoneda]
	) REFERENCES [dbo].[moneda] (
		[idmoneda]
	),
	CONSTRAINT [FK_deposito_usuario] FOREIGN KEY
	(
		[idempresa],
		[numempleado]
	) REFERENCES [dbo].[usuario] (
		[idempresa],
		[numempleado]
	)
GO

ALTER TABLE [dbo].[deposito_importar] ADD
	CONSTRAINT [FK_deposito_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[detalle_anexo] ADD
	CONSTRAINT [FK_detalle_anexo_detalle_solicitud] FOREIGN KEY
	(
```

```
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[detalle_boleto_avion] ADD
	CONSTRAINT [FK_detalle_boleto_avion_detalle_solicitud] FOREIGN KEY
	(
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[detalle_comensales] ADD
	CONSTRAINT [FK_detalle_comensales_detalle_solicitud] FOREIGN KEY
	(
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[detalle_deposito] ADD
	CONSTRAINT [FK_detalle_deposito_detalle_solicitud] FOREIGN KEY
	(
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[detalle_dias] ADD
	CONSTRAINT [FK_detalle_dias_detalle_solicitud] FOREIGN KEY
	(
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[detalle_kilometro] ADD
	CONSTRAINT [FK_detalle_kilometro_detalle_solicitud] FOREIGN KEY
	(
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[detalle_precio_unitario_unidades] ADD
	CONSTRAINT [FK_detalle_precio_unitario_unidades_detalle_solicitud] FOREIGN KEY
	(
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[detalle_principal_detalle_relacionado] ADD
	CONSTRAINT [FK_detalle_compra_detalle_pago_detalle_solicitud] FOREIGN KEY
```

```
	(
		[detalle_principal]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	),
	CONSTRAINT [FK_detalle_compra_detalle_pago_detalle_solicitud1] FOREIGN KEY
	(
		[detalle_relacionado]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[detalle_solicitud] ADD
	CONSTRAINT [FK_detalle_solicitud_concepto_gasto] FOREIGN KEY
	(
		[idconcepto],
		[idempresa]
	) REFERENCES [dbo].[concepto_gasto] (
		[idconcepto],
		[idempresa]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_detalle_solicitud_solicitud] FOREIGN KEY
	(
		[folio]
	) REFERENCES [dbo].[solicitud] (
		[folio]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[distribucion_solicitud] ADD
	CONSTRAINT [FK_distribucion_solicitud_detalle_solicitud] FOREIGN KEY
	(
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[empresa] ADD
	CONSTRAINT [FK_empresa_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	)
GO

ALTER TABLE [dbo].[empresa_importar] ADD
	CONSTRAINT [FK_empresa_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	) ON DELETE CASCADE
GO

ALTER TABLE [dbo].[flujo_autoriza] ADD
	CONSTRAINT [FK_flujo_autoriza_anticipo_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
```

```
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_flujo_autoriza_anticipo_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[flujo_autoriza_opcion] ADD
	CONSTRAINT [FK_flujo_autoriza_opcion_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[folio_principal_folio_relacionado] ADD
	CONSTRAINT [FK_folio_relacionado_solicitud] FOREIGN KEY
	(
		[folio_relacionado]
	) REFERENCES [dbo].[solicitud] (
		[folio]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[historial_solicitud] ADD
	CONSTRAINT [FK_historial_solicitud_solicitud] FOREIGN KEY
	(
		[folio]
	) REFERENCES [dbo].[solicitud] (
		[folio]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[iva] ADD
	CONSTRAINT [FK_iva_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	)
GO

ALTER TABLE [dbo].[iva_importar] ADD
	CONSTRAINT [FK_iva_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[kilometro] ADD
	CONSTRAINT [FK_kilometro_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE
```

```
GO

ALTER TABLE [dbo].[mensaje_status] ADD
	CONSTRAINT [FK_mensaje_status_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_mensaje_status_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[moneda] ADD
	CONSTRAINT [FK_moneda_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	)
GO

ALTER TABLE [dbo].[moneda_importar] ADD
	CONSTRAINT [FK_moneda_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[monto_limite] ADD
	CONSTRAINT [FK_monto_limite_concepto_gasto] FOREIGN KEY
	(
		[idconcepto],
		[idempresa]
	) REFERENCES [dbo].[concepto_gasto] (
		[idconcepto],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_monto_limite_moneda] FOREIGN KEY
	(
		[idmoneda]
	) REFERENCES [dbo].[moneda] (
		[idmoneda]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_monto_limite_nivel] FOREIGN KEY
	(
		[idnivel],
		[idempresa]
	) REFERENCES [dbo].[nivel] (
		[idnivel],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_monto_limite_territorio] FOREIGN KEY
	(
		[idterritorio]
	) REFERENCES [dbo].[territorio] (
```

```
		[idterritorio]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_monto_limite_tipo_operacion] FOREIGN KEY
	(
		[idempresa],
		[idtipo_operacion]
	) REFERENCES [dbo].[tipo_operacion] (
		[idempresa],
		[idtipo_operacion]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[objeto] ADD
	CONSTRAINT [FK_objeto_objeto_ubicacion] FOREIGN KEY
	(
		[idubicacion]
	) REFERENCES [dbo].[objeto_ubicacion] (
		[idubicacion]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[operacion_proceso] ADD
	CONSTRAINT [FK_operaciones_proceso_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_operaciones_proceso_perfil] FOREIGN KEY
	(
		[idperfil],
		[idempresa]
	) REFERENCES [dbo].[perfil] (
		[idperfil],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_operaciones_proceso_tipo_operacion] FOREIGN KEY
	(
		[idempresa],
		[idtipo_operacion]
	) REFERENCES [dbo].[tipo_operacion] (
		[idempresa],
		[idtipo_operacion]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[pago] ADD
	CONSTRAINT [FK_pago_proveedor] FOREIGN KEY
	(
		[idproveedor]
	) REFERENCES [dbo].[proveedor] (
		[idproveedor]
	),
	CONSTRAINT [FK_pago_solicitud] FOREIGN KEY
	(
		[folio]
	) REFERENCES [dbo].[solicitud] (
		[folio]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[perfil_objeto] ADD
```

```
	CONSTRAINT [FK_perfil_objeto_objeto] FOREIGN KEY
	(
		[idobjeto]
	) REFERENCES [dbo].[objeto] (
		[idobjeto]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_perfil_objeto_perfil] FOREIGN KEY
	(
		[idperfil],
		[idempresa]
	) REFERENCES [dbo].[perfil] (
		[idperfil],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[perfil_tipo_autorizacion] ADD
	CONSTRAINT [FK_perfil_tipo_autorizacion_perfil] FOREIGN KEY
	(
		[idperfil],
		[idempresa]
	) REFERENCES [dbo].[perfil] (
		[idperfil],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[ppto_detalle] ADD
	CONSTRAINT [FK_ppto_detalle_detalle_solicitud] FOREIGN KEY
	(
		[detalle]
	) REFERENCES [dbo].[detalle_solicitud] (
		[detalle]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_ppto_detalle_presupuesto] FOREIGN KEY
	(
		[idppto]
	) REFERENCES [dbo].[presupuesto] (
		[idppto]
	)
GO

ALTER TABLE [dbo].[presupuesto] ADD
	CONSTRAINT [FK_presupuesto_cuenta_contable] FOREIGN KEY
	(
		[idempresa],
		[ccontable]
	) REFERENCES [dbo].[cuenta_contable] (
		[idempresa],
		[ccontable]
	),
	CONSTRAINT [FK_presupuesto_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[presupuesto_importar] ADD
	CONSTRAINT [FK_presupuesto_importar_carga] FOREIGN KEY
	(
```

```
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[proveedor] ADD
	CONSTRAINT [FK_proveedor_banco] FOREIGN KEY
	(
		[idbanco]
	) REFERENCES [dbo].[banco] (
		[idbanco]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_proveedor_tipo_pago] FOREIGN KEY
	(
		[idtipo_pago]
	) REFERENCES [dbo].[tipo_pago] (
		[idtipo_pago]
	) ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[proveedor_concepto] ADD
	CONSTRAINT [FK_proveedor_concepto_concepto_gasto] FOREIGN KEY
	(
		[idconcepto],
		[idempresa]
	) REFERENCES [dbo].[concepto_gasto] (
		[idconcepto],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_proveedor_concepto_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_proveedor_concepto_proveedor] FOREIGN KEY
	(
		[idproveedor]
	) REFERENCES [dbo].[proveedor] (
		[idproveedor]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[proveedor_empresa] ADD
	CONSTRAINT [FK_proveedor_empresa_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_proveedor_empresa_proveedor] FOREIGN KEY
	(
		[idproveedor]
	) REFERENCES [dbo].[proveedor] (
		[idproveedor]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_proveedor_empresa_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
```

```
	)
GO

ALTER TABLE [dbo].[proveedor_empresa_importar] ADD
	CONSTRAINT [FK_proveedor_empresa_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[proveedor_importar] ADD
	CONSTRAINT [FK_proveedor_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[solicitud] ADD
	CONSTRAINT [FK_solicitud_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	),
	CONSTRAINT [FK_solicitud_moneda] FOREIGN KEY
	(
		[idmoneda]
	) REFERENCES [dbo].[moneda] (
		[idmoneda]
	),
	CONSTRAINT [FK_solicitud_moneda1] FOREIGN KEY
	(
		[idmoneda_pago]
	) REFERENCES [dbo].[moneda] (
		[idmoneda]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_solicitud_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	),
	CONSTRAINT [FK_solicitud_territorio] FOREIGN KEY
	(
		[idterritorio]
	) REFERENCES [dbo].[territorio] (
		[idterritorio]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_solicitud_usuario] FOREIGN KEY
	(
		[idempresa],
		[numempleado]
	) REFERENCES [dbo].[usuario] (
		[idempresa],
		[numempleado]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO
```

```
ALTER TABLE [dbo].[status_rechazo] ADD
	CONSTRAINT [FK_status_rechazo_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_status_rechazo_status1] FOREIGN KEY
	(
		[idstatus_rechazo]
	) REFERENCES [dbo].[status] (
		[idstatus]
	)
GO

ALTER TABLE [dbo].[tipo_cambio] ADD
	CONSTRAINT [FK_tipo_cambio_moneda] FOREIGN KEY
	(
		[idmoneda_origen]
	) REFERENCES [dbo].[moneda] (
		[idmoneda]
	),
	CONSTRAINT [FK_tipo_cambio_moneda1] FOREIGN KEY
	(
		[idmoneda_destino]
	) REFERENCES [dbo].[moneda] (
		[idmoneda]
	)
GO

ALTER TABLE [dbo].[tipo_cambio_importar] ADD
	CONSTRAINT [FK_tipo_cambio_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[tipo_concepto] ADD
	CONSTRAINT [FK_tipo_concepto_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	)
GO

ALTER TABLE [dbo].[tipo_pago_importar] ADD
	CONSTRAINT [FK_tipo_pago_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

ALTER TABLE [dbo].[usuario] ADD
	CONSTRAINT [FK_usuario_banco] FOREIGN KEY
	(
		[idbanco]
	) REFERENCES [dbo].[banco] (
```

```
		[idbanco]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_usuario_centro_costo] FOREIGN KEY
	(
		[ccostos],
		[idempresa]
	) REFERENCES [dbo].[centro_costo] (
		[idccostos],
		[idempresa]
	) ON DELETE CASCADE  ON UPDATE CASCADE ,
	CONSTRAINT [FK_usuario_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	),
	CONSTRAINT [FK_usuario_nivel] FOREIGN KEY
	(
		[idnivel],
		[idempresa]
	) REFERENCES [dbo].[nivel] (
		[idnivel],
		[idempresa]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_usuario_perfil] FOREIGN KEY
	(
		[idperfil],
		[idempresa]
	) REFERENCES [dbo].[perfil] (
		[idperfil],
		[idempresa]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_usuario_status] FOREIGN KEY
	(
		[idstatus]
	) REFERENCES [dbo].[status] (
		[idstatus]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_usuario_tipo_pago] FOREIGN KEY
	(
		[idtipo_pago]
	) REFERENCES [dbo].[tipo_pago] (
		[idtipo_pago]
	) ON UPDATE CASCADE ,
	CONSTRAINT [FK_usuario_tipo_usuario] FOREIGN KEY
	(
		[idtipo_usuario]
	) REFERENCES [dbo].[tipo_usuario] (
		[idtipo_usuario]
	) ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[usuario_area_autoriza] ADD
	CONSTRAINT [FK_usuario_area_autoriza_usuario] FOREIGN KEY
	(
		[idempresa],
		[numempleado]
	) REFERENCES [dbo].[usuario] (
		[idempresa],
		[numempleado]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO
```

```
ALTER TABLE [dbo].[usuario_area_autoriza_importar] ADD
	CONSTRAINT [FK_usuario_area_autoriza_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[usuario_direccion_autoriza] ADD
	CONSTRAINT [FK_usuario_direccion_autoriza_empresa] FOREIGN KEY
	(
		[idempresa]
	) REFERENCES [dbo].[empresa] (
		[idempresa]
	),
	CONSTRAINT [FK_usuario_direccion_autoriza_usuario] FOREIGN KEY
	(
		[idempresa],
		[numempleado]
	) REFERENCES [dbo].[usuario] (
		[idempresa],
		[numempleado]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[usuario_direccion_autoriza_importar] ADD
	CONSTRAINT [FK_usuario_direccion_autoriza_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	) ON DELETE CASCADE  ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[usuario_importar] ADD
	CONSTRAINT [FK_usuario_importar_carga] FOREIGN KEY
	(
		[llave_carga]
	) REFERENCES [dbo].[carga] (
		[llave_carga]
	)
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.so_status_catalogo    Script Date: 02/07/2008 04:36:22 p.m. ******/


/*
Inicia
Nombre:
so_status_catalogo
```

```
Funcionalidad:
Se obtiene la descripcion o nombre del status de los status de rechazo y de espera de autorizacion

Entradas:
Accion a seguir (por el momento no se usa, se dejo para uso posterior)

Proceso:
Obtiene nombre de los status

Salida:
Nombre de los status

Ejemplo:
exec so_status_catalogo ''
Termina
*/

CREATE procedure [dbo].so_status_catalogo
(
	@proviene char(1)
)
as
begin
	select
	idstatus,
	descripcion_1
	from status
	where	substring(idstatus,1,1) in ('e','r')
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_anticipo_inserta_modifica    Script Date: 02/07/2008 04:36:29 p.m.
******/




/*
Inicia
Nombre:
sp_anticipo_inserta_modifica
```

```
Funcionalidad:
Inserta, modifica o elimina una solicitud de anticipo

Entradas:
Folio del anticipo
Fecha de inicio del anticipo
Fecha de fin del anticipo
Opcion de accion
        (i = inserta nuevo anticipo)
        (u = actualiza anticipo existente)
        (d = elimina anticipo existente)

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta, actualiza o elimina un registro de anticipo

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
Termina
*/

CREATE procedure [dbo].sp_anticipo_inserta_modifica
(
        @folio int,
        @fecha_ini char(11),
        @fecha_fin char(11),
        @proviene char(1)
)
as
begin
        /*Inserta un anticipo*/
        if @proviene = 'i'
        begin
                set dateformat dmy
                insert into anticipo
                (
                folio,
                fecha_ini,
                fecha_fin
                )
                values
                (
                @folio,
                [dbo].sp_convierte_texto_a_fecha(@fecha_ini),
                [dbo].sp_convierte_texto_a_fecha(@fecha_fin)
                )
        end

        /*Modifica un anticipo*/
        if @proviene = 'u'
        begin
                set dateformat dmy
                update anticipo set
                fecha_ini = [dbo].sp_convierte_texto_a_fecha(@fecha_ini),
                fecha_fin = [dbo].sp_convierte_texto_a_fecha(@fecha_fin)
                where folio = @folio
        end

        /*Elimina un anticipo*/
        if @proviene = 'd'
        begin
```

```
                delete from anticipo
                where folio = @folio
        end

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_anticipo_muestra_datos_parti    Script Date: 02/07/2008 04:36:29
p.m. ******/




/*
Inicia
Nombre:
sp_anticipo_muestra_datos_parti



Funcionalidad:
Muestra los datos particulares de una solicitud de anticipo

Entradas:
Folio del anticipo
Opcion de conversion de fecha
        (0 = no convertir la fecha en formato texto)
        (1 = convertir la fecha en formato texto)

Proceso:
Presenta los datos particulares de la solicitud de anticipo en un formato u otro

Salida:
Datos de una solicitud de anticipo

Ejemplo:
exec sp_anticipo_muestra_datos_parti 4,0
Termina
*/


CREATE procedure [dbo].sp_anticipo_muestra_datos_parti
(
        @folio int,
```

```
	@convierte_fecha_a_texto tinyint
)
as
begin
	/*
	No convertir la fecha a formato texto
	*/
	if @convierte_fecha_a_texto = 0
	begin
		select top 1
		convert(char,fecha_ini,103),
		convert(char,fecha_fin,103)
		from anticipo
		where folio = @folio
	end

	/*
	Convertir la fecha a formato texto
	*/
	if @convierte_fecha_a_texto = 1
	begin
		select top 1
		[dbo].sp_convierte_fecha_a_texto(convert(char,fecha_ini,103)),
		[dbo].sp_convierte_fecha_a_texto(convert(char,fecha_fin,103))
		from anticipo
		where folio = @folio
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/*
Inicia
Nombre:
sp_anticipos_pagados_catalogo

Fecha de ultima modificacion:
12 de julio del 2005

Funcionalidad:
Obtiene la relacion de anticipos pagados que pueden ser asociados a una comprobacion de anticipo

Entradas:
Empresa en la que se trabaja
Numero de empleado que solicito el anticipo

Proceso:
Obtiene la relacion de anticipos pagados, es decir, status = pg
```

```
Salida:
Relacion de solicitudes de anticipos

Ejemplo:
exec [dbo].sp_anticipos_pagados_catalogo 1,123
Termina
*/

CREATE procedure [dbo].sp_anticipos_pagados_catalogo
(
	@idempresa int,
	@numempleado int
)
as
begin
	declare
	@idtipo_operacion int

	/*
	Obtiene el id del tipo de operacion de anticipos para esta empresa
	*/
	set @idtipo_operacion = (select [dbo].fn_obtiene_tipo_operacion (@idempresa,'anticipo'))

	select
	so.folio,
	so.descripcion,
	isnull(sum(de.total),0) as total,
	m1.descripcion as idmoneda,
	so.idmoneda,
	m2.descripcion as idmoneda_pago,
	so.idmoneda_pago,
	te.descripcion as territorio,
	te.idterritorio,
	convert(char,so.fecha_solicitud,103) as fecha_solicitud
	from solicitud as so
	left outer join detalle_solicitud as de on so.folio = de.folio
	inner join moneda as m1 on so.idmoneda = m1.idmoneda
	inner join moneda as m2 on so.idmoneda_pago = m2.idmoneda
	inner join territorio as te on so.idterritorio = te.idterritorio
	where	so.idtipo_operacion = @idtipo_operacion and
		so.idstatus = 'pg' and
		so.numempleado = @numempleado and
		so.folio not in (select folio_principal from folio_principal_folio_relacionado)
	group by
so.folio,m1.descripcion,so.idmoneda,m2.descripcion,so.idmoneda_pago,te.descripcion,te.idterritorio,so.fecha_
solicitud,so.descripcion
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Stored Procedure dbo.sp_area_autoriza_muestra_datos    Script Date: 02/07/2008 04:36:26
p.m. ******/




/*
Inicia
Nombre:
sp_area_autoriza_muestra_datos



Funcionalidad:
Muestra los datos particulares de un area de responsabilidad

Entradas:
Numero de la empresa en la que se esta trabajando
Identificador del area
Accion a tomar
        o = Cuando no se quiere saber el nombre del area
        n = Cuando se quiere saber el nombre del area

Proceso:
Presenta los datos particulares de un area de responsabilidad

Salida:
Nombre del area de responsabilidad
Nombre de la persona responsable de esa area

Ejemplo:
exec sp_area_autoriza_muestra_datos 1,2
Termina
*/


CREATE procedure [dbo].sp_area_autoriza_muestra_datos
(
        @idempresa int,
        @idarea int,
        @proviene char(1) = 'o'     --
)
as
begin
        if @idarea <> 0
        begin
                if @proviene = 'n'
                begin
                        select
                        descripcion
                        from area_responsabilidad
                        where   idempresa = @idempresa and
                                idarea = @idarea
                end
                else
                begin
                        select
                        us.numempleado,
                        us.nombre + ' ' + us.paterno + ' ' + us.materno,
```

```
                        ar.descripcion
                        from area_responsabilidad as ar
                        inner join area_responsabilidad_autoriza as aa on ar.idarea = aa.idarea
                        inner join usuario as us on aa.numempleado = us.numempleado
                        where   ar.idempresa = aa.idempresa and
                                aa.idempresa = us.idempresa and
                                ar.idempresa = @idempresa and
                                ar.idarea = @idarea
                end
        end
        else
        begin
                select
                us.numempleado,
                us.nombre + ' ' + us.paterno + ' ' + us.materno
                from area_responsabilidad as ar
                inner join area_responsabilidad_autoriza as aa on ar.idarea = aa.idarea
                inner join usuario as us on aa.numempleado = us.numempleado
                where   ar.idempresa = aa.idempresa and
                        aa.idempresa = us.idempresa and
                        ar.idempresa = @idempresa
        end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_area_responsabilidad_catalogo    Script Date: 02/07/2008 04:36:22
p.m. ******/




/*
Inicia
Nombre:
sp_area_responsabilidad_catalogo



Nota:
Este procedimiento se obtiene el catalogo de areas de responsabilidad por empresa

Funcionalidad:
Obtiene las areas de responsabilidad vigentes por empresa

Entradas:
```

Numero de empresa en la que se esta trabajando

Proceso:
Obtencion de las areas de responsabilidad vigentes por empresa

Salida:
Areas de responsabilidad

Ejemplo:
exec sp_area_responsabilidad_catalogo 1
Termina
*/

```
CREATE procedure [dbo].sp_area_responsabilidad_catalogo
(
	@idempresa int
)
as
begin
	select
	idarea,
	descripcion,
	idstatus
	from area_responsabilidad
	where	idempresa = @idempresa and
		idstatus = 'a'
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

/****** Object: Stored Procedure dbo.sp_autorizacion_perfil   Script Date: 02/07/2008 04:36:24 p.m. ******/

/*
Inicia
Nombre:
sp_autorizacion_perfil

Funcionalidad:
Este procedimiento inserta, actualiza o elimina el tipo de autorizacion que tiene un perfil de usuario. Es decir, si ese per

Entradas:
Empresa a la que pertenece el perfil
Identificador del perfil
Tipo de autorizacion
	d = por detalle
	m = masiva
Accion a tomar
	i = insertar
	d = eliminar

Proceso:
Inserta, actualiza o elimina permisos de autorizacion para un perfil

Salida:
Permisos de autorizacion otorgados para un perfil

Ejemplo:
exec [dbo].sp_autorizacion_perfil 1,1,'d','i'
Termina
*/

```
CREATE procedure [dbo].sp_autorizacion_perfil
(
	@idempresa int,
	@idperfil int,
	@tipo_autorizacion char(1),
	@proviene char(1)
)
as
begin
	if @proviene = 'i'
		begin
			if exists(
				select
				1
				from perfil_tipo_autorizacion
				where	idempresa = @idempresa and
					idperfil = @idperfil		)
			begin
				update perfil_tipo_autorizacion
				set
				tipo_autorizacion = @tipo_autorizacion
				where	idempresa = @idempresa and
					idperfil = @idperfil
			end
			else
			begin
				insert into perfil_tipo_autorizacion
				(
				idempresa,
				idperfil,
				tipo_autorizacion
				)
				values
				(
				@idempresa,
				@idperfil,
				@tipo_autorizacion
				)
			end
		end
```

```
	if @proviene = 'd'
		begin
			delete
			from perfil_tipo_autorizacion
			where	idempresa = @idempresa and
				idperfil = @idperfil
		end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_banco_catalogo    Script Date: 02/07/2008 04:36:24 p.m. ******/




/*
Inicia
Nombre:
sp_banco_catalogo



Funcionalidad:
Este procedimiento obtiene datos relevantes del catalogo de bancos

Entradas:
Accion a tomar
	a = presenta datos relevantes para los administradores
	u = presenta datos relevantes para los usuarios normales sin perfil de administrador

Proceso:
Presenta datos relevantes dependiendo de donde se haga la peticion para obtener el catalogo

Salida:
Datos de los bancos

Ejemplo:
exec [dbo].sp_banco_catalogo 'a'
Termina
*/

CREATE procedure [dbo].sp_banco_catalogo
(
	@proviene char(1)
```

```
)
as
begin
	if @proviene = 'u'
	begin
		select
		ba.idbanco,
		ba.nombre,
		case
			when ba.nac_ext = 'n' then 'Nacional'
			when ba.nac_ext = 'e' then 'Extranjero'
			else 'No definido'
		end as tipo,
		case
			when ba.cuenta_nac is null then 'No definida'
			else ba.cuenta_nac
		end as cuenta_nac,
		case
			when ba.cuenta_ext is null then 'No definida'
			else ba.cuenta_ext
		end as cuenta_ext
		from banco as ba
		where ba.idstatus = 'a'
	end

	if @proviene = 'a'
	begin
		select
		ba.nombre,
		case
			when ba.nac_ext = 'n' then 'Nacional'
			when ba.nac_ext = 'e' then 'Extranjero'
			else 'No definido'
		end as tipo,
		case
			when ba.cuenta_nac is null then 'No definida'
			else ba.cuenta_nac
		end as cuenta_nac,
		case
			when ba.cuenta_ext is null then 'No definida'
			else ba.cuenta_ext
		end as cuenta_ext,
		st.descripcion_1
		from banco as ba
		left outer join status as st on ba.idstatus = st.idstatus
	end

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Stored Procedure dbo.sp_cargar_banco    Script Date: 02/07/2008 04:36:24 p.m. ******/



/*
Inicia
Nombre:
sp_cargar_banco



Funcionalidad:
Este procedimiento carga el catalogo de bancos a partir de la tabla intermedia de bancos

Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla banco a partir de los datos en banco_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_banco 1
Termina
*/

CREATE PROC [dbo].sp_cargar_banco
        @commit tinyint = 1
AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,

        @banco_OK int,

        @idbanco_importar char(5),
        @nombre_importar varchar(50),
        @nac_ext_importar char(1),
        @cuenta_nac_importar char(20),
        @cuenta_ext_importar char(20),
        @status_importar char(2),
        @llave_carga int,

        @idbanco int,
        @nombre varchar(50),
        @nac_ext char(1),
        @cuenta_nac char(20),
        @cuenta_ext char(20),
        @idstatus char(2),
        @identificador char(5),
```

```
	@registro_exists int,

	@count int,
	@reject_message varchar(50)

SELECT
	@object_name='banco',
	@count = 0

EXECUTE sp_rtrn_sequential
		@object_name,
		@last_key_assigned OUTPUT,
		@locked_status_flag OUTPUT,
		's'
PRINT 'status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON

/* Inicio de lógica si objeto está OK en object control */
IF @locked_status_flag = 0
BEGIN

	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	BEGIN TRANSACTION trans_load_banco

	PRINT 'Bloquear object_control:'
	EXECUTE sp_lock_object_control @object_name

	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR
	FOR	SELECT
			idbanco,
			nombre,
			nac_ext,
			cuenta_nac,
			cuenta_ext,
			status
			FROM	banco_importar
		WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@idbanco_importar,
		@nombre_importar,
		@nac_ext_importar,
		@cuenta_nac_importar,
		@cuenta_ext_importar,
		@status_importar

/* Bucle de lectura-validación-escritura */
	WHILE @@FETCH_status = 0
	BEGIN
		SET	@banco_OK = 1
		SET	@reject_message=' '

/* Inicia validación de banco */
/* Revisar si nombrebanco viene en blanco */
		IF @nombre_importar IS NULL
```

```
		OR @nombre_importar = ' '
		BEGIN
			SET   @banco_OK = 0
			SET      @reject_message='Nombre banco en blanco'
			PRINT   @reject_message
		END

		if @status_importar != 'a'
		begin
			set @status_importar = 'b'
		end

/* Si hay condición de error, escribir mensaje y fecha hora */
		IF @banco_OK = 0
		BEGIN
			UPDATE banco_importar
			SET
				[reject_message]= @reject_message,
				[fecha_ult_act]=getdate()
			WHERE
				idbanco = @idbanco_importar
		END

/* Si los datos son válidos, escribir */
		IF @banco_OK = 1
		BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
			UPDATE banco_importar
			SET
				[reject_message]= @reject_message,
				[llave_carga]=@llave_carga,
				[fecha_ult_act]=NULL
			WHERE idbanco = @idbanco_importar

				SET @count = @count + 1
				PRINT CONVERT(char(8),@count)+' '+' Banco: '+@idbanco_importar

/* Cargar o ajustar variables con las que se va a escribir */
				SET @identificador = @idbanco_importar
				SET @nombre=UPPER(LEFT(@nombre_importar,50))

/* Cargar o ajustar variables que se van a escribir */
			if @status_importar is null or @status_importar = ' '
				begin
					SET @idstatus = 'A'
				end
			else
				begin
					SET @idstatus = @status_importar
				end


/* Revisar si existe */
				SET @registro_exists =
					(
					SELECT idbanco
					FROM banco
					WHERE identificador = @idbanco_importar
					)

/*		Caso de q ya exista, actualizar */
```

```
                                        IF @registro_exists IS NOT NULL
                                        BEGIN
                                                UPDATE banco
                                                SET
                                                nombre=@nombre_importar,
                                                nac_ext=@nac_ext_importar,
                                                cuenta_nac = @cuenta_nac_importar,
                                                cuenta_ext = @cuenta_ext_importar,
                                                idstatus=@idstatus
                                                WHERE identificador = @idbanco_importar
                                                PRINT 'Banco actualizado'
                                        END

/* Caso de que no esté dado de alta, insertar nuevo registro */
                                        IF @registro_exists IS NULL
                                        BEGIN
                                                SET @last_key_assigned=@last_key_assigned+1
                                                SET @idbanco=@last_key_assigned

                                                INSERT INTO banco
                                                        (
                                                        idbanco,
                                                        nombre,
                                                        nac_ext,
                                                        cuenta_nac,
                                                        cuenta_ext,
                                                        idstatus,
                                                        identificador
                                                        )
                                                VALUES
                                                        (
                                                        @idbanco,
                                                        @nombre_importar,
                                                        @nac_ext_importar,
                                                        @cuenta_nac_importar,
                                                        @cuenta_ext_importar,
                                                        @idstatus,
                                                        @idbanco_importar
                                                        )
                                                PRINT 'Insertado'
                                        END

/* Fin de lógica si @banco_OK = 1 */
                        END

                        FETCH cur_1
                        INTO
                                @idbanco_importar,
                                @nombre_importar,
                                @nac_ext_importar,
                                @cuenta_nac_importar,
                                @cuenta_ext_importar,
                                @status_importar

/* Regreso de bucle de lectura de cur_1 */
                END

                CLOSE cur_1
                DEALLOCATE cur_1

                PRINT 'Desbloquear object_control:'
                EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned
```

```
		PRINT 'Fin de '+ @object_name +' load '

		IF @commit = 1
		BEGIN
			COMMIT TRANSACTION trans_load_banco
			PRINT 'Transacción commited'
		END
		IF @commit <> 1
		BEGIN
			ROLLBACK TRANSACTION trans_load_banco
			PRINT 'Transacción rolled-back'
		END

/* Fin de lógica si objeto está desbloqueado en object control */

		EXEC sp_end_of_new_load @llave_carga
END

/* Casos de encontrar algo mal en estado de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
		PRINT @object_name+' '+'BLOQUEADO'
END
IF @locked_status_flag IS NULL
BEGIN
		PRINT upper(@object_name)+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_cargar_centro_costo    Script Date: 02/07/2008 04:36:24 p.m. ******/




/*
Inicia
Nombre:
sp_cargar_centro_costo



Funcionalidad:
Este procedimiento carga el catalogo de centros de costo a partir de la tabla intermedia de
centro_costo_importar
```

```
Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla banco a partir de los datos en centro_costo_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_centro_costo 1
Termina
*/

CREATE PROC [dbo].sp_cargar_centro_costo
                        @commit tinyint = 1


AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,

        @centro_costos_OK int,

        @idempresa_importar char(15),
        @ccostos_importar char(10),

        @idempresa int,
        @ccostos char(10),
        @descripcion varchar(255),
        @Status char(2),
        @llave_carga int,

        @count int,
        @reject_message varchar(50)

SET     @object_name='centro_costo'
SET     @count = 0

EXECUTE sp_rtrn_sequential
                @object_name,
                @last_key_assigned OUTPUT,
                @locked_status_flag OUTPUT,
                's'
PRINT 'Status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

/* Inicio de lógica cuando bandera indica que objecto está desbloqueado */
IF @locked_status_flag = 0
BEGIN

        EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

        BEGIN TRANSACTION trans_load_centro_costo

        PRINT 'Bloquear object_control:'
        EXECUTE sp_lock_object_control @object_name
```

```
	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR
	FOR	SELECT
			ccostos,
			empresa,
			descripcion,
			status
		FROM	centro_costo_importar
		WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@ccostos_importar,
		@idempresa_importar,
		@descripcion,
		@Status

/* Bucle de lectura-validación-escritura */
	WHILE @@FETCH_STATUS = 0
	BEGIN
		SET @count = @count + 1
		SET @centro_costos_OK = 1
		SET @reject_message=' '
		PRINT CONVERT(char(8),@count)+' Empresa: '+ @idempresa_importar +' CC:
'+@ccostos_importar

/* Revisar si empresa es válida */
		SET @idempresa =
			(
			SELECT idempresa
			FROM empresa
			WHERE no_empresa = @idempresa_importar
			)

		IF @idempresa IS NULL
		BEGIN
			SET	@centro_costos_OK = 0
			SET	@reject_message='EMPRESA '+ @idempresa_importar + ' NO
REGISTRADA'
			PRINT @reject_message
		END

/* Revisar si ... viene en blanco (no aplica a ccostos)*/
		IF @ccostos_importar IS NULL
		OR @ccostos_importar = ' '
		BEGIN
			SET	@centro_costos_OK = 0
			SET	@reject_message='Centro costos en blanco'
			PRINT	@reject_message
		END

		IF @descripcion IS NULL
		OR @descripcion = ' '
		BEGIN
			SET	@centro_costos_OK = 0
			SET	@reject_message='Descripcion en blanco'
			PRINT	@reject_message
		END

		IF @status != 'a'
```

```
                BEGIN
                        SET   @status = 'b'
                END


/* Si hay condición de error, escribir mensaje y fecha hora */
                IF @centro_costos_OK = 0
                BEGIN
                        UPDATE centro_costo_importar
                        SET     [reject_message]= @reject_message,
                                [fecha_ult_act]=getdate()
                        WHERE
                                ccostos = @ccostos_importar AND
                                empresa = @idempresa_importar
                END

/* Escribir registros que pasan validación */

                IF @centro_costos_OK = 1
                BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
                        UPDATE centro_costo_importar
                        SET
                                [reject_message]= @reject_message,
                                [llave_carga]=@llave_carga,
                                [fecha_ult_act]=NULL
                        WHERE
                                ccostos = @ccostos_importar AND
                                empresa = @idempresa_importar

/* Cargar o ajustar variables que se van a escribir */
                        --SET @Status = 'A'

/* Revisar si existe */
                        SET @ccostos =
                                (
                                SELECT identificador
                                FROM centro_costo
                                WHERE           idempresa = @idempresa and
                                        identificador = @ccostos_importar
                                )


                        IF @ccostos IS NOT NULL
                        BEGIN
                                UPDATE centro_costo
                                SET
                                [descripcion]=@descripcion,
                                [idstatus] = @Status,
                                [llave_carga]=@llave_carga
                                WHERE idempresa = @idempresa
                                AND identificador = @ccostos
                                PRINT 'Centro de costos actualizado'
                        END

                        IF @ccostos IS NULL
                        BEGIN
                                SET @last_key_assigned=@last_key_assigned+1
                                INSERT INTO centro_costo
                                        (
                                        idempresa,
```

```
                                        idccostos,
                                        identificador,
                                        descripcion,
                                        idStatus,
                                        llave_carga
                                        )
                                VALUES          (
                                        @idempresa,
                                        @last_key_assigned,
                                        @ccostos_importar,
                                        @descripcion,
                                        @Status,
                                        @llave_carga
                                        )
                                        PRINT 'Insertado'
                                END

                        END

                        FETCH cur_1
                        INTO
                        @ccostos_importar,
                        @idempresa_importar,
                        @descripcion,
                        @Status

                END

                CLOSE cur_1
                DEALLOCATE cur_1

                PRINT 'Desbloquear object_control:'
                EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

                PRINT 'Fin de '+ @object_name +' load '

                IF @commit = 1
                BEGIN
                        COMMIT TRANSACTION trans_load_centro_costo
                        PRINT 'Transacción commited'
                END
                ELSE
                BEGIN
                        ROLLBACK TRANSACTION trans_load_centro_costo
                        PRINT 'Transacción rolled-back'
                END

                EXEC sp_end_of_new_load @llave_carga

/* Fin de lógica cuando bandera indica que el objecto está desbloqueado */
END

/* Caso de encontrar objeto bloqueado */
IF @locked_status_flag = 1
BEGIN
        PRINT @object_name+' '+'BLOQUEADO'
END

/* Si la bandera contiene carácter nulo, significa objeto no registrado en object_control */
IF @locked_status_flag IS NULL
BEGIN
        PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
```

```
END



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_cargar_cuenta_contable    Script Date: 02/07/2008 04:36:25 p.m.
******/




/*
Inicia
Nombre:
sp_cargar_cuenta_contable



Funcionalidad:
Este procedimiento carga el catalogo de cuentas contables a partir de la tabla intermedia de
cuenta_contable_importar

Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla cuenta_contable a partir de los datos en cuenta_contable_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_cuenta_contable 1
Termina
*/

CREATE PROC [dbo].sp_cargar_cuenta_contable
                        @commit tinyint = 1
AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,
```

```
	@cuenta_contable_OK int,

	@ccontable_importar char(15),
	@idempresa_importar char(4),
	@nombre_importar varchar(50),
	@status_importar char(2),

	@idempresa int,
	@ccontable char(15),
	@nombre varchar(50),

	@registro_exists char(15),
	@llave_carga int,
	@count int,
	@reject_message varchar(50)

SET	@object_name='cuenta_contable'
SET	@count = 0

EXECUTE sp_rtrn_sequential
		@object_name,
		@last_key_assigned OUTPUT,
		@locked_status_flag OUTPUT,
		's'
PRINT 'Status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

/* Inicio de lógica cuando bandera indica que objecto está desbloqueado */
IF @locked_status_flag = 0
BEGIN

	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	BEGIN TRANSACTION trans_load_cuenta_contable

	PRINT 'Bloquear object_control:'
	EXECUTE sp_lock_object_control @object_name

	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR
	FOR	SELECT
			ccontable,
			empresa,
			nombre,
			status
		FROM	cuenta_contable_importar
		WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@ccontable_importar,
		@idempresa_importar,
		@nombre_importar,
		@status_importar

/* Bucle de lectura-validación-escritura */
	WHILE @@FETCH_STATUS = 0
	BEGIN
		SET @count = @count + 1
		SET @cuenta_contable_OK = 1
		SET @reject_message=' '
```

```
		PRINT CONVERT(char(8),@count)+' Empresa: '+ @idempresa_importar +' Cuenta
Contable: '+@ccontable_importar

/* Revisar si empresa es válida */
		SET @idempresa =
			(
			SELECT idempresa
			FROM empresa
			WHERE no_empresa = @idempresa_importar
			)

		IF @idempresa IS NULL
		BEGIN
			SET   @cuenta_contable_OK = 0
			SET   @reject_message='EMPRESA '+ @idempresa_importar + ' NO
REGISTRADA'
			PRINT @reject_message
		END

/* Revisar si cuenta contable viene en blanco*/
		IF @ccontable_importar IS NULL
		OR @ccontable_importar = ' '
		BEGIN
			SET   @cuenta_contable_OK = 0
			SET      @reject_message='Cuenta Contable en blanco'
			PRINT   @reject_message
		END


/* Si hay condición de error, escribir mensaje y fecha hora */
		IF @cuenta_contable_OK = 0
		BEGIN
			UPDATE cuenta_contable_importar
			SET      [reject_message]= @reject_message,
				[fecha_ult_act]=getdate()
			WHERE
				ccontable = @ccontable_importar AND
				empresa = @idempresa_importar


		END

/* Escribir registros que pasan validación */

		IF @cuenta_contable_OK = 1
		BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
			UPDATE cuenta_contable_importar
			SET
				[reject_message]= @reject_message,
				[llave_carga]=@llave_carga,
				[fecha_ult_act]=NULL
			WHERE
				ccontable = @ccontable_importar AND
				empresa = @idempresa_importar

/* Cargar o ajustar variables que se van a escribir */
			if @status_importar != 'a'
				begin
					SET @status_importar = 'b'
				end
```

```
/* Revisar si existe */
                        SET @registro_exists =
                                (
                                SELECT ccontable
                                FROM cuenta_contable
                                WHERE idempresa = @idempresa
                                AND ccontable = @ccontable_importar
                                )


                        IF @registro_exists IS NOT NULL
                        BEGIN
                                UPDATE cuenta_contable
                                SET
                                nombre= rtrim(ltrim(@nombre_importar)),
                                idstatus = @status_importar
                                WHERE idempresa = @idempresa
                                AND ccontable = @ccontable_importar
                                PRINT 'Cuenta contable actualizada'
                        END

                        IF @registro_exists IS NULL
                        BEGIN
                                SET @last_key_assigned=@last_key_assigned+1
                                INSERT INTO cuenta_contable
                                        (
                                        idempresa,
                                        ccontable,
                                        nombre,
                                        idStatus
                                        )
                                VALUES          (
                                        @idempresa,
                                        @ccontable_importar,
                                        rtrim(ltrim(@nombre_importar)),
                                        @status_importar
                                        )
                                        PRINT 'Insertado'
                        END

                END

                FETCH cur_1
                INTO
                        @ccontable_importar,
                        @idempresa_importar,
                        @nombre_importar,
                        @status_importar

        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
```

```
		BEGIN
			COMMIT TRANSACTION trans_load_cuenta_contable
			PRINT 'Transacción commited'
		END
		ELSE
		BEGIN
			ROLLBACK TRANSACTION trans_load_cuenta_contable
			PRINT 'Transacción rolled-back'
		END

		EXEC sp_end_of_new_load @llave_carga

/* Fin de lógica cuando bandera indica que el objecto está desbloqueado */
END

/* Caso de encontrar objeto bloqueado */
IF @locked_status_flag = 1
BEGIN
		PRINT @object_name+' '+'BLOQUEADO'
END

/* Si la bandera contiene carácter nulo, significa objeto no registrado en object_control */
IF @locked_status_flag IS NULL
BEGIN
		PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_cargar_deposito    Script Date: 02/07/2008 04:36:26 p.m. ******/




/*
Inicia
Nombre:
sp_cargar_deposito



Funcionalidad:
Este procedimiento carga el catalogo de depositos hechos por los empleados a partir de la tabla intermedia de
deposito_impor
```

```
Entradas:
Accion a tomar
	1 = commit
	0 = rollback

Proceso:
Carga la tabla deposito a partir de los datos en deposito_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_deposito 1
Termina
*/

CREATE PROC sp_cargar_deposito
				@commit tinyint = 1
AS
DECLARE
	@last_key_assigned int,
	@object_name varchar(50),
	@locked_status_flag tinyint,

	@deposito_OK int,
	@folio int,
	@no_empresa char(15),
	@empleado int,
	@fecha datetime,
	@idbanco char(5),
	@referencia varchar(20),
	@monto money,
	@moneda char(3),
	@llave_carga int,
	@count int,
	@reject_message varchar(50),

	@idempresa int,
	@numempleado int,
	@banco int,
	@idmoneda int
SELECT
	@object_name='deposito',
	@count = 0
/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON
EXECUTE sp_rtrn_sequential
		@object_name,
		@last_key_assigned OUTPUT,
		@locked_status_flag OUTPUT,
		's'
PRINT 'Status de'+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)
IF @locked_status_flag = 0
BEGIN
	BEGIN TRANSACTION trans_load_deposito
	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	EXEC sp_lock_object_control @object_name
	PRINT 'Object_control ha quedado bloqueado'
	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR
	FOR SELECT
```

```
                folio,
                no_empresa,
                empleado,
                fecha,
                idbanco,
                referencia,
                monto,
                idmoneda
        FROM    deposito_importar
        WHERE llave_carga = 0

        OPEN cur_1
        FETCH cur_1
        INTO
                @folio,
                @no_empresa,
                @empleado,
                @fecha,
                @idbanco,
                @referencia,
                @monto,
                @moneda

/* Bucle de lectura-validación-escritura */
        WHILE @@FETCH_STATUS = 0
        BEGIN
                SET @count = @count + 1
                SET   @deposito_OK = 1
                SET      @reject_message=' '
                PRINT CONVERT(char(8),@count)+' Referencia: '+@referencia

                set @idempresa = (
                        select idempresa
                        from empresa
                        where no_empresa = @no_empresa
                )
                if @idempresa is null or @idempresa = ' '
                begin
                        SET      @deposito_OK = 0
                        SET      @reject_message='Nombre empresa en blanco'
                        PRINT   @reject_message
                end

                set @numempleado = (
                        select numempleado
                        from usuario
                        where    idempresa = @idempresa and
                                 numempleado = @empleado
                )
                if @numempleado is null
                begin
                        SET      @deposito_OK = 0
                        SET      @reject_message='Usuario no registrado: ' + cast(@empleado as char)
                        PRINT   @reject_message
                end

                set @banco = (
                        select idbanco
                        from banco
                        where identificador = @idbanco
                )
```

```
		if @referencia is null
		begin
			SET	@deposito_OK = 0
			SET	@reject_message='Numero de referencia en blanco'
			PRINT	@reject_message
		end

		if @monto is null
		begin
			SET	@deposito_OK = 0
			SET	@reject_message='Monto en blanco'
			PRINT	@reject_message
		end

		set @idmoneda = (
			select idmoneda
			from moneda
			where	identificador = @moneda
		)

		if @idmoneda is null or @idmoneda = ' '
		begin
			SET	@deposito_OK = 0
			SET	@reject_message='Moneda no registrada : ' + @moneda
			PRINT	@reject_message
		end


/* Si hay condición de error, escribir mensaje y fecha hora */
		IF @deposito_OK = 0

		BEGIN
			UPDATE deposito_importar
			SET
				[reject_message]= @reject_message,
				[fecha_ult_act]=getdate()
			WHERE
				folio = @folio

		END
/* Escribir registros que pasan validación */

		IF @deposito_OK = 1
		BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
			UPDATE deposito_importar
			SET	[reject_message]= @reject_message,
				[llave_carga]=@llave_carga,
				[fecha_ult_act]=NULL
			WHERE
				folio = @folio

/* Ajustar variables */
/* Revisar: si existe, sólo actualizar los datos */

			if exists(
				select folio
				from deposito
				where folio = @folio
			)
			begin
```

```
                    update deposito
                    set
                    idempresa = @idempresa,
                    numempleado = @numempleado,
                    fecha = @fecha,
                    idbanco = @banco,
                    referencia = @referencia,
                    monto = @monto,
                    idmoneda = @idmoneda
                    where    folio = @folio

                    print 'Actualizado'
                end
                else
                begin
                    insert into deposito
                    (
                    folio,
                    idempresa,
                    numempleado,
                    fecha,
                    idbanco,
                    referencia,
                    monto,
                    idmoneda
                    )
                    values
                    (
                    @folio,
                    @idempresa,
                    @empleado,
                    @fecha,
                    @banco,
                    @referencia,
                    @monto,
                    @idmoneda
                    )
                    print 'Insertado'
                end

/* Fin de escribir registros que pasan validación (IF @empresa_OK = 1)*/
                END

                FETCH cur_1
                INTO
                    @folio,
                    @no_empresa,
                    @empleado,
                    @fecha,
                    @idbanco,
                    @referencia,
                    @monto,
                    @idmoneda
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned
        EXEC sp_end_of_new_load @llave_carga
        PRINT 'Fin de '+ @object_name +' load '
```

```
	IF @commit = 1
	BEGIN
		COMMIT TRANSACTION trans_load_deposito
		PRINT 'Transacción commited'
	END
	IF @commit <> 1
	BEGIN
		ROLLBACK TRANSACTION trans_load_deposito
		PRINT 'Transacción rolled-back'
	END
END
/* Casos de encontrar algo mal en estado de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
	PRINT @object_name+' '+'BLOQUEADO'
END
IF @locked_status_flag IS NULL
BEGIN
	PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_cargar_empresa    Script Date: 02/07/2008 04:36:24 p.m. ******/




/*
Inicia
Nombre:
sp_cargar_empresa



Funcionalidad:
Este procedimiento carga el catalogo de empresas a partir de la tabla intermedia de empresa_importar

Entradas:
Accion a tomar
	1 = commit
	0 = rollback

Proceso:
Carga la tabla empresa a partir de los datos en empresa_importar
```

```
Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_empresa 1
Termina
*/

CREATE PROC sp_cargar_empresa
                        @commit tinyint = 1
AS
DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,
        @empresa_OK int,
        @no_empresa char(15),
        @Nombre char(100),
        @RFC char(15),
        @direccion varchar(100),
        @colonia varchar(50),
        @delegacionmunicip varchar(40),
        @estado varchar(40),
        @pais varchar(40),
        @cp char(6),
        @telefono varchar(20),
        @fax varchar(20),
        @llave_carga int,
        @idempresa int,
        @mail varchar(100),
        @idstatus char(1),
        @count int,
        @reject_message varchar(50)
SELECT
        @object_name='empresa',
        @count = 0
/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON
EXECUTE sp_rtrn_sequential
                @object_name,
                @last_key_assigned OUTPUT,
                @locked_status_flag OUTPUT,
                's'
PRINT 'Status de'+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)
IF @locked_status_flag = 0
BEGIN
        BEGIN TRANSACTION trans_load_empresa
        EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

        EXEC sp_lock_object_control @object_name
        PRINT 'Object_control ha quedado bloqueado'
        PRINT 'Inicio de validación e inserción en '+@object_name
        DECLARE cur_1 CURSOR
        FOR SELECT
                no_empresa,
                Nombre,
                RFC,
                direccion,
                colonia,
                delegacionmunicip,
                estado,
                pais,
```

```
                cp,
                telefono,
                fax,
                mail,
                idstatus
        FROM    empresa_importar
        WHERE llave_carga = 0

        OPEN cur_1
        FETCH cur_1
        INTO
                @no_empresa,
                @Nombre,
                @RFC,
                @direccion,
                @colonia,
                @delegacionmunicip,
                @estado,
                @pais,
                @cp,
                @telefono,
                @fax,
                @mail,
                @idstatus

/* Bucle de lectura-validación-escritura */
        WHILE @@FETCH_STATUS = 0
        BEGIN
                SET @count = @count + 1
                SET   @empresa_OK = 1
                SET      @reject_message=' '
                PRINT CONVERT(char(8),@count)+' Empresa: '+@Nombre+' RFC: '+@RFC

/* Revisar si el nombre de la empresa es válido */
                IF @Nombre IS NULL
                OR @Nombre = ' '
                BEGIN
                        SET   @empresa_OK = 0
                        SET      @reject_message='Nombre empresa en blanco'
                        PRINT   @reject_message
                END

/* Revisar si el status es activo o baja */
                IF @idstatus != 'a'
                BEGIN
                        SET   @idstatus = 'b'
                END

/* Si hay condición de error, escribir mensaje y fecha hora */
                IF @empresa_OK = 0
                BEGIN
                        UPDATE empresa_importar
                        SET
                                [reject_message]= @reject_message,
                                [fecha_ult_act]=getdate()
                        WHERE
                                no_empresa = @no_empresa
                END
/* Escribir registros que pasan validación */

                IF @empresa_OK = 1
                BEGIN
```

```
/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
                        UPDATE empresa_importar
                        SET     [reject_message]= @reject_message,
                                [llave_carga]=@llave_carga,
                                [fecha_ult_act]=NULL
                        WHERE no_empresa = @no_empresa
/* Ajustar variables */
/* Revisar: si existe, sólo actualizar los datos */
                        SET @idempresa =
                                (
                                SELECT idempresa
                                FROM empresa
                                WHERE no_empresa = @no_empresa
                                )
/* Si ya existe, actualizar */
                        IF @idempresa IS NOT NULL
                        BEGIN

                                UPDATE empresa
                                SET
                                        Nombre=UPPER(@Nombre),
                                        RFC=@RFC,
                                        direccion=@direccion,
                                        colonia=@colonia,
                                        delegacionmunicip=@delegacionmunicip,
                                        estado=@estado,
                                        pais=@pais,
                                        cp=@cp,
                                        telefono=@telefono,
                                        fax=@fax,
                                        mail = @mail,
                                        idstatus = @idstatus
                                WHERE idempresa = @idempresa

                                PRINT 'empresa actualizada'
                        END
/* Si no existe, darla de alta */
                        IF @idempresa IS NULL
                        BEGIN
                                SELECT
                                        @last_key_assigned=@last_key_assigned+1,
                                        @idempresa=@last_key_assigned

                                INSERT INTO empresa
                                        (
                                        idempresa,
                                        Nombre,
                                        RFC,
                                        direccion,
                                        colonia,
                                        delegacionmunicip,
                                        estado,
                                        pais,
                                        cp,
                                        telefono,
                                        fax,
                                        no_empresa,
                                        mail,
                                        idstatus
                                        )
                                VALUES
```

```
                                        (
                                        @idempresa,
                                        UPPER(@Nombre),
                                        @RFC,
                                        @direccion,
                                        @colonia,
                                        @delegacionmunicip,
                                        @estado,
                                        @pais,
                                        @cp,
                                        @telefono,
                                        @fax,
                                        @no_empresa,
                                        @mail,
                                        @idstatus
                                        )
                                        PRINT 'Insertado'
                                END


/* Fin de escribir registros que pasan validación (IF @empresa_OK = 1)*/
                        END

                        FETCH cur_1
                        INTO
                                @no_empresa,
                                @Nombre,
                                @RFC,
                                @direccion,
                                @colonia,
                                @delegacionmunicip,
                                @estado,
                                @pais,
                                @cp,
                                @telefono,
                                @fax,
                                @mail,
                                @idstatus
                END

                CLOSE cur_1
                DEALLOCATE cur_1

                PRINT 'Desbloquear object_control:'
                EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned
                EXEC sp_end_of_new_load @llave_carga
                PRINT 'Fin de '+ @object_name +' load '
                IF @commit = 1
                BEGIN
                        COMMIT TRANSACTION trans_load_empresa
                        PRINT 'Transacción commited'
                END
                IF @commit <> 1
                BEGIN
                        ROLLBACK TRANSACTION trans_load_empresa
                        PRINT 'Transacción rolled-back'
                END
END
/* Casos de encontrar algo mal en estado de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
                PRINT @object_name+' '+'BLOQUEADO'
```

```
END
IF @locked_status_flag IS NULL
BEGIN
	PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_cargar_iva    Script Date: 02/07/2008 04:36:24 p.m. ******/




/*
Inicia
Nombre:
sp_cargar_iva



Funcionalidad:
Este procedimiento carga el catalogo de ivas a partir de la tabla intermedia de iva_importar

Entradas:
Accion a tomar
	1 = commit
	0 = rollback

Proceso:
Carga la tabla iva a partir de los datos en iva_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_iva 1
Termina
*/

CREATE PROC [dbo].sp_cargar_iva
				@commit tinyint = 1

AS

DECLARE
```

```
	@last_key_assigned int,
	@object_name varchar(50),
	@locked_status_flag tinyint,
	@iva_OK int,

	@identificador_importar char(4),
	@ejercicio_importar char(4),
	@porcentaje_importar numeric(10,4),
	@descripcion_importar varchar(50),
	@idstatus_importar char(2),
	@llave_carga int,

	@identificador char(4),
	@ejercicio char(4),
	@porcentaje numeric(10,4),
	@descripcion varchar(50),
	@idstatus char(2),
	@registro_exists char(4),

	@count int,
	@reject_message varchar(50)

SELECT
	@object_name='iva',
	@count = 0

/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON

EXECUTE sp_rtrn_sequential
		@object_name,
		@last_key_assigned OUTPUT,
		@locked_status_flag OUTPUT,
		's'
PRINT 'status de'+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

/* Inicio de lógica si objeto está OK en object control */
IF @locked_status_flag = 0
BEGIN

	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	BEGIN TRANSACTION trans_load_iva

	PRINT 'Bloquear object_control:'
	EXECUTE sp_lock_object_control @object_name

	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR
	FOR	SELECT
			identificador,
			--ejercicio,
			porcentaje,
			descripcion,
			idstatus
		FROM	iva_importar
		WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@identificador_importar,
```

```
                --@ejercicio_importar,
                @porcentaje_importar,
                @descripcion_importar,
                @idstatus_importar

/* Bucle de lectura-validación-escritura */
        WHILE @@FETCH_status = 0
        BEGIN
                SELECT
                        @iva_OK = 1,
                        @reject_message=' '

/* Inicia validación de iva */
/* Revisar si porcentaje viene en blanco */
                IF @porcentaje_importar IS NULL
                BEGIN
                        SELECT
                                @iva_OK = 0,
                                @reject_message='Porcentaje en blanco'
                        PRINT   @reject_message
                END

/* Revisar si código de contabilidad viene en blanco */
                IF @identificador_importar IS NULL
                OR @identificador_importar = ' '
                BEGIN
                        SELECT
                                @iva_OK = 0,
                                @reject_message='Código contable IVA en blanco'
                        PRINT   @reject_message
                END

                IF @idstatus_importar != 'a'
                BEGIN
                        set @idstatus_importar = 'b'
                END

/* Si hay condición de error, escribir mensaje y fecha hora */
                IF @iva_OK = 0
                BEGIN
                        UPDATE iva_importar
                        SET
                                [reject_message]= @reject_message,
                                [fecha_ult_act]=getdate()
                        WHERE
                                identificador = @identificador_importar
                                --AND ejercicio = @ejercicio_importar
                END

/* Si los datos son válidos, escribir iasiva */
                IF @iva_OK = 1
                BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
                        UPDATE iva_importar
                        SET     [reject_message]= @reject_message,
                                [llave_carga]=@llave_carga,
                                [fecha_ult_act]=NULL
                        WHERE           identificador = @identificador_importar
                                --AND ejercicio = @ejercicio_importar

/* Cargar o ajustar variables con las que se va a escribir */
```

```
                                SELECT
                                        @count = @count + 1,
                                        --@ejercicio = CONVERT(int,@ejercicio_importar),
                                        @porcentaje = CONVERT(int,@porcentaje_importar),
                                        @identificador = @identificador_importar

                                PRINT CONVERT(char(8),@count)+ --' '+CONVERT(char(5),@ejercicio)+
                                        ' IVA '+@identificador


/* Revisar si existe */
                                SET @registro_exists =
                                        (
                                        SELECT identificador
                                        FROM iva
                                        WHERE identificador = @identificador
                                        --AND ejercicio = @ejercicio
                                        )

/*      Caso de que ya exista, actualizar */
                                IF @registro_exists IS NOT NULL
                                BEGIN
                                        UPDATE iva
                                        SET
                                        porcentaje=@porcentaje_importar,
                                        identificador = @identificador_importar,
                                        descripcion = @descripcion_importar,
                                        idstatus = @idstatus_importar
                                        WHERE identificador = @identificador_importar
                                        --AND ejercicio = @ejercicio_importar

                                        PRINT 'IVA actualizado'
                                END

/* Caso de que no esté dado de alta, insertar nuevo registro */
                                IF @registro_exists IS NULL
                                BEGIN

                                        SET @last_key_assigned=@last_key_assigned+1
                                        INSERT INTO iva
                                                (
                                                identificador,
                                                --ejercicio,
                                                porcentaje,
                                                descripcion,
                                                idstatus
                                                )
                                        VALUES
                                                (
                                                @identificador_importar,
                                                --@ejercicio_importar,
                                                @porcentaje_importar,
                                                @descripcion_importar,
                                                @idstatus_importar
                                                )
                                        PRINT 'Insertado'
                                END

                END

                FETCH cur_1
```

```
                INTO
                        @identificador_importar,
                        --@ejercicio_importar,
                        @porcentaje_importar,
                        @descripcion_importar,
                        @idstatus_importar

/* Regreso de bucle de lectura de cur_1 */
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
        BEGIN
                COMMIT TRANSACTION trans_load_iva
                PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
                ROLLBACK TRANSACTION trans_load_iva
                PRINT 'Transacción rolled-back'
        END

/* Fin de lógica si objeto está desbloqueado en object control */

        EXEC sp_end_of_new_load @llave_carga
END

/* Casos de encontrar algo mal en estado de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
        PRINT @object_name+' '+'BLOQUEADO'
END
IF @locked_status_flag IS NULL
BEGIN
        PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_cargar_moneda    Script Date: 02/07/2008 04:36:24 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_cargar_moneda



Funcionalidad:
Este procedimiento carga el catalogo de monedas a partir de la tabla intermedia de moneda_importar

Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla moneda a partir de los datos en moneda_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_moneda 1
Termina
*/

CREATE PROC [dbo].sp_cargar_moneda
                        @commit tinyint = 1
AS
DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,
        @moneda_OK int,
        @llave_carga int,

        @identificador char(3),
        @descripcion varchar(50),
        @idstatus char(2),

        @idmoneda int,

        @count int,
        @reject_message varchar(50)
SELECT
        @object_name='moneda',
        @count = 0
/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON
EXECUTE sp_rtrn_sequential
                @object_name,
                @last_key_assigned OUTPUT,
                @locked_status_flag OUTPUT,
                's'
PRINT 'Status de'+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)
IF @locked_status_flag = 0
BEGIN
```

```
	BEGIN TRANSACTION trans_load_moneda
	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	EXEC sp_lock_object_control @object_name
	PRINT 'Object_control ha quedado bloqueado'
	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR
	FOR SELECT
		identificador,
		descripcion,
		idstatus
	FROM	moneda_importar
	WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@identificador,
		@descripcion,
		@idstatus

/* Bucle de lectura-validación-escritura */
	WHILE @@FETCH_STATUS = 0
	BEGIN
		SET @count = @count + 1
		SET @moneda_OK = 1
		SET @reject_message=' '
		PRINT CONVERT(char(8),@count)+' Moneda: '+@descripcion+' Identificador :
'+@identificador
/* Revisar si el nombre de la moneda es válido */
		IF @descripcion IS NULL
		OR @descripcion = ' '
		BEGIN
			SET	@moneda_OK = 0
			SET	@reject_message='Nombre de moneda en blanco'
			PRINT	@reject_message
		END
/* Revisar si el codigo de la moneda es válido */
		IF len(@identificador) <> 3
		BEGIN
			SET	@moneda_OK = 0
			SET	@reject_message='Identificador de la moneda no valido'
			PRINT	@reject_message
		END
/* Revisar si el status de la moneda es válido */
		IF @idstatus != 'a'
		BEGIN
			SET	@idstatus = 'b'
		END

/* Si hay condición de error, escribir mensaje y fecha hora */
		IF @moneda_OK = 0
		BEGIN
			UPDATE moneda_importar
			SET
				[reject_message]= @reject_message,
				[fecha_ult_act]=getdate()
			WHERE
				identificador = @identificador
		END
/* Escribir registros que pasan validación */
```

```
                    IF @moneda_OK = 1
                    BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
                        UPDATE moneda_importar
                        SET     [reject_message]= @reject_message,
                                [llave_carga]=@llave_carga,
                                [fecha_ult_act]=NULL
                        WHERE identificador = @identificador
/* Ajustar variables */
/* Revisar: si existe, sólo actualizar los datos */
                        SET @idmoneda =
                                (
                                SELECT idmoneda
                                FROM moneda
                                WHERE identificador = @identificador
                                )
/* Si ya existe, actualizar */
                        IF @idmoneda IS NOT NULL
                        BEGIN
                                UPDATE moneda
                                SET
                                        descripcion=UPPER(@descripcion),
                                        idstatus=@idstatus
                                WHERE identificador = @identificador

                                PRINT 'moneda actualizada'
                        END
/* Si no existe, darla de alta */
                        IF @idmoneda IS NULL
                        BEGIN
                                SELECT
                                        @last_key_assigned=@last_key_assigned+1,
                                        @idmoneda=@last_key_assigned

                                INSERT INTO moneda
                                        (
                                        idmoneda,
                                        identificador,
                                        descripcion,
                                        idstatus
                                        )
                                VALUES
                                        (
                                        @idmoneda,
                                        @identificador,
                                        @descripcion,
                                        @idstatus
                                        )
                                        PRINT 'Insertado'
                        END



/* Fin de escribir registros que pasan validación (IF @moneda_OK = 1)*/
                    END

                    FETCH cur_1
                    INTO
                        @identificador,
                        @descripcion,
                        @idstatus
```

```
		END

		CLOSE cur_1
		DEALLOCATE cur_1

		PRINT 'Desbloquear object_control:'
		EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned
		EXEC sp_end_of_new_load @llave_carga
		PRINT 'Fin de '+ @object_name +' load '
		IF @commit = 1
		BEGIN
			COMMIT TRANSACTION trans_load_moneda
			PRINT 'Transacción commited'
		END
		IF @commit <> 1
		BEGIN
			ROLLBACK TRANSACTION trans_load_moneda
			PRINT 'Transacción rolled-back'
		END
END
/* Casos de encontrar algo mal en estado de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
		PRINT @object_name+' '+'BLOQUEADO'
END
IF @locked_status_flag IS NULL
BEGIN
		PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_cargar_presupuesto    Script Date: 02/07/2008 04:36:26 p.m. ******/




/*
Inicia
Nombre:
sp_cargar_presupuesto



Funcionalidad:
Este procedimiento carga el presupuesto a partir de la tabla intermedia de presupuesto_importar
```

```
Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla presupuesto a partir de los datos en presupuesto_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_presupuesto 1
Termina
*/

CREATE PROC [dbo].sp_cargar_presupuesto
                        @commit tinyint = 1
AS
DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,
        @presupuesto_OK int,
        @llave_carga int,
        @count int,
        @reject_message varchar(50),

        @idregistro int,
        @empresa char(15),
        @ccontable char(6),
        @anio int,
        @mes int,
        @ppto_presupuestado money,
        @ppto_gastado money,
        @ppto_comprometido money,
        @moneda char(3),
        @centro_costo char(10),

        @idppto int,
        @idempresa int,
        @cuenta_contable char(6),
        @idmoneda int,
        @idccostos int

SELECT
        @object_name='presupuesto',
        @count = 0

/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON
EXECUTE sp_rtrn_sequential
                @object_name,
                @last_key_assigned OUTPUT,
                @locked_status_flag OUTPUT,
                's'
PRINT 'Status de'+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)
IF @locked_status_flag = 0
BEGIN
        BEGIN TRANSACTION trans_load_presupuesto
        EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT
```

```
	EXEC sp_lock_object_control @object_name
	PRINT 'Object_control ha quedado bloqueado'
	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR

	FOR SELECT
		idregistro,
		empresa,
		ccontable,
		anio,
		mes,
		ppto_presupuestado,
		idmoneda,
		centro_costo
	FROM	presupuesto_importar
	WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@idregistro,
		@empresa,
		@ccontable,
		@anio,
		@mes,
		@ppto_presupuestado,
		@moneda,
		@centro_costo

/* Bucle de lectura-validación-escritura */
	WHILE @@FETCH_STATUS = 0
	BEGIN
		SET @count = @count + 1
		SET @presupuesto_OK = 1
		SET @reject_message=' '


		set @idempresa = (
			select idempresa
			from empresa
			where no_empresa = @empresa
		)
		if @idempresa is null or @idempresa = ' '
		begin
			SET	@presupuesto_OK = 0
			SET	@reject_message='Nombre empresa en blanco'
			PRINT	@reject_message
		end

		set @cuenta_contable = (
			select ccontable
			from cuenta_contable
			where	idempresa = @idempresa and
				ccontable = ltrim(rtrim(@ccontable))
		)

		if @cuenta_contable is null
		begin
			SET	@presupuesto_OK = 0
			SET	@reject_message='La cuenta contable ' + ltrim(rtrim(@ccontable)) + ' no
existe'
			PRINT	@reject_message
```

```
		end

		if @anio is null or @anio <= 0
		begin
			SET	@presupuesto_OK = 0
			SET	@reject_message='El anio no es valido'
			PRINT	@reject_message
		end

		if @mes is null or @mes <= 0
		begin
			SET	@presupuesto_OK = 0
			SET	@reject_message='El mes no es valido'
			PRINT	@reject_message
		end

		if @ppto_presupuestado is null or @ppto_presupuestado < 0
		begin
			SET	@ppto_presupuestado = 0
		end

		set @idmoneda = (
			select idmoneda
			from moneda
			where	identificador = ltrim(rtrim(@moneda))
		)
		if @idmoneda is null
		begin
			SET	@presupuesto_OK = 0
			SET	@reject_message='La moneda ' + @moneda + ' no existe en el catalogo
de monedas'
			PRINT	@reject_message
		end

		set @idccostos = (
			select idccostos
			from centro_costo
			where	idempresa = @idempresa and
				identificador = ltrim(rtrim(@centro_costo))
		)
		if @idccostos is null
		begin
			SET	@presupuesto_OK = 0
			SET	@reject_message='El centro de costo ' + @centro_costo + ' no existe en el
catalogo de centros de costo'
			PRINT	@reject_message
		end

/* Si hay condición de error, escribir mensaje y fecha hora */
		IF @presupuesto_OK = 0
		BEGIN
			UPDATE presupuesto_importar
			SET
				[reject_message]= @reject_message,
				[fecha_ult_act]=getdate()
			WHERE
				idregistro = @idregistro
		END
/* Escribir registros que pasan validación */

		IF @presupuesto_OK = 1
		BEGIN
```

```
/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
                        UPDATE presupuesto_importar
                        SET     [reject_message]= @reject_message,
                                [llave_carga]=@llave_carga,
                                [fecha_ult_act]=NULL
                        WHERE idregistro = @idregistro
/* Ajustar variables */
/* Revisar: si existe, sólo actualizar los datos */
                        SET @idppto =
                                (
                                SELECT idppto
                                FROM presupuesto
                                WHERE idppto = @idregistro

                                )
/* Si ya existe, actualizar */
                        IF @idppto IS NOT NULL
                        BEGIN
                                UPDATE presupuesto
                                SET ppto_presupuestado = @ppto_presupuestado
                                WHERE idppto = @idregistro

                                PRINT 'ppto actualizado'
                        END
/* Si no existe, darla de alta */
                        IF @idppto IS NULL
                        BEGIN
                                /*
                                SELECT
                                        @last_key_assigned=@last_key_assigned+1,
                                        @idmoneda=@last_key_assigned
                                */

                                INSERT INTO presupuesto
                                        (
                                        idppto,
                                        idempresa,
                                        ccontable,
                                        anio,
                                        mes,
                                        ppto_presupuestado,
                                        idmoneda,
                                        identidad
                                        )
                                VALUES
                                        (
                                        @idregistro,
                                        @idempresa,
                                        @cuenta_contable,
                                        @anio,
                                        @mes,
                                        @ppto_presupuestado,
                                        @idmoneda,
                                        @idccostos
                                        )
                                        PRINT 'Insertado'
                        END

/* Fin de escribir registros que pasan validación (IF @moneda_OK = 1)*/
                END
```

```
                FETCH cur_1
                INTO
                        @idregistro,
                        @empresa,
                        @ccontable,
                        @anio,
                        @mes,
                        @ppto_presupuestado,
                        @moneda,
                        @centro_costo

        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned
        EXEC sp_end_of_new_load @llave_carga
        PRINT 'Fin de '+ @object_name +' load '
        IF @commit = 1
        BEGIN
                COMMIT TRANSACTION trans_load_presupuesto
                PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
                ROLLBACK TRANSACTION trans_load_presupuesto
                PRINT 'Transacción rolled-back'
        END
END
/* Casos de encontrar algo mal en estado de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
        PRINT @object_name+' '+'BLOQUEADO'
END
IF @locked_status_flag IS NULL
BEGIN
        PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_cargar_proveedor    Script Date: 02/07/2008 04:36:25 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_cargar_proveedor



Funcionalidad:
Este procedimiento carga el catalogo de proveedores a partir de la tabla intermedia de proveedor_importar

Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla proveedor a partir de los datos en proveedor_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_proveedor 1
Termina
*/
CREATE PROC sp_cargar_proveedor
                        @commit tinyint = 1

AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,

        @proveedor_OK tinyint,

        @identificador char(10),
        @nombre char(25),
        @rfc varchar(15),
        @calle varchar(60),
        @numero varchar(10),
        @interior varchar(10),
        @colonia char(30),
        @delegacion char(30),
        @cp char(5),
        @estado char(4),
        @pais char(5),
        @telefono char(30),
        @fax char(30),
        @mail varchar(30),
        @serv_producto varchar(50),
        @observaciones varchar(50),
        @banco char(5),
        @no_cuenta char(20),
        @tipo_pago char(3),
        @llave_carga int,
```

```
	@proveed_exist int,

	@idempresa int,

	@count int,
	@reject_message varchar(50)

/*
	Indicar nombre del objeto en object_control e inicializar variable contadora
	Cargar variables propias del proceso y parámetros
*/
SELECT
	@object_name='proveedor',
	@count = 0

/* Establecer valor variable del sistema, para ver/no ver "xx rows affected" */
SET NOCOUNT ON

EXECUTE sp_rtrn_sequential
		@object_name,
		@last_key_assigned OUTPUT,
		@locked_status_flag OUTPUT,
		's'

PRINT 'Status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

IF @locked_status_flag = 0
BEGIN

	BEGIN TRANSACTION trans_load_Proveedores

	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	EXECUTE sp_lock_object_control @object_name
	PRINT 'Object_control ha sido bloqueado'

	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR
	FOR SELECT
		identificador,
		nombre,
		rfc,
		calle,
		numero,
		interior,
		colonia,
		delegacion,
		cp,
		estado,
		pais,
		telefono,
		fax,
		mail,
		serv_producto,
		observaciones,
		banco,
		no_cuenta,
		tipo_pago
	FROM	proveedor_importar
	WHERE	llave_carga = 0
```

```
	OPEN cur_1
	FETCH cur_1
	INTO
		@identificador,
		@nombre,
		@rfc,
		@calle,
		@numero,
		@interior,
		@colonia,
		@delegacion,
		@cp,
		@estado,
		@pais,
		@telefono,
		@fax,
		@mail,
		@serv_producto,
		@observaciones,
		@banco,
		@no_cuenta,
		@tipo_pago

/* Bucle de (lectura-validación-escritura),(lectura siguiente...) */
	WHILE @@FETCH_STATUS = 0
	BEGIN
		SELECT
			@count = @count + 1,
			@proveedor_OK = 1,
			@reject_message=' '

		PRINT CONVERT(char(8),@count)
			+' Proveedor: '+ @identificador


/* Revisar que nombre no venga en blanco */
		IF @rfc IS NULL
		OR @rfc = ' '
		BEGIN
			SET   @proveedor_OK = 0
			SET      @reject_message='Razón social en blanco'
			PRINT   @reject_message
		END

		set @banco = (
			select idbanco
			from banco
			where identificador = @banco
		)

		if @banco is null or @banco = ' '
		begin
			SET   @proveedor_OK = 0
			SET      @reject_message='El banco no es valido'
			PRINT   @reject_message
		end


/* Si hay condición de error para proveedor_importar, escribir mensaje y fecha hora */
		IF @proveedor_OK = 0
		BEGIN
```

```
                        UPDATE proveedor_importar
                        SET     [reject_message]= @reject_message,
                                [fecha_ult_act]=getdate()
                        WHERE identificador = @identificador
                END

/* Inicio de lógica escribir registros que pasan validación */
                IF @proveedor_OK = 1
                BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
                        UPDATE proveedor_importar
                        SET     [reject_message]= @reject_message,
                                [llave_carga]=@llave_carga,
                                [fecha_ult_act]=NULL
                        WHERE identificador = @identificador


/* Buscar si existe en tabla proveedores */
                        SET @proveed_exist =
                                (
                                SELECT idproveedor
                                FROM proveedor
                                WHERE identificador = @identificador
                                )

/* Si existe, actualizar sus datos */
                        IF @proveed_exist IS NOT NULL
                        BEGIN
                                UPDATE proveedor
                                SET
                                        nombre=@nombre,
                                        rfc=@rfc,
                                        calle=@calle,
                                        numero=@numero,
                                        interior=@interior,
                                        colonia=@colonia,
                                        delegacion=@delegacion,
                                        cp=@cp,
                                        estado=@estado,
                                        pais=@pais,
                                        telefono=@telefono,
                                        fax=@fax,
                                        mail=@mail,
                                        serv_producto=@serv_producto,
                                        observaciones=@observaciones,
                                        idbanco=@banco,
                                        no_cuenta=@no_cuenta,
                                        idtipo_pago=@tipo_pago
                                WHERE           identificador = @identificador

                                PRINT 'Proveedor actualizado'
                        END

                        IF @proveed_exist IS NULL
                        BEGIN
                                SET @last_key_assigned=@last_key_assigned+1
                                INSERT INTO proveedor
                                        (
                                        idproveedor,
                                        nombre,
                                        rfc,
```

```
                                        calle,
                                        numero,
                                        interior,
                                        colonia,
                                        delegacion,
                                        cp,
                                        estado,
                                        pais,
                                        telefono,
                                        fax,
                                        mail,
                                        serv_producto,
                                        observaciones,
                                        idbanco,
                                        no_cuenta,
                                        idtipo_pago,
                                        identificador
                                        )
                                VALUES
                                        (
                                        @last_key_assigned,
                                        @nombre,
                                        @rfc,
                                        @calle,
                                        @numero,
                                        @interior,
                                        @colonia,
                                        @delegacion,
                                        @cp,
                                        @estado,
                                        @pais,
                                        @telefono,
                                        @fax,
                                        @mail,
                                        @serv_producto,
                                        @observaciones,
                                        @banco,
                                        @no_cuenta,
                                        @tipo_pago,
                                        @identificador
                                        )

                                        PRINT 'Insertado'
                        END

/* Fin de lógica escribir registros que pasan validación */
                        END

                        FETCH cur_1
                        INTO
                        @identificador,
                        @nombre,
                        @rfc,
                        @calle,
                        @numero,
                        @interior,
                        @colonia,
                        @delegacion,
                        @cp,
                        @estado,
                        @pais,
                        @telefono,
```

```
                    @fax,
                    @mail,
                    @serv_producto,
                    @observaciones,
                    @banco,
                    @no_cuenta,
                    @tipo_pago

/* Regreso de bucle leer-validar-escribir */
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        EXEC sp_end_of_new_load @llave_carga

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
        BEGIN
                COMMIT TRANSACTION trans_load_Proveedores
                PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
                ROLLBACK TRANSACTION trans_load_Proveedores
                PRINT 'Transacción rolled-back'
        END

END

/* Casos de encontrar algo mal en nivel de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
        PRINT @object_name+' '+'BLOQUEADO'
END
IF @locked_status_flag IS NULL
BEGIN
        PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_cargar_proveedor_empresa    Script Date: 02/07/2008 04:36:26 p.m.
******/
```

```
/*
Inicia
Nombre:
sp_cargar_proveedor_empresa



Funcionalidad:
Este procedimiento carga la relacion de que proveedores pertenecen a que empresa a partir de la tabla
intermedia de proveedo

Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla proveedor_empresa a partir de los datos en proveedor_empresa_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_proveedor_empresa 1
Termina
*/
CREATE PROC [dbo].sp_cargar_proveedor_empresa
                        @commit tinyint = 1
AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,

        @usuario_OK tinyint,

        @no_empresa_importar char(4),
        @proveedor_importar char(10),
        @status_importar char(2),
        @llave_carga int,

        @idempresa int,
        @proveedor int,
        @idstatus char(2),
        @count int,

        @reject_message varchar(50)

/* Cargar valores iniciales de variables de este proceso */
SELECT
        @object_name='proveedor_empresa',
        @count = 0


/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
```

```
SET NOCOUNT ON

/* Obtener último consecutivo asignado y estado del objeto en object_control */
EXECUTE sp_rtrn_sequential
                @object_name,
                @last_key_assigned OUTPUT,
                @locked_status_flag OUTPUT,
                's'

PRINT 'Status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

IF @locked_status_flag = 0
BEGIN

        BEGIN TRANSACTION trans_load_proveedor_empresa

        EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

        EXEC sp_lock_object_control @object_name
        PRINT 'Object_control ha sido bloqueado'

        PRINT 'Inicio de validación e inserción en '+@object_name

        DECLARE cur_1 CURSOR
        FOR SELECT
                no_empresa,
                proveedor,
                status
        FROM    proveedor_empresa_importar
        WHERE llave_carga = 0

        OPEN cur_1
        FETCH cur_1
        INTO
                @no_empresa_importar,
                @proveedor_importar,
                @status_importar

/*
        Bucle de (lectura-validación-escritura),(lectura siguiente...)
*/
        WHILE @@FETCH_STATUS = 0
        BEGIN
                set @usuario_OK = 1

                set @idempresa = (
                        SELECT idempresa
                        FROM empresa
                        WHERE no_empresa = @no_empresa_importar
                )

                if @idempresa is null or @idempresa = ' '
                begin
                        SET     @usuario_OK = 0
                        SET     @reject_message='Empresa no registrada : ' + cast(@idempresa as char)
                        PRINT   @reject_message
                end

                set @proveedor = (
                        SELECT idproveedor
                        FROM proveedor
                        WHERE identificador = @proveedor_importar
```

```
		)

		if @proveedor is null or @proveedor = ' '
		begin
			SET	@usuario_OK = 0
			SET	@reject_message='Proveedor no registrado : ' +
cast(@proveedor_importar as char)
			PRINT	@reject_message
		end

		if @status_importar != 'a'
		begin
			set @status_importar = 'b'
		end


/* Si hay condición de error para empleado_importar, escribir mensaje y fecha hora */
		IF @usuario_OK = 0
		BEGIN
			UPDATE proveedor_empresa_importar
			SET
				[reject_message]= @reject_message,
				[fecha_ult_act]=getdate()
			WHERE		no_empresa = @no_empresa_importar and
				proveedor = @proveedor_importar
		END


/* Inicio de lógica escribir registros que pasan validación */

		IF @usuario_OK = 1
		BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
			UPDATE proveedor_empresa_importar
			SET
				[reject_message]= @reject_message,
				[llave_carga]=@llave_carga,
				[fecha_ult_act]=NULL
			WHERE		no_empresa = @no_empresa_importar and
				proveedor = @proveedor_importar

/* Revisar: si existe, sólo actualizar los datos */

			if exists(
				select
				*
				from proveedor_empresa
				where	idempresa = @idempresa and
					idproveedor = @proveedor
			)
			begin
				update proveedor_empresa
				set
				idstatus = @status_importar
				where	idempresa = @idempresa and
					idproveedor = @proveedor
			end
			else
			begin
				SET @last_key_assigned=@last_key_assigned+1
				INSERT INTO proveedor_empresa
```

```
                                        (
                                        idproveedor,
                                        idempresa,
                                        idstatus
                                        )
                                VALUES
                                        (
                                        @proveedor,
                                        @idempresa,
                                        @status_importar
                                        )

                                        PRINT 'Insertado ' + @proveedor_importar
                        end


/* Fin de lógica escribir registros que pasan validación */
                END

                FETCH cur_1
                INTO
                        @no_empresa_importar,
                        @proveedor_importar,
                        @status_importar

/* Regreso de bucle leer-validar-escribir */
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
        BEGIN
                COMMIT TRANSACTION trans_load_proveedor_empresa
                PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
                ROLLBACK TRANSACTION trans_load_proveedor_empresa
                PRINT 'Transacción rolled-back'
        END

        EXEC sp_end_of_new_load @llave_carga

END

/*
        Casos de encontrar el objeto bloqueado, o algo irregular en object_control
*/

IF @locked_status_flag = 1
        BEGIN
                PRINT @object_name+' '+'BLOQUEADO'
        END
ELSE
        IF @locked_status_flag IS NULL
```

```
	BEGIN
		PRINT 'OBJETO ' + @object_name +
		' NO ESTA REGISTRADO EN TABLA object_control. FIN DEL PROCESO'
	END
	ELSE
		IF @locked_status_flag != 0
		BEGIN
			PRINT @object_name+' '+'BLOQUEADO CON STATUS IRREGULAR'
		END



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_cargar_tipo_cambio    Script Date: 02/07/2008 04:36:25 p.m. ******/



/*
Inicia
Nombre:
sp_cargar_tipo_cambio



Funcionalidad:
Este procedimiento carga el catalogo de tipos de cambio a partir de la tabla intermedia de
tipo_cambio_importar

Entradas:
Accion a tomar
	1 = commit
	0 = rollback

Proceso:
Carga la tabla tipo_cambio a partir de los datos en tipo_cambio_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_tipo_cambio 1
Termina
*/

CREATE PROC [dbo].sp_cargar_tipo_cambio
				@commit tinyint = 1
```

```
AS

DECLARE
	@last_key_assigned int,
	@object_name varchar(50),
	@locked_status_flag tinyint,

	@tipo_cambio_OK int,

	@idmoneda_origen_importar char(3),
	@idmoneda_destino_importar char(3),
	@fecha datetime,
	@tipo_cambio_importar money,
	@llave_carga int,

	@idmoneda_origen int,
	@idmoneda_destino int,

	@tipo_cambio money,

	@registro_exists int,

	@count int,
	@reject_message varchar(50)

SET	@object_name='tipo_cambio'
SET	@count = 0

EXECUTE sp_rtrn_sequential
		@object_name,
		@last_key_assigned OUTPUT,
		@locked_status_flag OUTPUT,
		's'
PRINT 'status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

/* Inicio de lógica si objeto está OK en object control */
IF @locked_status_flag = 0
BEGIN

	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	BEGIN TRANSACTION trans_load_tipo_cambio

	PRINT 'Bloquear object_control:'
	EXECUTE sp_lock_object_control @object_name

	PRINT 'Inicio de validación e inserción en '+@object_name
	DECLARE cur_1 CURSOR
	FOR	SELECT
			idmoneda_origen,
			idmoneda_destino,
			fecha,
			tipo_cambio
		FROM	tipo_cambio_importar
		WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@idmoneda_origen_importar,
		@idmoneda_destino_importar,
```

```
                @fecha,
                @tipo_cambio_importar

/* Bucle de lectura-validación-escritura */
        WHILE @@FETCH_status = 0
        BEGIN
                SET   @tipo_cambio_OK = 1
                SET      @reject_message=' '

/* Inicia validación */
/* Revisar si tipo_cambio viene en blanco */
                IF @tipo_cambio_importar IS NULL
                BEGIN
                        SET   @tipo_cambio_OK = 0
                        SET      @reject_message='tipo_cambio no válido: vacío'
                        PRINT   @reject_message
                END

/* Revisar si la moneda está dada de alta */
                SET @idmoneda_origen = (select idmoneda from moneda where identificador =
@idmoneda_origen_importar)
                IF NOT EXISTS
                        (
                        SELECT idmoneda
                        FROM moneda
                        WHERE identificador = @idmoneda_origen_importar
                        )
                BEGIN
                        SET   @tipo_cambio_OK = 0
                        SET      @reject_message='Moneda origen no registrada ' +
@idmoneda_origen_importar
                        PRINT   @reject_message
                END

                SET @idmoneda_destino = (select idmoneda from moneda where identificador =
@idmoneda_destino_importar)
                IF NOT EXISTS
                        (
                        SELECT idmoneda
                        FROM moneda
                        WHERE identificador = @idmoneda_destino_importar
                        )
                BEGIN
                        SET   @tipo_cambio_OK = 0
                        SET      @reject_message='Moneda destino no registrada ' +
@idmoneda_destino_importar
                        PRINT   @reject_message
                END

/* Si hay condición de error, escribir mensaje y fecha hora */
                IF @tipo_cambio_OK = 0
                BEGIN
                        UPDATE tipo_cambio_importar
                        SET
                                [reject_message]= @reject_message,
                                [fecha_ult_act]=getdate()
                        WHERE
                                idmoneda_origen = @idmoneda_origen_importar AND
                                idmoneda_destino = @idmoneda_destino_importar AND
                                fecha = @fecha
                END
```

```
/* Si los datos son válidos, actualizar o insertar TC1 en A1000500 */
                        IF @tipo_cambio_OK = 1
                        BEGIN

/* Si pasó validación, quitar mensaje de error (si lo hubiera) y ligar a clave de carga */
                                UPDATE tipo_cambio_importar
                                SET     [reject_message]= @reject_message,
                                        [llave_carga]=@llave_carga,
                                        [fecha_ult_act]=NULL
                                WHERE           idmoneda_origen = @idmoneda_origen_importar AND
                                        idmoneda_destino = @idmoneda_destino_importar AND
                                        fecha = @fecha

/* Incrementar contador de ciclos */
                                        SET @count = @count + 1
                                        PRINT CONVERT(char(8),@count)+' Moneda
'+@idmoneda_origen_importar
                                                +' '+CONVERT(char(10),@fecha,6)

/* Revisar si existe */
                                        SET @registro_exists =
                                                (

                                                select  tipo_cambio
                                                from    tipo_cambio
                                                where   idmoneda_origen in (select idmoneda from moneda
where identificador = @idmoneda_origen_importar) AND
                                                        idmoneda_destino in (select idmoneda from moneda
where identificador = @idmoneda_destino_importar) AND
                                                        fecha = @fecha
                                                )

/*          Caso de que ya exista, actualizar */
                                        IF @registro_exists IS NOT NULL
                                        BEGIN
                                                UPDATE tipo_cambio
                                                SET     tipo_cambio=@tipo_cambio_importar
                                                WHERE           idmoneda_origen in (select idmoneda from
moneda where identificador = @idmoneda_origen_importar) AND
                                                        idmoneda_destino in (select idmoneda from moneda
where identificador = @idmoneda_destino_importar) AND
                                                        fecha = @fecha

                                                PRINT 'Actualizada'
                                        END

/* Caso de que no esté dado de alta, insertar nuevo registro */
                                        IF @registro_exists IS NULL
                                        BEGIN
                                                SET @last_key_assigned=@last_key_assigned+1
                                                INSERT INTO tipo_cambio
                                                        (
                                                        idmoneda_origen,
                                                        idmoneda_destino,
                                                        fecha,
                                                        tipo_cambio
                                                        )
                                                VALUES
                                                        (
                                                        @idmoneda_origen,
                                                        @idmoneda_destino,
                                                        @fecha,
```

```
                                        @tipo_cambio_importar
                                        )
                                PRINT 'Insertado'
                        END

                END

                FETCH cur_1
                INTO
                        @idmoneda_origen_importar,
                        @idmoneda_destino_importar,
                        @fecha,
                        @tipo_cambio_importar

/* Regreso de bucle de lectura de cur_1 */
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
        BEGIN
                COMMIT TRANSACTION trans_load_tipo_cambio
                PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
                ROLLBACK TRANSACTION trans_load_tipo_cambio
                PRINT 'Transacción rolled-back'
        END

/* Fin de lógica si objeto está desbloqueado en object control */

        EXEC sp_end_of_new_load @llave_carga
END

/* Casos de encontrar algo mal en estado de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
        PRINT @object_name+' '+'BLOQUEADO'
END
IF @locked_status_flag IS NULL
BEGIN
        PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
```

```
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_cargar_tipo_pago    Script Date: 02/07/2008 04:36:24 p.m. ******/




/*
Inicia
Nombre:
sp_cargar_tipo_pago



Funcionalidad:
Este procedimiento carga el catalogo de tipos de pago a partir de la tabla intermedia de tipo_pago_importar

Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla tipo_pago a partir de los datos en tipo_pago_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_tipo_pago 1
Termina
*/

CREATE PROC [dbo].sp_cargar_tipo_pago
                        @commit tinyint = 1

/* Instancia de SAP default es la instancia de pruebas */

AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,

        @tipo_pago_OK int,

        @idtipopag int,
        @nombre_tipo_pago varchar(50),
        @nombre varchar(50),
        @status char(2),
        @llave_carga int,

        @idtipopag_exist int,

        @count int,
```

```
        @reject_message varchar(50)

SET     @object_name='tipo_pago'
SET     @count = 0


EXECUTE sp_rtrn_sequential
                @object_name,
                @last_key_assigned OUTPUT,
                @locked_status_flag OUTPUT,
                's'
PRINT 'status de'+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

/* Inicio de lógica si objeto está OK en object control */
IF @locked_status_flag = 0
BEGIN

        EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

        BEGIN TRANSACTION trans_load_tipopago

        PRINT 'Bloquear object_control:'
        EXECUTE sp_lock_object_control @object_name

        PRINT 'Inicio de validación e inserción en '+@object_name
        DECLARE cur_1 CURSOR
        FOR     SELECT
                        idtipopag,
                        nombre
                        FROM    tipo_pago_importar
                WHERE llave_carga = 0

        OPEN cur_1
        FETCH cur_1
        INTO
                @idtipopag,
                @nombre_tipo_pago

/* Bucle de lectura-validación-escritura */
        WHILE @@FETCH_status = 0
        BEGIN
                SET   @tipo_pago_OK = 1
                SET   @reject_message=' '

/* Inicia validación de tipopago */
/* Revisar si nombre viene en blanco */
                IF @nombre_tipo_pago IS NULL
                OR @nombre_tipo_pago = ' '
                BEGIN
                        SET   @tipo_pago_OK = 0
                        SET     @reject_message='Nombre tipo de pago en blanco'
                        PRINT   @reject_message
                END

/* Si hay condición de error, escribir mensaje y fecha hora */
                IF @tipo_pago_OK = 0
                BEGIN
                        UPDATE tipo_pago_importar
                        SET     [reject_message]= @reject_message,
                                [fecha_ult_act]=getdate()
                        WHERE idtipopag = @idtipopag
                END
```

```
/* Si los datos son válidos, escribir todos los tipopago para todas las empresas */
		IF @tipo_pago_OK = 1
		BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
				UPDATE tipo_pago_importar
				SET	[reject_message]= @reject_message,
					[llave_carga]=@llave_carga,
					[fecha_ult_act]=NULL
				WHERE idtipopag = @idtipopag

/* Preparar para escritura  */
				SET @count = @count + 1
				PRINT	CONVERT(char(8),@count)+' '+' Tipo pago '+ cast(@idtipopag as char)

/* Cargar o ajustar variables con las que se va a escribir */
				SET @nombre=UPPER(LEFT(@nombre_tipo_pago,50))

/* Revisar vigencia */
				SET @status = 'A'

/* Revisar si existe */
				SET @idtipopag_exist =
					(
					SELECT idtipo_pago
					FROM tipo_pago
					WHERE idtipo_pago = @idtipopag
					)

/*		Caso de q ya exista, actualizar */
					IF @idtipopag_exist IS NOT NULL
					BEGIN
						UPDATE tipo_pago
						SET
						[descripcion]=@nombre
						WHERE idtipo_pago = @idtipopag

						PRINT 'Tipo pago actualizado'
					END

/* Caso de que no esté dado de alta, insertar nuevo registro */
					IF @idtipopag_exist IS NULL
					BEGIN
						SET @last_key_assigned=@last_key_assigned+1
						INSERT INTO tipo_pago
							(
							[idtipo_pago],
							[descripcion]
							)
						VALUES
							(
							@idtipopag,
							@nombre
							)

						PRINT 'Insertado'
					END

		END

		FETCH cur_1
```

```
                INTO
                        @idtipopag,
                        @nombre_tipo_pago

/* Regreso de bucle de lectura de cur_1 */
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
        BEGIN
                COMMIT TRANSACTION trans_load_tipopago
                PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
                ROLLBACK TRANSACTION trans_load_tipopago
                PRINT 'Transacción rolled-back'
        END

/* Fin de lógica si objeto está desbloqueado en object control */

        EXEC sp_end_of_new_load @llave_carga
END

/* Casos de encontrar algo mal en estado de bloqueo del objeto */
IF @locked_status_flag = 1
BEGIN
        PRINT @object_name+' '+'BLOQUEADO'
END
IF @locked_status_flag IS NULL
BEGIN
        PRINT @object_name+' '+'NO ESTA REGISTRADO. FIN DEL PROCESO'
END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_cargar_usuario    Script Date: 02/07/2008 04:36:26 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_cargar_usuario



Funcionalidad:
Este procedimiento carga el catalogo de usuarios a partir de la tabla intermedia de usuario_importar

Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla usuario a partir de los datos en usuario_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_usuario 1
Termina
*/

CREATE PROC [dbo].sp_cargar_usuario
                        @commit tinyint = 1

AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,

        @usuario_OK tinyint,

        @no_empresa_importar char(4),
        @numempleado_importar int,
        @ccostos_importar char(10),
        @nombre_puesto_importar varchar(50),
        @paterno_importar varchar(30),
        @materno_importar varchar(30),
        @nombre_importar varchar(30),
        @rfc_importar varchar(13),
        @banco_importar varchar(15),
        @cta_cheques_importar varchar(20),
        @sucursal_bancaria_importar varchar(10),
        @status_importar char(2),
        @email_importar varchar(50),
        @idjefe_importar int,
        @llave_carga int,

        @idempresa int,
        @numempleado int,
        @idtipo_usuario int,
        @idperfil int,
        @idnivel int,
        @idbanco int,
```

```
	@ccostos int,
	@idstatus char(2),
	@paterno varchar(30),
	@materno varchar(30),
	@nombre varchar(30),
	@rfc varchar(15),
	@puesto varchar(50),
	@cta_cheques varchar(20),
	@clave_acceso char(10),
	@email varchar(50),
	@num_jefe int,

	@idusuario_exist int,

	@count int,

	@reject_message varchar(50)

/* Cargar valores iniciales de variables de este proceso */
SELECT
	@object_name='usuario',
	@count = 0

/*
	Valores por defecto
*/
SELECT
	@idperfil = 2,
	@idnivel = 1,
	@idtipo_usuario = 1,
	@clave_acceso = '1234567890'

/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON

/* Obtener último consecutivo asignado y estado del objeto en object_control */
EXECUTE sp_rtrn_sequential
		@object_name,
		@last_key_assigned OUTPUT,
		@locked_status_flag OUTPUT,
		's'

PRINT 'Status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

IF @locked_status_flag = 0
BEGIN

	BEGIN TRANSACTION trans_load_usuario

	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	EXEC sp_lock_object_control @object_name
	PRINT 'Object_control ha sido bloqueado'

	PRINT 'Inicio de validación e inserción en '+@object_name

	DECLARE cur_1 CURSOR
	FOR SELECT
		no_empresa,
		numempleado,
		ltrim(rtrim(ccostos)),
		nombre_puesto,
```

```
		paterno,
		materno,
		nombre,
		rfc,
		banco,
		no_cuenta,
		status,
		email,
		idjefe
	FROM	usuario_importar
	WHERE	llave_carga = 0



	OPEN cur_1
	FETCH cur_1
	INTO
		@no_empresa_importar,
		@numempleado_importar,
		@ccostos_importar,
		@nombre_puesto_importar,
		@paterno_importar,
		@materno_importar,
		@nombre_importar,
		@rfc_importar,
		@banco_importar,
		@cta_cheques_importar,
		@status_importar,
		@email_importar,
		@idjefe_importar

/*
	Bucle de (lectura-validación-escritura),(lectura siguiente...)
*/
	WHILE @@FETCH_STATUS = 0
	BEGIN

		SELECT
			@count = @count + 1,
			@usuario_OK = 1,
			@idbanco = (select idbanco from banco where identificador = @banco_importar) ,
			@reject_message=' ',
			@idempresa =
				(
				SELECT idempresa
				FROM empresa
				WHERE no_empresa = @no_empresa_importar
				)

		IF @status_importar != 'a'
			BEGIN
				SET @status_importar = 'b'
			END

/* Revisar si ccostos es válido */

		set @ccostos = (
			SELECT idccostos
			FROM centro_costo
			WHERE idempresa = @idempresa and
				ltrim(rtrim(identificador)) = (ltrim(rtrim(@ccostos_importar)))
		)
```

```
		if @ccostos is null
		begin
			SET	@usuario_OK = 0
			SET	@reject_message='Ccostos no válido'
			PRINT	@reject_message
		end

/* Revisar que banco sea válido */
		IF NOT EXISTS
			(
			SELECT idbanco
			FROM banco
			WHERE identificador = @banco_importar
			)
		BEGIN
			SET	@usuario_OK = 0
			SET	@reject_message='Banco no válido'
			PRINT	@reject_message
		END
		else
		begin
			set @idbanco = (
				SELECT idbanco
				FROM banco
				WHERE identificador = @banco_importar
			)
		end

/*
	Quitar guiones de rfc
*/
		SET @rfc = REPLACE (@rfc_importar,'-','')

/* Si hay condición de error para empleado_importar, escribir mensaje y fecha hora */
		IF @usuario_OK = 0
		BEGIN
			UPDATE usuario_importar
			SET
				[reject_message]= @reject_message,
				[fecha_ult_act]=getdate()
			WHERE		no_empresa = @no_empresa_importar
			AND	numempleado = @numempleado_importar
		END


/* Inicio de lógica escribir registros que pasan validación */

		IF @usuario_OK = 1
		BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
			UPDATE usuario_importar
			SET
				[reject_message]= @reject_message,
				[llave_carga]=@llave_carga,
				[fecha_ult_act]=NULL
			WHERE no_empresa = @no_empresa_importar
			AND	numempleado = @numempleado_importar

/* Ajustar variables */
			SET	@puesto=LEFT(@nombre_puesto_importar,15)
```

```
/* Revisar: si existe, sólo actualizar los datos */
                        SET @idusuario_exist =
                                (
                                SELECT numempleado
                                FROM usuario
                                WHERE           idempresa = @idempresa
                                AND     numempleado = @numempleado_importar
                                )

                        IF @idusuario_exist IS NOT NULL
                        BEGIN
                                UPDATE usuario
                                SET
                                        Idbanco=@idbanco,
                                        ccostos=@ccostos,
                                        idstatus=@status_importar,
                                        paterno=@paterno_importar,
                                        materno=@materno_importar,
                                        nombre=@nombre_importar,
                                        rfc=@rfc,
                                        puesto=@nombre_puesto_importar,
                                        cta_cheques=@cta_cheques_importar,
                                        email=@email_importar,
                                        num_jefe=@idjefe_importar
                                WHERE           idempresa = @idempresa
                                AND     Numempleado = @numempleado_importar

                                PRINT 'Usuario actualizado'
                        END

                        IF @idusuario_exist IS NULL
                        BEGIN
                                SET @last_key_assigned=@last_key_assigned+1
                                INSERT INTO usuario
                                        (
                                        idempresa,
                                        numempleado,
                                        idtipo_usuario,
                                        idperfil,
                                        idnivel,
                                        idbanco,
                                        ccostos,
                                        idstatus,
                                        paterno,
                                        materno,
                                        nombre,
                                        rfc,
                                        puesto,
                                        cta_cheques,
                                        clave_acceso,
                                        email,
                                        num_jefe
                                        )
                                VALUES
                                        (
                                        @idempresa,
                                        @numempleado_importar,
                                        1,
                                        @idperfil,
                                        @idnivel,
                                        @idbanco,
```

```
                                        @ccostos,
                                        @status_importar,
                                        @paterno_importar,
                                        @materno_importar,
                                        @nombre_importar,
                                        @rfc,
                                        @nombre_puesto_importar,
                                        @cta_cheques_importar,
                                        [dbo].user_pwd(rtrim(@clave_acceso),1),
                                        @email_importar,
                                        @idjefe_importar
                                        )

                                        PRINT 'Insertado'
                                END

/* Fin de lógica escribir registros que pasan validación */
                        END

                        FETCH cur_1
                        INTO
                                @no_empresa_importar,
                                @numempleado_importar,
                                @ccostos_importar,
                                @nombre_puesto_importar,
                                @paterno_importar,
                                @materno_importar,
                                @nombre_importar,
                                @rfc_importar,
                                @banco_importar,
                                @cta_cheques_importar,
                                @status_importar,
                                @email_importar,
                                @idjefe_importar

/* Regreso de bucle leer-validar-escribir */
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
        BEGIN
                COMMIT TRANSACTION trans_load_usuario
                PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
                ROLLBACK TRANSACTION trans_load_usuario
                PRINT 'Transacción rolled-back'
        END

        EXEC sp_end_of_new_load @llave_carga

END

/*
```

```
	Casos de encontrar el objeto bloqueado, o algo irregular en object_control
*/

IF @locked_status_flag = 1
BEGIN
	PRINT @object_name+' '+'BLOQUEADO'
END ELSE
	IF @locked_status_flag IS NULL
	BEGIN
		PRINT 'OBJETO ' + @object_name +
		' NO ESTA REGISTRADO EN TABLA object_control. FIN DEL PROCESO'
	END ELSE
		IF @locked_status_flag != 0
		BEGIN
			PRINT @object_name+' '+'BLOQUEADO CON STATUS IRREGULAR'
		END




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_cargar_usuario_area_autoriza    Script Date: 02/07/2008 04:36:27
p.m. ******/




/*
Inicia
Nombre:
sp_cargar_usuario_area_autoriza



Funcionalidad:
Este procedimiento carga el catalogo de autorizadores de area por usuario a partir de la tabla intermedia de
usuario_area_au

Entradas:
Accion a tomar
	1 = commit
	0 = rollback

Proceso:
Carga la tabla usuario_area_autoriza a partir de los datos en usuario_area_autoriza_importar

Salida:
```

Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_usuario_area_autoriza 1
Termina
*/

```
CREATE PROC [dbo].sp_cargar_usuario_area_autoriza
                                @commit tinyint = 1
AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,

        @usuario_OK tinyint,

        @no_empresa_importar char(4),
        @numempleado_importar int,
        @autorizador_importar int,
        @llave_carga int,

        @idempresa int,
        @numempleado int,
        @autorizador int,
        @count int,

        @reject_message varchar(50)

/* Cargar valores iniciales de variables de este proceso */
SELECT
        @object_name='usuario_area_autoriza',
        @count = 0


/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON

/* Obtener último consecutivo asignado y estado del objeto en object_control */
EXECUTE sp_rtrn_sequential
                @object_name,
                @last_key_assigned OUTPUT,
                @locked_status_flag OUTPUT,
                's'

PRINT 'Status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

IF @locked_status_flag = 0
BEGIN

        BEGIN TRANSACTION trans_load_usuario_area

        delete from usuario_area_autoriza

        EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

        EXEC sp_lock_object_control @object_name
        PRINT 'Object_control ha sido bloqueado'

        PRINT 'Inicio de validación e inserción en '+@object_name
```

```
	DECLARE cur_1 CURSOR
	FOR SELECT
		no_empresa,
		numempleado,
		autorizador
	FROM	usuario_area_autoriza_importar
	WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@no_empresa_importar,
		@numempleado_importar,
		@autorizador_importar

/*
	Bucle de (lectura-validación-escritura),(lectura siguiente...)
*/
	WHILE @@FETCH_STATUS = 0
	BEGIN
		set @usuario_OK = 1

		set @idempresa = (
			SELECT idempresa
			FROM empresa
			WHERE no_empresa = @no_empresa_importar
		)

		if @idempresa is null or @idempresa = ' '
		begin
			SET	@usuario_OK = 0
			SET	@reject_message='Empresa no registrada : ' + cast(@idempresa as char)
			PRINT	@reject_message
		end


		if not exists(
			select numempleado
			from usuario
			where	numempleado = @numempleado_importar and
				idempresa = @idempresa
		)
		begin
			SET	@usuario_OK = 0
			SET	@reject_message='Usuario no registrado : ' +
cast(@numempleado_importar as char)
			PRINT	@reject_message
		end

		if not exists(
			select numempleado
			from usuario
			where	numempleado = @autorizador_importar and
				idempresa = @idempresa
		)
		begin
			SET	@usuario_OK = 0
			SET	@reject_message='Usuario no registrado : ' + cast(@autorizador_importar
as char)
			PRINT	@reject_message
		end
```

```
		if @numempleado_importar = @autorizador_importar		begin
			SET	@usuario_OK = 0
			SET	@reject_message='Un empleado no puede ser su propio autorizador'
			PRINT	@reject_message
		end



/* Si hay condición de error para empleado_importar, escribir mensaje y fecha hora */
		IF @usuario_OK = 0
		BEGIN
			UPDATE usuario_area_autoriza_importar
			SET
				[reject_message]= @reject_message,
				[fecha_ult_act]=getdate()
			WHERE		no_empresa = @no_empresa_importar and
				numempleado = @numempleado_importar and
				autorizador = @autorizador_importar
		END


/* Inicio de lógica escribir registros que pasan validación */

		IF @usuario_OK = 1
		BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
			UPDATE usuario_area_autoriza_importar
			SET
				[reject_message]= @reject_message,
				[llave_carga]=@llave_carga,
				[fecha_ult_act]=NULL
			WHERE		no_empresa = @no_empresa_importar and
				numempleado = @numempleado_importar and
				autorizador = @autorizador_importar


/* Revisar: si existe, sólo actualizar los datos */

			SET @last_key_assigned=@last_key_assigned+1
			INSERT INTO usuario_area_autoriza
				(
				idempresa,
				numempleado,
				autorizador
				)
			VALUES
				(
				@idempresa,
				@numempleado_importar,
				@autorizador_importar
				)

				PRINT 'Insertado'


/* Fin de lógica escribir registros que pasan validación */
		END

		FETCH cur_1
		INTO
			@no_empresa_importar,
```

```
                        @numempleado_importar,
                        @autorizador_importar

/* Regreso de bucle leer-validar-escribir */
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
        BEGIN
                COMMIT TRANSACTION trans_load_usuario_area
                PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
                ROLLBACK TRANSACTION trans_load_usuario_area
                PRINT 'Transacción rolled-back'
        END

        EXEC sp_end_of_new_load @llave_carga

END

/*
        Casos de encontrar el objeto bloqueado, o algo irregular en object_control
*/

IF @locked_status_flag = 1
        BEGIN
                PRINT @object_name+' '+'BLOQUEADO'
        END
ELSE
        IF @locked_status_flag IS NULL
        BEGIN
                PRINT 'OBJETO ' + @object_name +
                ' NO ESTA REGISTRADO EN TABLA object_control. FIN DEL PROCESO'
        END
        ELSE
                IF @locked_status_flag != 0
                BEGIN
                        PRINT @object_name+' '+'BLOQUEADO CON STATUS IRREGULAR'
                END



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Stored Procedure dbo.sp_cargar_usuario_direccion_autoriza    Script Date: 02/07/2008
04:36:27 p.m. ******/




/*
Inicia
Nombre:
sp_cargar_usuario_direccion_autoriza



Funcionalidad:
Este procedimiento carga el catalogo de autorizadores de direccion por usuario a partir de la tabla intermedia
de usuario_di

Entradas:
Accion a tomar
        1 = commit
        0 = rollback

Proceso:
Carga la tabla usuario_area_autoriza a partir de los datos en usuario_direccion_autoriza_importar

Salida:
Tabla con datos

Ejemplo:
exec [dbo].sp_cargar_usuario_direccion_autoriza 1
Termina
*/

CREATE PROC [dbo].sp_cargar_usuario_direccion_autoriza
                        @commit tinyint = 1
AS

DECLARE
        @last_key_assigned int,
        @object_name varchar(50),
        @locked_status_flag tinyint,

        @usuario_OK tinyint,

        @no_empresa_importar char(4),
        @numempleado_importar int,
        @autorizador_importar int,
        @llave_carga int,

        @idempresa int,
        @numempleado int,
        @autorizador int,
        @count int,

        @reject_message varchar(50)
```

```
/* Cargar valores iniciales de variables de este proceso */
SELECT
	@object_name='usuario_direccion_autoriza',
	@count = 0


/* Que presente (o no presente) mensaje "(1 row(s) affected)": ON = no presentar */
SET NOCOUNT ON

/* Obtener último consecutivo asignado y estado del objeto en object_control */
EXECUTE sp_rtrn_sequential
		@object_name,
		@last_key_assigned OUTPUT,
		@locked_status_flag OUTPUT,
		's'

PRINT 'Status de '+@object_name+' en object_control: '+CONVERT(char(1),@locked_status_flag)

IF @locked_status_flag = 0
BEGIN

	BEGIN TRANSACTION trans_load_usuario_area

	delete from usuario_direccion_autoriza

	EXEC sp_get_new_load_key @object_name, @llave_carga OUTPUT

	EXEC sp_lock_object_control @object_name
	PRINT 'Object_control ha sido bloqueado'

	PRINT 'Inicio de validación e inserción en '+@object_name

	DECLARE cur_1 CURSOR
	FOR SELECT
		no_empresa,
		numempleado,
		autorizador
	FROM	usuario_direccion_autoriza_importar
	WHERE llave_carga = 0

	OPEN cur_1
	FETCH cur_1
	INTO
		@no_empresa_importar,
		@numempleado_importar,
		@autorizador_importar

/*
	Bucle de (lectura-validación-escritura),(lectura siguiente...)
*/
	WHILE @@FETCH_STATUS = 0
	BEGIN
		set @usuario_OK = 1

		set @idempresa = (
			SELECT idempresa
			FROM empresa
			WHERE no_empresa = @no_empresa_importar
		)

		if @idempresa is null or @idempresa = ' '
		begin
```

```
                SET     @usuario_OK = 0
                SET     @reject_message='Empresa no registrada : ' + cast(@idempresa as char)
                PRINT   @reject_message
        end


        if not exists(
                select numempleado
                from usuario
                where   numempleado = @numempleado_importar and
                        idempresa = @idempresa
        )
        begin
                SET     @usuario_OK = 0
                SET     @reject_message='Usuario no registrado : ' +
cast(@numempleado_importar as char)
                PRINT   @reject_message
        end

        if not exists(
                select numempleado
                from usuario
                where   numempleado = @autorizador_importar and
                        idempresa = @idempresa
        )
        begin
                SET     @usuario_OK = 0
                SET     @reject_message='Usuario no registrado : ' + cast(@autorizador_importar
as char)
                PRINT   @reject_message
        end

        if @numempleado_importar = @autorizador_importar
        begin
                SET     @usuario_OK = 0
                SET     @reject_message='Un empleado no puede ser su propio autorizador'
                PRINT   @reject_message
        end



/* Si hay condición de error para empleado_importar, escribir mensaje y fecha hora */
        IF @usuario_OK = 0
        BEGIN
                UPDATE usuario_direccion_autoriza_importar
                SET
                        [reject_message]= @reject_message,
                        [fecha_ult_act]=getdate()
                WHERE           no_empresa = @no_empresa_importar and
                        numempleado = @numempleado_importar and
                        autorizador = @autorizador_importar
        END


/* Inicio de lógica escribir registros que pasan validación */

        IF @usuario_OK = 1
        BEGIN

/* Si pasó validación, quitar mensaje de error y ligar a clave de carga */
                UPDATE usuario_direccion_autoriza_importar
                SET
```

```
                    [reject_message]= @reject_message,
                    [llave_carga]=@llave_carga,
                    [fecha_ult_act]=NULL
                WHERE       no_empresa = @no_empresa_importar and
                    numempleado = @numempleado_importar and
                    autorizador = @autorizador_importar


/* Revisar: si existe, sólo actualizar los datos */

                SET @last_key_assigned=@last_key_assigned+1
                INSERT INTO usuario_direccion_autoriza
                    (
                    idempresa,
                    numempleado,
                    autorizador
                    )
                VALUES
                    (
                    @idempresa,
                    @numempleado_importar,
                    @autorizador_importar
                    )

                    PRINT 'Insertado'


/* Fin de lógica escribir registros que pasan validación */
            END

            FETCH cur_1
            INTO
                @no_empresa_importar,
                @numempleado_importar,
                @autorizador_importar

/* Regreso de bucle leer-validar-escribir */
        END

        CLOSE cur_1
        DEALLOCATE cur_1

        PRINT 'Desbloquear object_control:'
        EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned

        PRINT 'Fin de '+ @object_name +' load '

        IF @commit = 1
        BEGIN
            COMMIT TRANSACTION trans_load_usuario_area
            PRINT 'Transacción commited'
        END
        IF @commit <> 1
        BEGIN
            ROLLBACK TRANSACTION trans_load_usuario_area
            PRINT 'Transacción rolled-back'
        END

        EXEC sp_end_of_new_load @llave_carga

END
```

```
/*
	Casos de encontrar el objeto bloqueado, o algo irregular en object_control
*/

IF @locked_status_flag = 1
	BEGIN
		PRINT @object_name+' '+'BLOQUEADO'
	END
ELSE
	IF @locked_status_flag IS NULL
	BEGIN
		PRINT 'OBJETO ' + @object_name +
		' NO ESTA REGISTRADO EN TABLA object_control. FIN DEL PROCESO'
	END
	ELSE
		IF @locked_status_flag != 0
		BEGIN
			PRINT @object_name+' '+'BLOQUEADO CON STATUS IRREGULAR'
		END



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_centro_costo_autoriza_catalogo    Script Date: 02/07/2008 04:36:27
p.m. ******/




/*
Inicia
Nombre:
sp_centro_costo_autoriza_catalogo



Funcionalidad:
Obtiene el catalogo de autorizadores de centros de costos de una empresa

Entradas:
Empresa en la que se trabaja

Proceso:
Obtiene el catalogo de centros de costo

Salida:
Catalogo de centros de costo
```

```
Ejemplo:
exec [dbo].sp_centro_costo_autoriza_catalogo 1
Termina
*/

CREATE procedure [dbo].sp_centro_costo_autoriza_catalogo
(
	@idempresa int
)
as
begin
	select
	cc.idccostos,
	cc.descripcion,
	ca.numempleado,
	rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as responsable
	from centro_costo as cc
	inner join centro_costo_autoriza as ca on cc.idccostos = ca.idccostos
	inner join usuario as us on ca.numempleado = us.numempleado
	where	cc.idempresa = ca.idempresa and
		ca.idempresa = us.idempresa and
		cc.idempresa = @idempresa
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_centro_costo_catalogo    Script Date: 02/07/2008 04:36:25 p.m.
******/




/*
Inicia
Nombre:
sp_centro_costo_catalogo



Funcionalidad:
Obtiene el catalogo de centros de costos de una empresa

Entradas:
Empresa en la que se trabaja
Datos a presentar
```

```
		a = datos relevantes al administrador
		u = datos relevantes a los usuarios normales

Proceso:
Obtiene el catalogo de centros de costo

Salida:
Catalogo de centros de costo

Ejemplo:
exec [dbo].sp_centro_costo_catalogo 1, 'u'
Termina
*/

CREATE procedure [dbo].sp_centro_costo_catalogo
(
		@idempresa int,
		@proviene char(1)
)
as
begin
		if @proviene = 'u'
		begin
			select
			cc.idccostos,
			rtrim(cc.identificador) as identificador,
			cc.descripcion
			from centro_costo as cc
			where	cc.idempresa = @idempresa and
				cc.idstatus = 'a'
		end

		if @proviene = 'a'
		begin
			select
			cc.idccostos,
			rtrim(cc.identificador) as identificador,
			cc.descripcion,
			st.descripcion_1
			from centro_costo as cc
			left outer join status as st on cc.idstatus = st.idstatus
			where	cc.idempresa = @idempresa

		end
end


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_clave_acceso_inserta_modifica    Script Date: 02/07/2008 04:36:21
p.m. ******/
```

```
/*
Inicia
Nombre:
sp_clave_acceso_inserta_modifica



Funcionalidad:
Modifica las claves de acceso de los usuarios

Entradas:
Empresa en la que se trabaja
Numero de empleado que pretende cambiar su clave de acceso
Clave actual
Clave nueva

Proceso:
Modifica las claves de acceso de los usuarios

Salida:
Clave de acceso actualizada

Ejemplo:
exec [dbo].sp_clave_acceso_inserta_modifica 1,123,'clave_anterior','clave_nueva'
Termina
*/

CREATE procedure [dbo].sp_clave_acceso_inserta_modifica
(
	@idempresa int,
	@numempleado int,
	@clave_actual varchar(10),
	@clave_nueva varchar(10)
)
as
begin
	declare
	@respuesta varchar(250)

	/*
	Se verifica que la clave de acceso exista para ese usuario para esa empresa
	*/
	if exists(
		select
		idempresa
		from usuario
		where	idempresa = @idempresa and
			numempleado = @numempleado and
			clave_acceso = [dbo].user_pwd(rtrim(@clave_actual),1)
	)
	begin
		update usuario
		set
		clave_acceso = [dbo].user_pwd(rtrim(@clave_nueva),1)
		where	idempresa = @idempresa and
			numempleado = @numempleado and
			clave_acceso = [dbo].user_pwd(rtrim(@clave_actual),1)


		set @respuesta = 'Clave de Acceso Actualizada'
```

```
	end
	else
	begin
		set @respuesta = 'La Clave de Acceso no se encontro<br>El registro no se pudo modificar'
	end

	select @respuesta as mensaje

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_compara_tablas    Script Date: 02/07/2008 04:36:21 p.m. ******/



/*
Inicia
Nombre:
sp_compara_tablas



Funcionalidad:
Compara dos tablas. Se utiliza principalmente para evaluar si una solicitud es igual a la solicitud con la que se
le relacio

Entradas:
Folio de la solicitud 1
Folio de la solicitud 2

Proceso:
Evalua si las tablas son iguales

Salida:
1 para decir que si son iguales
0 para decir que son diferentes

Como se utiliza para revisar el flujo que llevaran las solicitudes, cuando una solicitud no es igual a la
relacionada se manda un 0, 'dp', indicando que el flujo de autorizacion de la solicitud sera desde el  principio.
La combinacion 1,'df' significa que las solicitudes son iguales  y que el flujo de autorizacion de la solicitud
relacionada se hara desde el final es decir, nada mas es necesario que autorice el ultimo nivel de autorizacion

Ejemplo:
```

```
exec sp_compara_tablas 1,2
Termina
*/

CREATE PROCEDURE [dbo].sp_compara_tablas
(
	@folio_tabla_1 int,
	@folio_tabla_2 int
)
as
begin
	declare
	@SQL0 varchar(8000),
	@SQL1 varchar(8000),
	@SQL2 varchar(8000)

		set @SQL1 = '
			select
			"Tabla" as tabla,
			de.idconcepto as idconcepto,
			di.ccostos as ccostos,
			sum((isnull(de.subtotal,0) + isnull(de.propina,0))*di.porcentaje/100) as total
			from detalle_solicitud as de
			inner join distribucion_solicitud as di on de.detalle = di.detalle
			where de.folio = ' + rtrim(cast(@folio_tabla_1 as char)) + '
			group by de.idconcepto,di.ccostos
		'
		set @SQL2 = '
			select
			"Tabla" as tabla,
			de.idconcepto as idconcepto,
			di.ccostos as ccostos,
			sum((isnull(de.subtotal,0) + isnull(de.propina,0))*di.porcentaje/100) as total
			from detalle_solicitud as de
			inner join distribucion_solicitud as di on de.detalle = di.detalle
			where de.folio = ' + rtrim(cast(@folio_tabla_2 as char)) + '
			group by de.idconcepto,di.ccostos
		'

		set @SQL0 = @SQL1 + ' UNION ALL ' + @SQL2

		set @SQL0 = '
			if exists(
				SELECT
				tabla,
				idconcepto,
				ccostos,
				total
				FROM ('+@SQL0+') temp
				GROUP BY tabla,idconcepto,ccostos,total
				HAVING COUNT(*) = 1
			)
			begin
				select
				0 as iguales,
				"dp"
			end
			else
			begin
				select
				1 as iguales,
				"df"
```

```
                        end
                '
                exec (@SQL0)

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_concepto_gasto_catalogo    Script Date: 02/07/2008 04:36:27 p.m.
******/



/*
Inicia
Nombre:
sp_concepto_gasto_catalogo


Funcionalidad:
Obtiene el catalogo de conceptos de gasto relacionado a una empresa y a un tipo de operacion. Obtiene el
catalogo de concept

Entradas:
Empresa en la que se trabaja
Tipo de operacion que se solicita
Proviene
        u = proviene del usuario (pagina de solicitudes)
        a = proviene de la administracion (pagina de catalogo de conceptos de gasto)

Proceso:
Obtiene el catalogo de conceptos de gasto

Salida:
Catalogo de conceptos de gasto

Ejemplo:
exec [dbo].sp_concepto_gasto_catalogo 1,'anticipo','u'
Termina
*/

CREATE procedure [dbo].sp_concepto_gasto_catalogo
(
        @idempresa int,
```

```
	@tipo_operacion varchar(50),
	@proviene char(1),
	@idproveedor int = 0
)
as
begin
	declare
	@idtipo_operacion int

	if @proviene = 'u'
	begin
		set @idtipo_operacion = (select
[dbo].fn_obtiene_tipo_operacion(@idempresa,@tipo_operacion))

		select
		cg.idconcepto,
		cg.descripcion
		from concepto_gasto as cg
		inner join concepto_gasto_tipo_concepto as ct on cg.idconcepto = ct.idconcepto
		inner join tipo_operacion as tp on ct.idtipo_operacion = tp.idtipo_operacion
		inner join tipo_concepto as tc on ct.idtipo_concepto = tc.idtipo_concepto
		where	cg.idempresa = ct.idempresa and
			ct.idempresa = tc.idempresa and
			tc.idempresa = tp.idempresa and
			cg.idempresa = @idempresa and
			cg.idstatus = 'a' and
			tp.idtipo_operacion = @idtipo_operacion and
			tp.idstatus = 'a'
		order by cg.descripcion asc
	end

	if @proviene = 'a'
	begin
		select
		cg.idconcepto,
		cg.descripcion,
		cg.idstatus,
		st.descripcion_1,
		cg.porcentaje_deducible,
		cg.ccontable_deducible,
		cg.porcentaje_no_deducible,
		cg.ccontable_no_deducible
		from concepto_gasto as cg
		left outer join status as st on cg.idstatus = st.idstatus
		where cg.idempresa = @idempresa
		order by cg.descripcion asc
	end

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
```

```
GO


/****** Object:  Stored Procedure dbo.sp_concepto_gasto_inserta_modifica    Script Date: 02/07/2008 04:36:27
p.m. ******/




/*
Inicia
Nombre:
sp_concepto_gasto_inserta_modifica



Funcionalidad:
Inserta, actualiza o modifica conceptos de gasto

Entradas:
Identificador del concepto, si es que se va a hacer actualizacion
Empresa en la que se esta trabajando
Nombre del concepto de gasto
Status
        a = activo
        b = baja
Porcentaje deducible del concepto
Cuenta contable deducible del concepto
Porcentaje no deducible del concepto
Cuenta contable no deducible
Accion a seguir
        i = insertar
        u = actualizar
        d = eliminar

Proceso:
Inserta, actualiza o elimina un concepto de gasto

Salida:
Tabla de concepto de gasto actualizada

Ejemplo:
exec [dbo].sp_concepto_gasto_inserta_modifica 2,1,'actualizar todo','a',100,'123456',0,'789012',8,1,3,'i'
Termina
*/

CREATE procedure [dbo].sp_concepto_gasto_inserta_modifica
(
        @id int,
        @idempresa int,
        @descripcion varchar(50),
        @idstatus char(2),
        @porcentaje_deducible decimal(8,5),
        @ccontable_deducible char(6),
        @porcentaje_no_deducible decimal(8,5),
        @ccontable_no_deducible char(6),

        @proviene char(1)
)
```

```
as
begin
	declare
		@idconcepto int


	begin transaction concepto_gasto

	if @ccontable_deducible = 0 or @porcentaje_deducible = 0
	begin
		set @ccontable_deducible = null
	end

	if @ccontable_no_deducible = 0 or @porcentaje_no_deducible = 0
	begin
		set @ccontable_no_deducible = null
	end

	if @proviene = 'i'
	begin
		set @idconcepto = (select isnull(max(idconcepto),0) + 1 from concepto_gasto)

		insert into concepto_gasto
		(
		idconcepto,
		idempresa,
		descripcion,
		idstatus,
		porcentaje_deducible,
		ccontable_deducible,
		porcentaje_no_deducible,
		ccontable_no_deducible
		)
		values
		(
		@idconcepto,
		@idempresa,
		@descripcion,
		@idstatus,
		@porcentaje_deducible,
		@ccontable_deducible,
		@porcentaje_no_deducible,
		@ccontable_no_deducible
		)
	end

	if @proviene = 'u'
	begin
		update concepto_gasto
		set
		descripcion = @descripcion,
		idstatus = @idstatus,
		porcentaje_deducible = @porcentaje_deducible,
		ccontable_deducible = @ccontable_deducible,
		porcentaje_no_deducible = @porcentaje_no_deducible,
		ccontable_no_deducible = @ccontable_no_deducible
		where	idconcepto = @id and
			idempresa = @idempresa

		set @idconcepto = @id
	end
```

```
	if @proviene = 'd'
	begin
		delete from concepto_gasto
		where	idconcepto = @id and
			idempresa = @idempresa
	end

	/*
	Si un concepto se elimina se elimina tambien la relacion que tiene con los
	tipos de concepto
	*/
	delete from concepto_gasto_tipo_concepto
	where	idconcepto = @idconcepto and
		idempresa = @idempresa

	/*
	y con las areas de responsabilidad a la que pertenece
	*/
	delete from concepto_gasto_area
	where	idconcepto = @idconcepto and
		idempresa = @idempresa

	select @idconcepto

	commit transaction concepto_gasto


end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_concepto_gasto_muestra_datos    Script Date: 02/07/2008 04:36:27
p.m. ******/




/*
Inicia
Nombre:
sp_concepto_gasto_muestra_datos


Funcionalidad:
Muestra los datos particulares de un concepto de gasto
```

Entradas:
Numero de la empresa en la que se esta trabajando
Concepto de gasto

Proceso:
Presenta los datos particulares de un concepto de gasto

Salida:
Numero del concepto de gasto
Descripcion del concepto de gasto
Porcentaje deducible
Cuenta contable deducible
Porcentaje no deducible
Cuenta contable no deducible
Tipo del concepto de gasto
Area de responsabilidad a la que pertenece el concepto de gasto

Ejemplo:
exec sp_concepto_gasto_muestra_datos 1,2
Termina
*/

```
CREATE procedure [dbo].sp_concepto_gasto_muestra_datos
(
	@idempresa int,
	@idconcepto int
)
as
begin
	select top 1
	cg.idconcepto,
	cg.descripcion,
	cg.porcentaje_deducible,
	cg.ccontable_deducible,
	cg.porcentaje_no_deducible,
	cg.ccontable_no_deducible,
	case
		when ct.idtipo_concepto is null then 0
		else ct.idtipo_concepto
	end as idtipo_concepto,
	case
		when ca.idarea is null then 0
		else ca.idarea
	end as idarea
	from concepto_gasto as cg
	left outer join concepto_gasto_tipo_concepto as ct on cg.idconcepto = ct.idconcepto and
cg.idempresa = ct.idempresa
	left outer join concepto_gasto_area as ca on cg.idconcepto = ca.idconcepto and cg.idempresa =
ca.idempresa
	where	cg.idempresa = @idempresa and
		cg.idconcepto = @idconcepto
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
```

```
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_contabilidad_autoriza_catalogo    Script Date: 02/07/2008 04:36:27
p.m. ******/




/*
Inicia
Nombre:
sp_contabilidad_autoriza_catalogo



Funcionalidad:
Obtiene el catalogo de autorizadores de contabilidad de una empresa

Entradas:
Empresa en la que se trabaja

Proceso:
Obtiene el catalogo de autorizadores de contabilidad

Salida:
Catalogo de autorizadores de contabilidad

Ejemplo:
exec [dbo].sp_centro_costo_autoriza_catalogo 1
Termina
*/

CREATE procedure [dbo].sp_contabilidad_autoriza_catalogo
(
	@idempresa int
)
as
begin
	select
	cc.numempleado,
	rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as responsable,
	cc.rol
	from contabilidad_autoriza as cc
	inner join usuario as us on cc.numempleado = us.numempleado
	where	cc.idempresa = us.idempresa and
		cc.idempresa = @idempresa
end



GO
SET QUOTED_IDENTIFIER OFF
```

```
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_convertir_a_fecha_sist_contable    Script Date: 02/07/2008 04:36:21 p.m. ******/


CREATE PROC sp_convertir_a_fecha_sist_contable
                                @fecha_entrada datetime,
                                @fecha_contable char(10) OUTPUT
AS

SELECT @fecha_contable = CONVERT( char(10),@fecha_entrada,103 )


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_convertir_d_fecha_sist_contable    Script Date: 02/07/2008 04:36:22 p.m. ******/
```

```
CREATE PROC sp_convertir_d_fecha_sist_contable
                                @fecha_entrada char(10),
                                @fecha_convertida datetime OUTPUT
AS

SELECT @fecha_convertida = CONVERT( datetime,@fecha_entrada,103 )









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_costo_kilometro_catalogo    Script Date: 02/07/2008 04:36:25 p.m.
******/







/*
Inicia
Nombre:
sp_costo_kilometro_catalogo



Funcionalidad:
Obtiene el costo por kilometro por moneda

Entradas:
Empresa en la que se esta trabajando
Moneda de la que se quiere obtener el costo por kilometro

Proceso:
Obtiene el costo por kilometro de una o varias monedas
```

```
Salida:
Informacion de costos por kilometro

Ejemplo:
exec [dbo].sp_costo_kilometro_catalogo 1,0
Termina
*/

CREATE procedure [dbo].sp_costo_kilometro_catalogo
(
	@idempresa int,
	@idmoneda int
)
as
begin
	if @idmoneda = 0
	begin
		select
		mn.idmoneda,
		mn.descripcion,
		ck.costo_por_km
		from costo_por_km as ck
		inner join moneda as mn on ck.idmoneda = mn.idmoneda
		where	ck.idempresa = @idempresa and
			mn.idstatus = 'a'
	end
	else
	begin
		select
		mn.idmoneda,
		round(ck.costo_por_km,2)
		from costo_por_km as ck
		inner join moneda as mn on ck.idmoneda = mn.idmoneda
		where	ck.idempresa = @idempresa and
			mn.idstatus = 'a' and
			ck.idmoneda = @idmoneda
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_costo_kilometro_inserta_modifica    Script Date: 02/07/2008 04:36:25
p.m. ******/
```

```
/*
Inicia
Nombre:
sp_costo_kilometro_inserta_modifica



Funcionalidad:
Actualiza el catalogo de costo por kilometro para una o varias monedas

Entradas:
Empresa en la que se esta trabajando
Moneda de la que se quiere actualizar el costo por kilometro
Costo por kilometro
Accion a seguir
        i = modificar solo para la moneda especificada
        m = modificar para todas las monedas

Proceso:
Inserta el costo por kilometro de una o varias monedas

Salida:
Tabla de costos por kilometro actualizada

Ejemplo:
exec [dbo].sp_costo_kilometro_inserta_modifica 1,1,3.45,'u'
Termina
*/

CREATE procedure [dbo].sp_costo_kilometro_inserta_modifica
(
        @idempresa int,
        @idmoneda int,
        @costo money,
        @proviene char(1) = 'm'    -- i = (Un registro, solo el de la moneda especificada)
                                   -- m = (Multiples registros, el de la moneda especificada se toma como
base
                                   --          y se calculan los de las demas monedas)
)
as
begin
        declare
        @c_idempresa int,
        @c_idmoneda int,
        @c_costo money

        if @proviene = 'i'
        begin
                if exists(
                        select *
                        from costo_por_km
                        where   idempresa = @idempresa and
                                idmoneda = @idmoneda
                )
                begin
                        update costo_por_km
                        set
                        costo_por_km = @costo
                        where   idempresa = @idempresa and
                                idmoneda = @idmoneda
                end
```

```
		else
		begin
			insert into costo_por_km
			(
			idempresa,
			idmoneda,
			costo_por_km
			)
			values
			(
			@idempresa,
			@idmoneda,
			@costo
			)
		end
	end

	/*
	Si se quieren actualizar varias monedas, se manda a llamar este mismo procedimiento
	tantas veces como monedas existan
	*/
	if @proviene = 'm'
	begin
		declare cursor_1 cursor for
			select
			@idempresa as idempresa,
			@idmoneda as idmoneda_origen,
			idmoneda as idmoneda,
			[dbo].fn_obtiene_tipo_cambio(@idmoneda,idmoneda,getdate(),@costo) as costo
			from moneda
			where idstatus = 'a'

		open cursor_1

		fetch next from cursor_1
			into
				@c_idempresa,
				@idmoneda,
				@c_idmoneda,
				@c_costo

		while @@fetch_status = 0
			begin
				if @c_costo != 0
				begin
					exec [dbo].sp_costo_kilometro_inserta_modifica
@c_idempresa,@c_idmoneda,@c_costo,'i'
				end

				fetch next from cursor_1
				into
					@c_idempresa,
					@idmoneda,
					@c_idmoneda,
					@c_costo
			end

		close		cursor_1
		deallocate	cursor_1

	end
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_cuenta_contable_catalogo    Script Date: 02/07/2008 04:36:25 p.m.
******/



/*
Inicia
Nombre:
sp_cuenta_contable_catalogo


Nota:
Este procedimiento se obtiene el catalogo de cuentas contables vigentes por empresa

Funcionalidad:
Obtiene las cuentas contables vigentes por empresa

Entradas:
Numero de empresa en la que se esta trabajando
Datos a presentar
        u = datos relevantes a los usuarios
        a = datos relevantes para los administradores

Proceso:
Obtencion de las cuentas contables vigentes por empresa

Salida:
Datos de las cuentas contables

Ejemplo:
exec sp_cuenta_contable_catalogo 1,'u'
Termina
*/

CREATE procedure [dbo].sp_cuenta_contable_catalogo
(
        @idempresa int,
        @proviene char(1)
)
as
begin
```

```
	if @proviene = 'u'
	begin
		select
		cc.ccontable,
		cc.nombre
		from cuenta_contable as cc
		where	cc.idempresa = @idempresa and
			cc.idstatus = 'a'
	end

	if @proviene = 'a'
	begin
		select
		cc.ccontable,
		cc.nombre,
		st.descripcion_1
		from cuenta_contable as cc
		left outer join status as st on cc.idstatus = st.idstatus
		where	cc.idempresa = @idempresa
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_delete_old_load_records    Script Date: 02/07/2008 04:36:23 p.m.
******/







CREATE PROC sp_delete_old_load_records
				@days_b4_today smallint = 365
AS

DECLARE
	@del_date datetime

/* Restar días */
SET @del_date = DATEADD ( dy,-@days_b4_today,getdate() )

DELETE FROM carga
WHERE date_time_started IS NOT NULL
```

```
AND date_time_started < @del_date




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/*
Inicia
Nombre:
sp_detalle_boleto_avion_inserta_modifica

Funcionalidad:
Inserta, modifica o elimina un registro de boleto de avion

Entradas:
Detalle de la solicitud
Origen
Destino
Aerolinea
Hora de salida
Opcion de accion
        (i = inserta)
        (u = actualiza)
        (d = elimina)

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta, actualiza o elimina

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
exec sp_detalle_boleto_avion_inserta_modifica 1,'Mexico', 'Villahermosa','Aviacsa','17.00 hrs','u'
Termina
*/



CREATE procedure [dbo].sp_detalle_boleto_avion_inserta_modifica
(
        @detalle int,
        @origen varchar(50),
        @destino varchar(50),
```

```
	@aerolinea varchar(50),
	@hora_salida varchar(20),
	@proviene char(1)
)
as
begin

	if @proviene = 'i'
	begin
		insert into detalle_boleto_avion
		(
		detalle,
		origen,
		destino,
		aerolinea,
		hora_salida
		)
		values
		(
		@detalle,
		@origen,
		@destino,
		@aerolinea,
		@hora_salida
		)
	end

	if @proviene = 'u'
	begin
		update detalle_boleto_avion set
		origen = @origen,
		destino = @destino,
		aerolinea = @aerolinea,
		hora_salida = @hora_salida
		where detalle = @detalle
	end

	if @proviene = 'd'
	begin
		delete from detalle_boleto_avion where detalle = @detalle
	end
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_detalle_deposito_inserta_modifica    Script Date: 02/07/2008
04:36:30 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_detalle_deposito_inserta_modifica



Funcionalidad:
Inserta, modifica o elimina un registro de deposito a favor de la empresa

Entradas:
Detalle de la solicitud
Folio del deposito
Opcion de accion
        (i = inserta)
        (u = actualiza)
        (d = elimina)

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta, actualiza o elimina

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
exec sp_detalle_deposito_inserta_modifica 1,2,'d'
Termina
*/


CREATE procedure [dbo].sp_detalle_deposito_inserta_modifica
(
        @detalle int,
        @folio_deposito int,
        @proviene char(1)
)
as
begin
        if @proviene = 'i'
        begin
                insert into detalle_deposito
                (
                detalle,
                folio_deposito
                )
                values
                (
                @detalle,
                @folio_deposito
                )
        end

        if @proviene = 'u'
        begin
                update detalle_deposito set
                folio_deposito = @folio_deposito
                where detalle = @detalle
```

```
	end

	if @proviene = 'd'
	begin
		delete from detalle_deposito where detalle = @detalle
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_detalle_dias_inserta_modifica    Script Date: 02/07/2008 04:36:30
p.m. ******/




/*
Inicia
Nombre:
sp_detalle_dias_inserta_modifica



Funcionalidad:
Inserta, modifica o elimina un registro de dias

Entradas:
Detalle de la solicitud
Numero de dias
Opcion de accion
	(i = inserta)
	(u = actualiza)
	(d = elimina)

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta, actualiza o elimina

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
exec sp_detalle_dias_inserta_modifica 1,2,'d'
Termina
*/
```

```
CREATE procedure [dbo].sp_detalle_dias_inserta_modifica
(
	@detalle int,
	@dias int,
	@proviene char(1)
)
as
begin
	if @proviene = 'i'
	begin
		insert into detalle_dias
		(
		detalle,
		dias
		)
		values
		(
		@detalle,
		@dias
		)
	end

	if @proviene = 'u'
	begin
		update detalle_dias set
		dias = @dias
		where detalle = @detalle
	end

	if @proviene = 'd'
	begin
		delete from detalle_dias where detalle = @detalle
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_detalle_kilometro_inserta_modifica    Script Date: 02/07/2008
04:36:30 p.m. ******/




/*
```

```
Inicia
Nombre:
sp_detalle_kilometro_inserta_modifica



Funcionalidad:
Inserta, modifica o elimina un registro de viaje

Entradas:
Detalle de la solicitud
Identificador del registro del viaje
Numero de viajes para ese origen/destino
Opcion de accion
        (i = inserta)
        (u = actualiza)
        (d = elimina)

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta, actualiza o elimina

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
exec sp_detalle_kilometro_inserta_modifica 1,1,2.5,'d'
Termina
*/


CREATE procedure [dbo].sp_detalle_kilometro_inserta_modifica
(
        @detalle int,
        @idkm int,
        @num_viajes float,
        @proviene char(1)
)
as
begin
        if @proviene = 'i'
        begin
                insert into detalle_kilometro
                (
                detalle,
                idkm,
                num_viajes
                )
                values
                (
                @detalle,
                @idkm,
                @num_viajes
                )
        end

        if @proviene = 'u'
        begin
                update detalle_kilometro set
                idkm = @idkm,
                num_viajes = @num_viajes
                where detalle = @detalle
        end
```

```
	if @proviene = 'd'
	begin
		delete from detalle_kilometro where detalle = @detalle
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_detalle_precio_unitario_unidades_inserta_modifica    Script Date:
02/07/2008 04:36:30 p.m. ******/




/*
Inicia
Nombre:
sp_detalle_precio_unitario_unidades_inserta_modifica



Funcionalidad:
Inserta, modifica o elimina un registro de precio y unidades de un articulo

Entradas:
Detalle de la solicitud
Precio unitario
Numero de unidades
Opcion de accion
	(i = inserta)
	(u = actualiza)
	(d = elimina)

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta, actualiza o elimina

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
exec sp_detalle_precio_unitario_unidades_inserta_modifica 1,2356.89,1,'u'
Termina
*/
```

```
CREATE procedure [dbo].sp_detalle_precio_unitario_unidades_inserta_modifica
(
	@detalle int,
	@precio_unitario money,
	@unidades float,
	@proviene char(1)
)
as
begin
	if @proviene = 'i'
	begin
		insert into detalle_precio_unitario_unidades
		(
		detalle,
		precio_unitario,
		unidades
		)
		values
		(
		@detalle,
		@precio_unitario,
		@unidades
		)
	end

	if @proviene = 'u'
	begin
		update detalle_precio_unitario_unidades set
		precio_unitario = @precio_unitario,
		unidades = @unidades
		where detalle = @detalle
	end

	if @proviene = 'd'
	begin
		delete from detalle_precio_unitario_unidades where detalle = @detalle
	end
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_detalle_solicitud_inserta_modifica    Script Date: 02/07/2008 04:36:29
p.m. ******/
```

```
/*
Inicia
Nombre:
sp_detalle_solicitud_inserta_modifica



Funcionalidad:
Inserta, modifica o elimina un detalle

Entradas:
Identificador del registro del detalle de la solicitud
Empresa en la que se trabaja
Concepto de gasto
Comentario acerca del detalle
Fecha de gasto
Subtotal del detalle
IVA
Identificador del IVA utilizado
Propina
TUA
total
tarjeta_corporativa
incumplimiento de politicas
Opcion de accion
        (i = inserta)
        (u = actualiza)
        (d = elimina)
Variable commit

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta, actualiza o elimina

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
Termina
*/


CREATE procedure [dbo].sp_detalle_solicitud_inserta_modifica
(
        @id int,
        @idempresa int,
        @idconcepto int,
        @comentario varchar(250),
        @fecha_gasto char(11),
        @subtotal money,
        @iva money,
        @idiva char(3),
        @propina money,
        @tua money,
        @total money,
        @tarjeta_corporativa tinyint,
        @inc_pol tinyint,

        @proviene char(1),
        @commit tinyint = 1
```

```
)
as
begin
	declare
	@detalle int

	begin transaction trans

	if @proviene = 'i'
	begin
		set @detalle = (select [dbo].fn_detalle_obtiene_folio_consecutivo())

		set dateformat dmy insert into detalle_solicitud
		(
		detalle,
		folio,
		idempresa,
		idconcepto,
		comentario,
		fecha_gasto,
		subtotal,
		iva,
		idiva,
		propina,
		tua,
		total,
		tarjeta_corporativa,
		inc_pol
		)
		values(
		@detalle,
		@id,
		@idempresa,
		@idconcepto,
		upper(left(@comentario,250)),
		[dbo].sp_convierte_texto_a_fecha(@fecha_gasto),
		@subtotal,
		@iva,
		@idiva,
		@propina,
		@tua,
		@total,
		@tarjeta_corporativa,
		@inc_pol
		)

		select @detalle as detalle

	end

	if @proviene = 'u'
	begin
		set dateformat dmy
		update detalle_solicitud
		set
		comentario = upper(left(@comentario,250)),
		fecha_gasto = [dbo].sp_convierte_texto_a_fecha(@fecha_gasto),
		subtotal = @subtotal,
		iva = @iva,
		idiva = @idiva,
		propina = @propina,
		tua = @tua,
```

```
                total = @total,
                tarjeta_corporativa = @tarjeta_corporativa,
                inc_pol = @inc_pol
                where detalle = @id
        end

        if @proviene = 'd'
        begin
                delete from detalle_solicitud where detalle = @id
        end

        if @commit = 1
        begin
                commit transaction trans
                print 'Transaction commited'

        end

        if @commit <> 1
        begin
                rollback transaction trans
                print 'Transaction rolled-back'
        end



end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_distancias_catalogo    Script Date: 02/07/2008 04:36:25 p.m. ******/




/*
Inicia
Nombre:
sp_distancias_catalogo


Funcionalidad:
Obtiene el catalogo de distancias de una empresa
```

```
Entradas:
Empresa en la que se trabaja
Moneda de la solicitud, para obtener los costos por kilometro para esa moneda

Proceso:
Obtiene el catalogo de distancias

Salida:
Catalogo de distancias

Ejemplo:
exec [dbo].sp_distancias_catalogo 12,3
Termina
*/

CREATE procedure [dbo].sp_distancias_catalogo
(
	@idempresa int,
	@idmoneda int
)
as
begin
	select
	km.idkm,
	km.origen,
	km.destino,
	km.distancia,
	case
		when ck.costo_por_km is null then 0
		else ck.costo_por_km
	end as costo_por_km
	from kilometro as km
	left outer join costo_por_km as ck on km.idempresa = ck.idempresa and ck.idmoneda = @idmoneda
	where	km.idempresa = @idempresa
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_distribucion_inserta_modifica    Script Date: 02/07/2008 04:36:30
p.m. ******/




/*
```

```
Inicia
Nombre:
sp_distribucion_inserta_modifica



Funcionalidad:
Inserta o elimina la distribucion de un detalle entre los diferentes centros de costo

Entradas:
Identificador del registro de distribucion
Numero del detalle
Centro de costo
Porcentaje a distribuir
Folio de la solicitud al que pertenece el detalle
Empresa a la que pertenece el detalle
Numero empleado que hace la solicitud
Opcion de accion
        (i = inserta)
        (d = elimina)
Variable commit

Proceso:
Dependiendo de la variable @proviene, el procedimiento inserta o elimina

Salida:
Operacion de insercion o eliminacion

Ejemplo:
exec sp_distribucion_inserta_modifica 0,2,1,12,14,12,0,'i'
Termina
*/



CREATE procedure [dbo].sp_distribucion_inserta_modifica
(
        @id int,
        @detalle int,
        @ccostos int,
        @porcentaje float,
        @folio int,
        @idempresa int,
        @numempleado int,
        @proviene char(1),

        @commit tinyint = 1
)
as begin

        declare
        @distribucion int,
        @excede_100 tinyint,
        @tiene_dueno tinyint,
        @error tinyint,
        @existe tinyint,
        @porcentaje_acumulado float,
        @mensaje varchar(250)

        begin transaction trans
```

```
	/*Si se pretende insertar un registro nuevo*/
	if @proviene = 'i'
	begin
		set @existe = 0

		if @numempleado <> 0
		begin
			/*
			Cuando se inserta por primera vez, se hace la distribucion al centro de costos
			por default del empleado
			*/
			set @ccostos =
([dbo].fn_obtiene_ccostos_por_empleado(@idempresa,@numempleado))
		end

		/*
		Si no es primera insercion
		*/
		set @error = 0
		/*
		Se busca si la nueva distribucion no causa que se rebase el 100%
		Ya que un detalle no puede estar distribuido mas que el 100%
		*/
		set @excede_100 = (select
[dbo].fn_revisa_excede_distribucion_por_detalle(@detalle,@porcentaje))
		/*
		Se busca que el centro de costo tenga dueno
		*/
		set @tiene_dueno = (select [dbo].fn_revisa_ccostos_autoriza(@idempresa,@ccostos))
		/*
		Se busca si esa distribucion existe
		*/
		set @existe = (
			select 1
			from distribucion_solicitud
			where	ccostos = @ccostos and
				idempresa = @idempresa and
				detalle = @detalle
		)

		if @existe = 1
		begin
			set @existe = 1
		end
		else
		begin
			set @existe = 0
		end

		if @tiene_dueno = 0
		begin
			set @error = 1
			set @mensaje = 'El centro de costo no tiene dueño, favor de notificar al
administrador'
		end

		if @excede_100 = 1
		begin
			set @error = 1
			set @mensaje = 'No se puede distribuir mas del 100%'
		end
```

```
		if @error = 0
		begin
			/*
			Si la distribucion no existe entonces se inserta
			*/
			if @existe = 0
			begin
				set @distribucion = (select
[dbo].fn_distribucion_obtiene_folio_consecutivo())

				set dateformat dmy
				insert into distribucion_solicitud
				(
				distribucion,
				detalle,
				ccostos,
				porcentaje,
				folio,
				idempresa
				)
				values(
				@distribucion,
				@detalle,
				@ccostos,
				@porcentaje,
				@folio,
				@idempresa
				)
				select ''
			end
			/*
			Si la distribucion ya existe entonces se actualiza la distribucion
			*/
			if @existe = 1
			begin
				set @porcentaje_acumulado = (
					select isnull(sum(porcentaje),0)
					from distribucion_solicitud
					where	ccostos=@ccostos and
						idempresa = @idempresa and
						detalle=@detalle
				)

				update distribucion_solicitud
				set porcentaje = @porcentaje + @porcentaje_acumulado
				where	ccostos=@ccostos and
					idempresa = @idempresa and
					detalle=@detalle
				select ''
			end
		end
		else
		begin
			select @mensaje
		end

	end

	/*
	EN ESTE PROCEDIMIENTO NO EXISTE LA ACTUALIZACION PORQUE SE HACE EN LA OPCION
DE INSERTAR
	CUANDO UNA DISTRIBUCION A UN CENTRO DE COSTO YA EXISTE
```

```
	*/

	if @proviene = 'd'
	begin
		if @id = 0
		begin
			delete from distribucion_solicitud where detalle = @detalle
		end
		if @id <> 0
		begin
			delete from distribucion_solicitud where distribucion = @id
		end

	end


	if @commit = 1
	begin
		commit transaction trans
		print 'Transaction commited'

	end

	if @commit <> 1
	begin
		rollback transaction trans
		print 'Transaction rolled-back'
	end
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_elemento_ha_sido_utilizado    Script Date: 02/07/2008 04:36:22 p.m.
******/




/*
Inicia
Nombre:
sp_elemento_ha_sido_utilizado


Funcionalidad:
```

Evalua si un elemento ha sido utilizado dentro de la aplicacion. Se usa para saber, por ejemplo, si el concepto de gasto X h

Entradas:
El elemento a buscar
La tabla en la que hay que buscar
Las condiciones de busqueda

Proceso:
Regresa cadena = true, si el elemento ha sido usado
Regresa cadena = false, si el elemento no ha sido usado

Salida:
Catalogo de distancias

Ejemplo:
exec [dbo].sp_elemento_ha_sido_utilizado 'numempleado','usuario','numempleado=100000 and idempresa=0'
Termina
*/

```
CREATE procedure [dbo].sp_elemento_ha_sido_utilizado
(
	@id varchar(1000),
	@tabla varchar(1000),
	@where varchar(1000)
)
as
begin
	declare
	@sql varchar(4000)

	set @sql = '
		select '+ ltrim(rtrim(cast(@id as char)))+ '
		from ' + ltrim(rtrim(@tabla))+ '
		where ' + ltrim(rtrim(@where))+ '
	'

	set @sql = '
		declare
		@existe varchar(5)

		if exists('+rtrim(ltrim(@sql))+')
		begin
			set @existe = "true"
		end
		else
		begin
			set @existe = "false"
		end

		select @existe
		'

	exec(@sql)

end



GO
SET QUOTED_IDENTIFIER OFF
```

```
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_elimina_ppto_solicitud    Script Date: 02/07/2008 04:36:30 p.m. ******/




/*
Inicia
Nombre:
sp_elimina_ppto_solicitud



Funcionalidad:
Se elimina la relacion del detalle con la tabla de ppto

Entradas:
Folio de la solicitud de la que se quiere eliminar la relacion

Proceso:
Elimina de la tabla (ppto_detalle) todos los detalles que corresponden a la solicitud mencionada

Salida:
Relacion de los detalles de la solicitud con el ppto comprometido o gastado eliminada

Ejemplo:
exec [dbo].sp_elimina_ppto_solicitud 2
Termina
*/

CREATE procedure [dbo].sp_elimina_ppto_solicitud
(
	@folio int
)
as
begin
	declare
	@numempleado int

	set @numempleado = (select numempleado from solicitud where folio = @folio)

	delete from ppto_detalle
	where detalle in (
		select detalle
		from detalle_solicitud
		where folio = @folio
	)

end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_empresa_catalogo    Script Date: 02/07/2008 04:36:24 p.m. ******/



/*
Inicia
Nombre:
sp_empresa_catalogo


Funcionalidad:
Obtiene el catalogo de empresas. A cada tipo de usuario se le presenta el tipo de informacion de su interes

Entradas:
Proveniencia de la petiticion
        u = peticion de usuario normal
        a = peticion de usuario administrador

Proceso:
Obtencion de las empresas

Salida:
Datos de las empresas

Ejemplo:
exec sp_empresa_catalogo 'a'
Termina
*/

CREATE procedure [dbo].sp_empresa_catalogo
(
        @proviene char(1)
)
as
begin
        if @proviene = 'u'
        begin
                select
                em.idempresa,
                em.nombre
                from empresa as em
```

```
		where	em.idstatus = 'a'
	end

	if @proviene = 'a'
	begin
		select
		em.idempresa,
		em.nombre,
		em.telefono,
		lower(em.mail) as email,
		em.no_empresa,
		st.descripcion_1
		from empresa as em
		left outer join status as st on em.idstatus = st.idstatus
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_empresa_detalle_catalogo    Script Date: 02/07/2008 04:36:24 p.m.
******/



/*
Inicia
Nombre:
sp_empresa_detalle_catalogo



Funcionalidad:
Obtiene el detalle de cada empresa

Entradas:
Identificador de la empresa de la que se quiere obtener el detalle

Proceso:
Obtencion del detalle de la empresa en cuestion

Salida:
Datos de las empresas

Ejemplo:
exec sp_empresa_detalle_catalogo 1
Termina
```

```
*/


CREATE procedure [dbo].sp_empresa_detalle_catalogo
(
	@idempresa int
)
as
begin
	select
	em.no_empresa,
	em.nombre,
	em.rfc,
	em.direccion,
	em.colonia,
	em.delegacionmunicip,
	em.estado,
	em.pais,
	em.cp,
	em.telefono,
	em.fax,
	lower(em.mail) as email,
	em.logo,
	st.descripcion_1
	from empresa as em
	left outer join status as st on em.idstatus = st.idstatus
	where idempresa = @idempresa

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_end_of_new_load    Script Date: 02/07/2008 04:36:23 p.m. ******/









CREATE PROC sp_end_of_new_load
		@load_key int = 0
AS
UPDATE		carga
SET	[date_time_ended]=GETDATE()
```

```
WHERE llave_carga = @load_key




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_existe_perfil_objeto    Script Date: 02/07/2008 04:36:26 p.m. ******/



/*
Inicia
Nombre:
sp_existe_perfil_objeto



Funcionalidad:
Revisa si el perfil de usuario existe para una u otra condicion. Ver del codigo del store procedure para mas
detalle

Entradas:
Empresa en la que se esta trabajando
Numero de empleado o identificador del perfil, depende de la opcion seleccionada
Identificador del objeto del menu
Accion a seguir
        nu = si se quiere averiguar si ese empleado tiene acceso a ese objeto del menu
        pe = si se quiere averiguar si ese perfil puede acceder esa opcion del menu

Proceso:
Evalua una de las dos opciones

Salida:
Regresa 1 si la condicion se cumple
Regresa 0 si la condicion no se cumple

Ejemplo:
exec [dbo].sp_existe_perfil_objeto 1,100000,1,'nu'
Termina
```

```
*/

CREATE procedure [dbo].sp_existe_perfil_objeto
(
	@idempresa int,
	@id int,
	@idobjeto int,
	@proviene char(2)
)
as
begin
	declare
	@idperfil int,
	@sql varchar(1000)

	if @proviene = 'nu'
	begin
		set @idperfil = (select idperfil from usuario where idempresa = @idempresa and
numempleado = @id)
		set @sql = '
			select
			po.idobjeto
			from perfil_objeto as po
			inner join objeto as ob on po.idobjeto = ob.idobjeto
			left outer join tipo_operacion as tp on lower(rtrim(ob.descripcion)) =
lower(rtrim(tp.nombre)) and po.idempresa = tp.idempresa
			where	po.idempresa = ' + rtrim(cast(@idempresa as char))+ ' and

				po.idperfil =  ' + rtrim(cast(@idperfil  as char))+ ' and
				po.idobjeto =  ' + rtrim(cast(@idobjeto  as char))+ ' and
				(
					(
						tp.nombre is not null and tp.idstatus = ''a''
					)
					or
					(
						tp.nombre is null and
						po.idobjeto in	(
									select idobjeto
									from objeto as ob
									inner join objeto_ubicacion as
ou on ob.idubicacion = ou.idubicacion
									where	ou.descripcion !=
''solicitudes''
								)
					)

				)
		'
	end

	if @proviene = 'pe'
	begin
		set @sql = '
			select
			idobjeto
			from perfil_objeto
			where	idempresa = ' + rtrim(cast(@idempresa as char))+ ' and

				idperfil =  ' + rtrim(cast(@id as char))+ ' and
				idobjeto =  ' + rtrim(cast(@idobjeto  as char))+ '
		'
```

```
	end

	set @sql = '
		if exists(
			'+ rtrim(@sql) +'
		)
		begin
			select 1
		end
		else
		begin
			select 0
		end

	'
	exec(@sql)
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_flujo_autoriza    Script Date: 02/07/2008 04:36:27 p.m. ******/




/*
Inicia
Nombre:
sp_flujo_autoriza



Funcionalidad:
Asigna a la solicitud el status que le corresponde

Entradas:
Numero del empleado autorizador
Empresa en la que se trabaje
Status que se autoriza
Folio que se autoriza
Identificador del tipo de operacion

Proceso:
Asigna a la solicitud el status que le corresponde

Salida:
Solicitud actualizada o no actualizada

Ejemplo:
```

```
exec sp_flujo_autoriza 123,1,'nu',1,4,0
Termina
*/

CREATE procedure [dbo].sp_flujo_autoriza
(
	@numempleado int,
	@idempresa int,
	@idstatus char(2),
	@folio int,
	@idtipo_operacion int,
	@disponibilidad tinyint
)
as
begin
	declare
	@cf int,			--Variable donde se almacenara el numero de empleados que faltan por
autorizar
					--la solicitud

	@idstatus_sig char(2)	--Status siguiente que le corresponde a la solicitud de acuerdo al flujo

	/*
	Con este procedimiento se obtienen cuantos autorizadores faltan por autorizar ese status.
	El resultado se guarda en la variable @cf (cuantos faltan)
	*/

	exec sp_flujo_autoriza_cuenta_autorizadores
@numempleado,@idempresa,@idstatus,@folio,'cf',@disponibilidad,@cf output

	/*
	Si ya no falta nadie por autorizar entonces se actualiza la solicitud al status correspondiente
	Este proceso se ejecutara hasta que un status todavia tenga autorizaciones pendientes
	*/

	if @idstatus = 'ep' and @disponibilidad = 1
	begin
		set @cf = 0
		if not exists(
			select top 1 '1'
			from historial_solicitud as hs
			where	hs.folio = @folio and
				hs.idhistorial > [dbo].fn_flujo_autoriza_obtiene_ultimo_id_rechazo(@folio)
and
				hs.idstatus = 'ep'
		)
		begin
			declare
			@solicitante int

			set @solicitante = (select numempleado from solicitud where folio = @folio)
			exec sp_registra_historial @folio,@solicitante,@idstatus,'Presupuesto disponible'
		end
	end

	--print 'para el status ' +@idstatus + ' faltan ' + cast(@cf as char)

	if @cf <= 0
	begin
		set @idstatus_sig = (
			select idstatus_sig
			from flujo_autoriza
```

```
                        where   idempresa = @idempresa and
                                idstatus = @idstatus and
                                idtipo_operacion = @idtipo_operacion and
                                idstatus_sig is not null
                )
                if @idstatus_sig is not null
                begin
                        update solicitud set idstatus = @idstatus_sig where folio = @folio
                        exec sp_flujo_autoriza
@numempleado,@idempresa,@idstatus_sig,@folio,@idtipo_operacion,@disponibilidad
                end
        end
        /*
        Si todavia faltan autorizadores por autorizar ese status,
        no se hace nada y la solicitud mantiene el status original
        */
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_flujo_autoriza_cuenta_autorizadores    Script Date: 02/07/2008
04:36:30 p.m. ******/



/*
Inicia
Nombre:
sp_flujo_autoriza_cuenta_autorizadores


Funcionalidad:
Cuenta cuantos autorizadores quedan pendientes por autorizar la solicitud

Entradas:
Numero del empleado autorizador
Empresa en la que se trabaje
Status que se autoriza
Folio que se autoriza
Opcion de resultados
        (ca = cuantos tienen que autorizar la solicitud)
        (ch = cuantos han autorizado la solicitud)
        (cf = cuantos faltan por autorizar la solicitud)
Resultado de la opcion anterior
```

```
Proceso:
   ###########################################################################################

Salida:
Numero entero

Ejemplo:
declare
@faltan int
set @faltan = 0
exec sp_flujo_autoriza_cuenta_autorizadores_anticipo 1,3,'at',1,'cf', @faltan output
print 'Resultado: ' + cast(@faltan as char)
Termina
*/

CREATE procedure [dbo].sp_flujo_autoriza_cuenta_autorizadores
(
	@numempleado int,	--Empleado que autoriza la solicitud
	@idempresa int,		--Empresa en la que se esta trabajando
	@idstatus char(2),	--Status que se autoriza
	@folio int,		--Folio de la solicitud
	@opcion char(2),--Opcion de resultado
	@disponibilidad tinyint,
	@res int output		--Resultado que depende de la opcion que se elija
)
as
begin
	declare
	@ca int,	--Cuantos tienen que autorizar la solicitud
	@ch int,	--Cuantos han autorizado la solicitud
	@cf int,		--Cuantos faltan por autorizar la solicitud
	@ultimo_id int		--Id despues del ultimo rechazo

	/*
	Se crea una tabla temporal para guardar los autorizadores. Posteriormente se contara
	el numero de registros en esta tabla y esos seran "Los que tienen que autorizar"
	la solicitud (@ca)
	*/
	create table #temporal (autorizador int)

	/*
	Si se esta autorizando el status ep, los autorizadores son los responsables de presupuesto
	de los centros de costo a los que se haya distribuido esa solicitud
	Se utiliza distinct en la consulta porque puede haber empleados que son dueños de mas de un
	centro de costo
	*/
	if @idstatus = 'ep'
	begin
		if @disponibilidad = 1
		begin
			insert into #temporal
			select @numempleado
		end
		else
		begin
			insert into #temporal
				select distinct ca.numempleado
				from distribucion_solicitud as di
				inner join centro_costo as ce on rtrim(di.ccostos) = rtrim(ce.idccostos)
				inner join centro_costo_autoriza as ca on rtrim(ce.idccostos) =
rtrim(ca.idccostos)
```

```
                        where   di.folio = @folio and
                                di.idempresa = ce.idempresa and
                                ce.idempresa = ca.idempresa and
                                di.idempresa = @idempresa
        end
end

/*
Si se esta autorizando el status ej, el autorizador solo es uno: el jefe del solicitante
*/
if @idstatus = 'ej'
begin
        insert into #temporal
        select distinct 1

end

/*
Si se esta autorizando el status ea, hay dos tipos de autorizadores
Los autorizadores del concepto de gasto y los autorizadores del usuario
Los autorizadores del concepto de gasto son aquellos que son los responsables del area
a la que pertenezca un concepto de gasto
Los autorizadores del usuario son aquellos definidos para autorizarle a ese usuario
cada vez que este haga una solicitud
*/
if @idstatus = 'ea'
begin
        insert into #temporal
        select aa.numempleado
        from detalle_solicitud as de
        inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
        inner join concepto_gasto_area as ca on cg.idconcepto = ca.idconcepto
        inner join area_responsabilidad as ar on ca.idarea = ar.idarea
        inner join area_responsabilidad_autoriza as aa on ar.idarea = aa.idarea
        inner join usuario as us on aa.numempleado = us.numempleado
        where   de.idempresa = cg.idempresa and
                cg.idempresa = ca.idempresa and
                ca.idempresa = ar.idempresa and
                ar.idempresa = aa.idempresa and
                aa.idempresa = us.idempresa and
                cg.idstatus = 'a' and
                ar.idstatus = 'a' and
                us.idstatus = 'a' and
                de.folio = @folio and
                de.idempresa = @idempresa
        union
        select ua.autorizador
        from solicitud as an
        inner join usuario as u1 on an.numempleado = u1.numempleado
        inner join usuario_area_autoriza as ua on u1.numempleado = ua.numempleado
        inner join usuario as u2 on ua.autorizador = u2.numempleado
        where   an.idempresa = u1.idempresa and
                u1.idempresa = ua.idempresa and
                ua.idempresa = u2.idempresa and
                an.idempresa = @idempresa and
                u2.idstatus = 'a' and
                an.folio = @folio
end

/*
Si se esta autorizando el status ed, el autorizador es el director del area a la
que pertenezca el empleado
```

```
	*/
	if @idstatus = 'ed'
	begin
		insert into #temporal
		select ua.autorizador
		from solicitud as an
		inner join usuario as u1 on an.numempleado = u1.numempleado
		inner join usuario_direccion_autoriza as ua on u1.numempleado = ua.numempleado
		inner join usuario as u2 on ua.autorizador = u2.numempleado
		where	an.idempresa = u1.idempresa and
			u1.idempresa = ua.idempresa and
			ua.idempresa = u2.idempresa and
			an.idempresa = @idempresa and
			u2.idstatus = 'a' and
			an.folio = @folio
	end

	/*
	Si se esta autorizando el status ec, solo hay un autorizador: el responsable de contabilidad
	Aunque haya varios responsables de contabilidad, la autorizacion del primero se tomara como valida
	*/
	if @idstatus = 'ec'
	begin
		insert into #temporal
		select distinct 1
	end

	/*
	El status at no tiene autorizadores, ya que es el ultimo status antes de ser tomado por la interface
	de contabilidad. Pero se tiene que especificar que tiene un autorizador para evitar que se quede en
este
	status y que pueda ser tomado por la interfase de contabilidad
	*/
	if @idstatus = 'at'
	begin
		insert into #temporal
		select distinct 1
	end

	/*
	Se guarda en la variable @ca el numero de autorizadores que tienen que autorizar
	Este numero se obtiene haciendo la cuenta de los registros en la tabla #temporal
	*/
	set @ca = (select count(autorizador) from #temporal where autorizador != 0)
	drop table #temporal

	/*
	Se almacena en la variable @ultimo_id el idhistorial del ultimo rechazo de esa solicitud
	Con esto se tiene un control de validar siempre los registro mas actuales
	*/
	set @ultimo_id = (select [dbo].fn_flujo_autoriza_obtiene_ultimo_id_rechazo(@folio))


	/*
	Se almacena en la variable @ch el numero de autorizadores que han autorizado la solicitud
	*/
	set @ch = (	select count(distinct hd.numempleado_delega) as quienes_han_autorizado
			from historial_solicitud as ha
			inner join historial_delegacion_autoridad_solicitud as hd on ha.idhistorial =
hd.idhistorial
			where	ha.folio = @folio and
				ha.idstatus = @idstatus and
```

```
                                        ha.idhistorial > @ultimo_id

                )
        /*
        Operacion matematica para obtener "Cuantos Faltan por autorizar"
        */
        set @cf = @ca - @ch

        /*
        Asigna a @res el valor correspondiente a la opcion deseada
        Cuantos tienen que autorizar
        */
        if @opcion = 'ca'
        begin
                set @res = @ca
        end
        /*
        Cuantos tienen han autorizado
        */
        if @opcion = 'ch'
        begin
                set @res = @ch
        end
        /*
        Cuantos faltan por autorizar
        */
        if @opcion = 'cf'
        begin
                set @res = @cf
        end

        --print 'Para el status: ' + @idstatus
        --print '------------------------'
        --print 'Tienen que autorizar: ' + rtrim(cast(@ca as char))
        --print 'Han autorizado: ' + rtrim(cast(@ch as char))
        --print 'Faltan por autorizar: ' + rtrim(cast(@cf as char))

        return(@res)
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_flujo_autoriza_empleado_juega_rol    Script Date: 02/07/2008
04:36:22 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_flujo_autoriza_empleado_juega_rol



Funcionalidad:
Obtiene si el usuario en cuestion o quienes le han delegado su autoridad pueden autorizar solicitudes con el
status dado

Entradas:
Empresa en la que se esta trabajando
Numempleado que autoriza
Status que se pretende autorizar

Proceso:
Revisa si el empleado esta facultado para autorizar el status asignado

Salida:
1 = si esta autorizado
0 = no esta autorizado

Ejemplo:
exec [dbo].sp_flujo_autoriza_empleado_juega_rol 1,100000,'ed'
Termina
*/


CREATE procedure [dbo].sp_flujo_autoriza_empleado_juega_rol
(
	@idempresa int,
	@numempleado int,
	@idstatus char(2)
)
as
begin
	declare
	@existe tinyint

	set @existe = 0

	/*
	Se crea tabla temporal que almacenara al autorizador y quienes le delegaron su autoridad
	*/
	create table #temporal_1 (autorizador int,idempresa int)
	insert into #temporal_1
	select @numempleado,@idempresa
	union
	select numempleado_delega,@idempresa
	from transferencia_autoridad
	where	numempleado_delegado = @numempleado
		and idempresa = @idempresa
		and fecha_ini <= [dbo].fn_obtiene_fecha_hoy(getdate())
		and fecha_fin >= [dbo].fn_obtiene_fecha_hoy(getdate())
		and idstatus = 'a'
	/*
	Si el status que se esta autorizando es ep, los autorizadores son los responsables del
	presupuesto de los centros de costo entre los que se haya distribuido la solicitud
	Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
	estar en la tabla temporal_2
	Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
```

```
y no haber autorizado con anterioridad
*/
if @idstatus = 'ep'
	begin
		if exists(
			select
			ca.numempleado
			from centro_costo as ce
			inner join centro_costo_autoriza as ca on ce.idccostos = ca.idccostos
			inner join #temporal_1 as te on ca.numempleado = te.autorizador
			inner join usuario as us on te.autorizador = us.numempleado
			where	ce.idempresa = ca.idempresa and
				ca.idempresa = te.idempresa and
				te.idempresa = us.idempresa and
				ce.idempresa = @idempresa and
				ce.idstatus = 'a' and
				us.idstatus = 'a'
		)
		begin
			set @existe = 1
		end
		else
		begin
			set @existe = 0
		end
	end
/*
Si el status que se esta autorizando es ej, el autorizador es el jefe inmediato del solicitante
Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
estar en la tabla temporal_2
Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
y no haber autorizado con anterioridad
*/
if @idstatus = 'ej'
	begin
		if exists(
			select
			us.num_jefe
			from usuario as us
			inner join #temporal_1 as te on us.num_jefe = te.autorizador

			where	us.idempresa = te.idempresa and
				us.idempresa = @idempresa and
				us.idstatus = 'a'
		)
		begin
			set @existe = 1
		end
		else
		begin
			set @existe = 0
		end
	end
/*
Si el status que se esta autorizando es ea, los autorizadores son los responsables
de las areas a las que pertenezcan los conceptos de gasto y los responsables
de area a la que pertenezca el usuario
Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
estar en la tabla temporal_2
Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
y no haber autorizado con anterioridad
*/
```

```
	if @idstatus = 'ea'
		begin
			if exists(
				select
				ar.numempleado
				from area_responsabilidad as aa
				inner join area_responsabilidad_autoriza as ar on aa.idarea = ar.idarea
				inner join #temporal_1 as te on ar.numempleado = te.autorizador
				where	aa.idempresa = ar.idempresa and
					ar.idempresa = te.idempresa and
					aa.idempresa = @idempresa and
					aa.idstatus = 'a'
				union
				select
				ua.autorizador
				from usuario as us
				inner join usuario_area_autoriza as ua on us.numempleado =
ua.numempleado
				inner join #temporal_1 as te on ua.autorizador = te.autorizador
				where	us.idempresa = ua.idempresa and
					ua.idempresa = te.idempresa and
					us.idempresa = @idempresa and
					us.idstatus = 'a'
			)
			begin
				set @existe = 1
			end
			else
			begin
				set @existe = 0
			end
		end
	/*
	Si el status que se esta autorizando es ed, el responsable es el director
	del usuario que haga la solicitud
	Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
	estar en la tabla temporal_2
	Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
	y no haber autorizado con anterioridad
	*/
	if @idstatus = 'ed'
		begin
			if exists(
				select
				ua.autorizador
				from usuario as us
				inner join usuario_direccion_autoriza as ua on us.numempleado =
ua.numempleado
				inner join #temporal_1 as te on ua.autorizador = te.autorizador
				where	us.idempresa = ua.idempresa and
					ua.idempresa = te.idempresa and
					us.idempresa = 1 and
					us.idstatus = 'a'
			)
			begin
				set @existe = 1
			end
			else
			begin
				set @existe = 0
			end
		end
```

```
	/*
	Si el status que se esta autorizando es ec, los autorizadores son los responsables
	de contabilidad de la empresa en cuestion
	Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
	estar en la tabla temporal_2
	Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
	y no haber autorizado con anterioridad
	*/
	if @idstatus = 'ec'
		begin
			if exists(
				select
				ca.numempleado
				from contabilidad_autoriza as ca
				inner join #temporal_1 as te on ca.numempleado = te.autorizador
				where	ca.idempresa = te.idempresa and
					ca.idempresa = @idempresa

			)
			begin
				set @existe = 1
			end
			else
			begin
				set @existe = 0
			end
		end

	select @existe as juega_rol

	/*
	Se eliminan ambas tablas temporales
	*/
	drop table #temporal_1
end


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_flujo_autoriza_mandar_autorizar    Script Date: 02/07/2008 04:36:31
p.m. ******/




/*
Inicia
Nombre:
sp_flujo_autoriza_mandar_autorizar
```

Funcionalidad:
Cuenta cuantos autorizadores quedan pendientes por autorizar la solicitud

Entradas:
Folio que se manda a autorizar
Numero del empleado autorizador
Empresa en la que se trabaje
Status que se autoriza
Parametro que indica si el flujo empieza desde el principo del flujo establecido para esa empresa o si empieza desde el ultimo nivel de autorizacion

Proceso:
Se manda a autorizar la solicitud con el status que se le mande como parametro. Es decir, el flujo comienza a partir del status que se le envie como parametro

Salida:
Solicitud registrada

Ejemplo:
exec [dbo].sp_flujo_autoriza_mandar_autorizar 1,123,1,'nu'
Termina
*/

```
CREATE procedure [dbo].sp_flujo_autoriza_mandar_autorizar
(
	@folio int,
	@numempleado int,
	@idempresa int,
	@idstatus char(2),
	@desde_final tinyint = 0
)
as
begin
	CREATE TABLE #tmp([id] int)
	if @desde_final = 1
	begin
		/*insert into #tmp*/
		exec sp_flujo_autoriza_registra_autorizacion
@folio,@numempleado,@idempresa,@idstatus,'Solicitud enviada a autorizar','a',1,1
	end
	else
	begin
		/*insert into #tmp*/
		 exec sp_flujo_autoriza_registra_autorizacion
@folio,@numempleado,@idempresa,@idstatus,'Solicitud enviada a autorizar','a',1
	end


	select @folio as folio, st.descripcion_1, so.idstatus
	from solicitud as so
	inner join status as st on so.idstatus = st.idstatus
	where folio = @folio

	drop table #tmp
end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
```

```
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_flujo_autoriza_obtiene_flujo_autorizacion    Script Date: 02/07/2008
04:36:25 p.m. ******/




/*
Inicia
Nombre:
sp_flujo_autoriza_obtiene_flujo_autorizacion



Funcionalidad:
Obtiene el flujo de autorizacion de una empresa

Entradas:
Empresa en la que se trabaje

Proceso:
Se obtiene el flujo de autorizacion de una empresa. Y se ordenan de una manera determinada por el
administrador tecnico

Salida:
Flujo de autorizacion

Ejemplo:
exec sp_flujo_autoriza_obtiene_flujo_autorizacion 13
Termina
*/

CREATE procedure [dbo].sp_flujo_autoriza_obtiene_flujo_autorizacion
(
	@idempresa int
)
as
begin
	select distinct
	st.idstatus,
	st.descripcion_1,
	case
		when st.idstatus = 'ep'	then 1
		when st.idstatus = 'ej'	then 2
		when st.idstatus = 'ea'	then 3
		when st.idstatus = 'ed'	then 4
		when st.idstatus = 'ec'	then 5
	end as idflujo

	from flujo_autoriza as fa
	inner join status as st on fa.idstatus = st.idstatus
```

```
		where	idempresa = @idempresa and
			st.idstatus in ('ep','ej','ea','ed','ec')
		order by idflujo
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_flujo_autoriza_obtiene_pendientes    Script Date: 02/07/2008
04:36:27 p.m. ******/



/*
Inicia
Nombre:
sp_flujo_autoriza_obtiene_pendientes



Funcionalidad:
Muestra al autorizador, las solicitudes que tiene pendientes por autorizar

Entradas:
Empresa en la que se trabaje
Folio que se autoriza
Numero del empleado autorizador
Status que se autoriza

Proceso:
Revisa las solicitudes y muestra las pendientes por autorizar al autorizador correspondiente

Salida:
Reporte de solicitudes pendientes por status y por autorizador

Ejemplo:
exec [dbo].sp_flujo_autoriza_obtiene_pendientes 1,124,'ep','compra'
Termina
*/

CREATE procedure [dbo].sp_flujo_autoriza_obtiene_pendientes
(
		@idempresa int,
		@numempleado int,
		@idstatus char(2),
		@tipo_operacion varchar(50)
)
as
```

```
begin
	declare
	@idtipo_operacion int

	set @idtipo_operacion = (select [dbo].fn_obtiene_tipo_operacion(@idempresa,@tipo_operacion))

	if exists(
		select
		so.folio
		from solicitud as so
		where	so.idempresa = @idempresa and
			so.idtipo_operacion = @idtipo_operacion and
			so.idstatus = @idstatus
	)begin

		/*
		Se crea tabla temporal que almacenara al autorizador y quienes le delegaron su autoridad
		*/
		create table #temporal_1 (autorizador int,idempresa int)
		insert into #temporal_1
		select @numempleado,@idempresa
		union
		select numempleado_delega,@idempresa
		from transferencia_autoridad
		where	numempleado_delegado = @numempleado
			and idempresa = @idempresa
			and fecha_ini <= [dbo].fn_obtiene_fecha_hoy(getdate())
			and fecha_fin >= [dbo].fn_obtiene_fecha_hoy(getdate())
			and idstatus = 'a'
		/*
		Se crea tabla temporal que almacenara quienes han autorizado a partir del ultimo rechazo
		Esto es para que siempre se tenga la informacion mas actual y no hacer comparaciones con
		autorizaciones previas al ultimo rechazo
		*/
		create table #temporal_2 (folio int, autorizador int)
		insert into #temporal_2
		select ha.folio, hd.numempleado_delega
		from historial_solicitud as ha
		inner join historial_delegacion_autoridad_solicitud as hd on ha.idhistorial = hd.idhistorial
		where	ha.idstatus = @idstatus and
			ha.idhistorial > [dbo].fn_flujo_autoriza_obtiene_ultimo_id_rechazo(ha.folio)

		/*
		Si el status que se esta autorizando es ep, los autorizadores son los responsables del
		presupuesto de los centros de costo entre los que se haya distribuido la solicitud
		Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
		estar en la tabla temporal_2
		Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
		y no haber autorizado con anterioridad
		*/
		if @idstatus = 'ep'
			begin

				select distinct
				an.folio as folio,
				rtrim(us.nombre) + ' ' + rtrim(us.paterno) as beneficiario,
				[dbo].fn_obtiene_total_por_folio(an.folio) as total,
				convert(char,an.fecha_solicitud,103) as fecha_solicitud
				from solicitud as an
				inner join detalle_solicitud as de on an.folio = de.folio
				inner join distribucion_solicitud as di on de.detalle = di.detalle
				inner join centro_costo as ce on rtrim(di.ccostos) = rtrim(ce.idccostos)
```

```
                                        inner join centro_costo_autoriza as ca on rtrim(ce.idccostos) =
rtrim(ca.idccostos)
                                        inner join #temporal_1 as te on ca.numempleado = te.autorizador
                                        inner join usuario as us on an.numempleado = us.numempleado
                                        where   an.idempresa = ce.idempresa and

                                                ce.idempresa = ca.idempresa and
                                                ca.idempresa = te.idempresa and
                                                te.idempresa = us.idempresa and
                                                an.idempresa = @idempresa and
                                                an.idstatus = @idstatus and
                                                an.idtipo_operacion = @idtipo_operacion and
                                                ca.numempleado not in (select autorizador from #temporal_2
where folio = an.folio)
                                        order by an.folio asc


                        end
                /*
                Si el status que se esta autorizando es ej, el autorizador es el jefe inmediato del
                solicitante
                Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
                estar en la tabla temporal_2
                Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
                y no haber autorizado con anterioridad
                */
                if @idstatus = 'ej'
                        begin
                                select distinct
                                an.folio as folio,
                                rtrim(us.nombre) + ' ' + rtrim(us.paterno) as beneficiario,
                                [dbo].fn_obtiene_total_por_folio(an.folio) as total,
                                convert(char,an.fecha_solicitud,103) as fecha_solicitud
                                from solicitud as an
                                inner join detalle_solicitud as de on an.folio = de.folio
                                inner join usuario as us on an.numempleado = us.numempleado
                                inner join #temporal_1 as te on us.num_jefe = te.autorizador
                                where   an.idempresa = us.idempresa and
                                        an.idempresa = @idempresa and
                                        an.idempresa = te.idempresa and
                                        an.idstatus = @idstatus and
                                        an.idtipo_operacion = @idtipo_operacion and
                                        us.num_jefe not in (select autorizador from #temporal_2 where
folio = an.folio)
                                order by an.folio asc
                        end
                /*
                Si el status que se esta autorizando es ea, los autorizadores son los dueños de las areas
                a las que pertenezcan los conceptos de gasto y las personas designadas para autorizar
                al solicitante cada vez que haga una solicitud
                Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
                estar en la tabla temporal_2
                Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
                y no haber autorizado con anterioridad
                */
                if @idstatus = 'ea'
                        begin
                                select distinct
                                an.folio as folio,
                                rtrim(ua.nombre) + ' ' + rtrim(ua.paterno) as beneficiario,
                                [dbo].fn_obtiene_total_por_folio(an.folio) as total,
                                convert(char,an.fecha_solicitud,103) as fecha_solicitud
```

```
                                        from solicitud as an
                                        inner join detalle_solicitud as de on an.folio = de.folio
                                        inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
                                        inner join concepto_gasto_area as ca on cg.idconcepto = ca.idconcepto
                                        inner join area_responsabilidad as ar on ca.idarea = ar.idarea
                                        inner join area_responsabilidad_autoriza as aa on ar.idarea = aa.idarea
                                        inner join usuario as us on aa.numempleado = us.numempleado
                                        inner join #temporal_1 as te on aa.numempleado = te.autorizador
                                        inner join usuario as ua on an.numempleado = ua.numempleado
                                        where   an.idempresa = cg.idempresa and
                                                cg.idempresa = ca.idempresa and
                                                ca.idempresa = ar.idempresa and
                                                ar.idempresa = aa.idempresa and
                                                aa.idempresa = us.idempresa and
                                                us.idempresa = te.idempresa and
                                                te.idempresa = ua.idempresa and
                                                an.idempresa = @idempresa and
                                                an.idstatus = @idstatus and
                                                cg.idstatus = 'a' and
                                                ar.idstatus = 'a' and
                                                us.idstatus = 'a' and
                                                an.idtipo_operacion = @idtipo_operacion and
                                                aa.numempleado not in (select autorizador from #temporal_2
where folio = an.folio)
                                        union
                                        select distinct
                                        an.folio as folio,
                                        rtrim(u1.nombre) + ' ' + rtrim(u1.paterno) as beneficiario,
                                        [dbo].fn_obtiene_total_por_folio(an.folio) as total,
                                        convert(char,an.fecha_solicitud,103) as fecha_solicitud
                        from solicitud as an
                                        inner join detalle_solicitud as de on an.folio = de.folio
                                        inner join usuario as u1 on an.numempleado = u1.numempleado
                                        inner join usuario_area_autoriza as ua on u1.numempleado =
ua.numempleado
                                        inner join usuario as u2 on ua.autorizador = u2.numempleado
                                        inner join #temporal_1 as te on ua.autorizador = te.autorizador
                                        where   an.idempresa = u1.idempresa and
                                                u1.idempresa = ua.idempresa and
                                                ua.idempresa = u2.idempresa and
                                                u2.idempresa = te.idempresa and
                                                an.idempresa = @idempresa and
                                                an.idstatus = @idstatus and
                                                u2.idstatus = 'a' and
                                                an.idtipo_operacion = @idtipo_operacion and
                                                ua.autorizador not in (select autorizador from #temporal_2 where
folio = an.folio)
                                        order by an.folio asc
                                end
                        /*
                        Si el status que se esta autorizando es ed, los autorizadores son los directores de la
empresa
                        Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
                        estar en la tabla temporal_2
                        Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
                        y no haber autorizado con anterioridad
                        */
                        if @idstatus = 'ed'
                                begin
                                        select distinct
                                        an.folio as folio,
                                        rtrim(u1.nombre) + ' ' + rtrim(u1.paterno) as beneficiario,
```

```
                                        [dbo].fn_obtiene_total_por_folio(an.folio) as total,
                                        convert(char,an.fecha_solicitud,103) as fecha_solicitud
                                        from solicitud as an
                                        inner join detalle_solicitud as de on an.folio = de.folio
                                        inner join usuario as u1 on an.numempleado = u1.numempleado
                                        inner join usuario_direccion_autoriza as ua on u1.numempleado =
ua.numempleado
                                        inner join usuario as u2 on ua.autorizador = u2.numempleado
                                        inner join #temporal_1 as te on ua.autorizador = te.autorizador
                                        where   an.idempresa = u1.idempresa and
                                                u1.idempresa = ua.idempresa and
                                                ua.idempresa = u2.idempresa and
                                                u2.idempresa = te.idempresa and
                                                an.idempresa = @idempresa and
                                                an.idstatus = @idstatus and
                                                u2.idstatus = 'a' and
                                                an.idtipo_operacion = @idtipo_operacion and
                                                ua.autorizador not in (select autorizador from #temporal_2 where
folio = an.folio)
                                        order by an.folio asc
                                end

                        /*
                        Si el status que se esta autorizando es ej, los autorizadores son los responsables
                        de contabilidad de la empresa
                        Estos autorizadores tienen que estar en la tabla temporal_1 y no tienen que
                        estar en la tabla temporal_2
                        Es decir, tienen que estar dentro de quienes le han delegado la autoridad al autorizador
                        y no haber autorizado con anterioridad
                        */
                        if @idstatus = 'ec'
                                begin
                                        select distinct
                                        an.folio as folio,
                                        rtrim(us.nombre) + ' ' + rtrim(us.paterno) as beneficiario,
                                        [dbo].fn_obtiene_total_por_folio(an.folio) as total,
                                        convert(char,an.fecha_solicitud,103) as fecha_solicitud
                                        from solicitud as an
                                        inner join detalle_solicitud as de on an.folio = de.folio
                                        inner join contabilidad_autoriza as ca on an.idempresa = ca.idempresa
                                        inner join #temporal_1 as te on ca.numempleado = te.autorizador
                                        inner join usuario as us on an.numempleado = us.numempleado
                                        where   an.idstatus = @idstatus and
                                                ca.idempresa = te.idempresa and
                                                te.idempresa = us.idempresa and
                                                an.idempresa = @idempresa and
                                                an.idtipo_operacion = @idtipo_operacion and
                                                ca.numempleado not in (select autorizador from #temporal_2
where folio = an.folio)
                                        order by an.folio asc
                                end
                        /*
                        Se eliminan ambas tablas temporales
                        */
                        drop table #temporal_1
                        drop table #temporal_2
        end
        else

        begin
                        select
                        null as folio,
```

```
                    null as beneficiario,
                    null as total,
                    null as fecha_solicitud
                    where 1 = 2
        end
end


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_flujo_autoriza_registra_autorizacion    Script Date: 02/07/2008
04:36:31 p.m. ******/




/*
Inicia
Nombre:
sp_flujo_autoriza_registra_autorizacion



Funcionalidad:
Procedimiento general para obtener el flujo de autorizacion de la solicitud

Entradas:
Folio de la solicitud
Numero del empleado que autoriza
Status que se autoriza
Comentarios de la autorizacion o el rechazo
Variable de autorizacion o rechazo
        (a = autorizacion)
        (r = rechazo)
Variable commit o roll-back
        (1 = commit)
        (0 = roll-back)

Proceso:
Proceso de autorizacion o rechazo de la solicitud

Salida:
Solicitud autorizada o rechazada

Ejemplo:
exec sp_flujo_autoriza_registra_autorizacion 412,100000,1,'ej','Autorizacion','a',1
Termina
*/

CREATE procedure [dbo].sp_flujo_autoriza_registra_autorizacion
```

```
(
	@folio int,				--Numero de folio de la solicitud
	@numempleado int,			--Numero de empleado que registra la autorizacion
	@idempresa_s int,			--Empresa a la que se supone que pertenece la solicitud
	@idstatus char(2),			--Status que se esta autorizando
	@observacion varchar(250),		--Comentarios acerca de la autorizacion o rechazo
	@autorizacion_o_rechazo char(1), --Bandera de autorizacion o rechazo (a,r)
	@commit tinyint = 1,			--Variable commit(0,1)
	@desde_final tinyint = 0		--Si el flujo comienza desde el final (1) o desde el comienzo(0)
)
as
begin
	declare
	@c_idstatus char(2),			--Cursor que va a almacenar el status que se busque
	@idempresa int,				--Empresa en la que se esta trabajando
	@leyenda_existe char(2),  --Leyenda si el empledo existe o esta autorizado para autorizar un status
(si, no)
	@opcion_flujo_masivo tinyint,		--Opcion de flujo de autorizacion (0 = Los niveles se autorizan
uno por uno, 1 = Cuando una persona tenga diferentes roles, una autorizacion vale por todas)
	@empleado_existe int,			--Si el empleado autorizador existe o no, como autorizador de
ese status (0 = no existe, 1 = si existe)
	@faltan int,				--Variable donde se guarda cuantos faltan por autorizar la
solicitud
	@idhistorial_supuesto int,  --Variable donde se guarda el idhistorial de los registros ya almacenados
para esa solicitud
						--Esta variable se utiliza cuando un autorizador quiere ver como
se guardarian
						--los registros antes de que sean autorizados o rechazados
	@idtipo_operacion int,
	@disponibilidad tinyint,		--disponibilidad de ppto para esta solicitud

	/*
	Estas declaraciones se hacen en caso de que se rechace la solicitud
	para enviarle un correo al solicitante
	*/
	@mail_origen varchar(50),
	@mail_destino varchar(50),
	@mensaje varchar(250),
	@body varchar(4000)

	set @idempresa = (select idempresa from solicitud where folio = @folio)
	set @idtipo_operacion = (select idtipo_operacion from solicitud where folio = @folio)
	/*
	Comienzo de la transaccion
	*/
	begin transaction trans_registra_autorizacion

	/*
	Si existe el nivel de PPTO en el flujo de autorizacion
	para esa empresa y tipo de operacion
	*/
	if exists(
		select
		*
		from flujo_autoriza
		where	idempresa = @idempresa and
			idtipo_operacion = @idtipo_operacion and
			idstatus = 'ep'
	)
	and
	/*
	Y no existe un autorizacion de PPTO a partir del ultimo rechazo
```

```
	de la solicitud
	*/
	not exists(
			select top  1 '1'
			from historial_solicitud as hs
			where	hs.folio = @folio and
				hs.idhistorial > [dbo].fn_flujo_autoriza_obtiene_ultimo_id_rechazo(@folio)
and
				hs.idstatus = 'ep'
		)
	begin
		/*
		Se obtiene la disponibilidad de PPTO para ese folio
		*/
		exec [dbo].sp_revisa_disponibilidad_ppto @folio,@disponibilidad output
	end
	else
	begin
		/*
		Se asume que si hay disponibilidad de PPTO porque no hay flujo de autorizacion
		de PPTO
		*/
		set @disponibilidad = 1
	end

	/*
	Se verifica que la empresa a la que pertenece el folio sea igual a la que dice el autorizador (se manda
como parametro)
	*/
	if @idempresa_s = @idempresa
	begin

		set @idhistorial_supuesto =  (select isnull(max(idhistorial),0) + 1 from historial_solicitud
where folio = @folio)
		/*
		Se evalua si lo que se quiere hacer es una autorizacion o un rechazo
		Si se trata de una autorizacion:
		*/
		if @autorizacion_o_rechazo = 'A'
			begin
			/*
			Se obtiene la empresa a la que pertenece ese folio

			Se revisa si ese autorizador puede autorizar ese status
			El resultado (0 o 1) se guarda en la variable
			@empleado_existe
			*/
			exec sp_flujo_autoriza_revisa_autorizadores_status
@idempresa,@folio,@numempleado,@idstatus,@disponibilidad,@empleado_existe output
			/*
			Si el autorizador existe entonces se registra la autorizacion con ese status
			*/
			if @empleado_existe = 1
			begin
				update solicitud set idstatus = @idstatus where folio = @folio
				exec sp_registra_historial
@folio,@numempleado,@idstatus,@observacion
				/*
				Se busca la opcion de autorizacion del flujo
				1 = Cuando un autorizador pertenezca a varios roles, una autorizacion le
da el vo.bo a todos los roles
```

```
                                        0 = Los niveles se van autorizando uno por uno. Si una persona tiene n
roles, tiene que autorizar n veces
                                        */
                                        set @opcion_flujo_masivo = (select flujo_masivo from
flujo_autoriza_opcion where idempresa = @idempresa)

                                        if @opcion_flujo_masivo = 1
                                        begin
                                                /*
                                                Se buscan todos los status por los que pasara la solicitud,
                                                excepto el que ya ha sido autorizado, esos status se almacenan
                                                en @c_idstatus
                                                */

                                                if @desde_final = 1
                                                begin
                                                        declare cursor_1 cursor for

                                                                select
        [dbo].fn_obtiene_ultimo_status_anterior(@idempresa,@idtipo_operacion,'at')
                                                end
                                                else
                                                begin
                                                        declare cursor_1 cursor for

                                                                select  idstatus
                                                                from    flujo_autoriza
                                                                where   idempresa = @idempresa
                                                                        and idflujo > (
                                                                                select idflujo
                                                                                from flujo_autoriza
                                                                                where   idempresa =
@idempresa and
                                                                                        idstatus =
@idstatus and

        idtipo_operacion = @idtipo_operacion
                                                                                ) and
                                                                        idtipo_operacion = @idtipo_operacion
                                                                order by idflujo asc
                                                end

                                                open cursor_1

                                                fetch next from cursor_1
                                                        into
                                                                @c_idstatus

                                                while @@fetch_status = 0
                                                        begin
                                                                /*
                                                                Se inicializan las variables @empleado_existe y
@faltan en cero

                                                                Esto se hace para limpiar el resultado de la
operacion anterior en el ciclo
                                                                */
                                                                set @empleado_existe = 0
                                                                set @faltan = 0
                                                                /*
                                                                Con el siguiente procedimiento se evalua que el
empleado y/o quienes le
```

```
                                                delegaron la autoridad, esten autorizados para
autorizar para el status
                                                en curso
                                                El resultado (1 = existe, 0 = no existe) se
almacena en la variable @empleado_existe
                                                */
                                                exec
sp_flujo_autoriza_revisa_autorizadores_status
@idempresa,@folio,@numempleado,@c_idstatus,@disponibilidad,@empleado_existe output
                                                if @empleado_existe = 1
                                                begin
                                                        if @disponibilidad = 1 and @c_idstatus
= 'ep'
                                                        begin
                                                                set @observacion =
'presupuesto disponible'
                                                        end
                                                        else
                                                        begin
                                                                set @observacion =
'autorizacion anticipada'                                               end


                                                        /*
                                                        Se cuentan cuantos autorizadores
faltan por autorizar esa solicitud para el status en curso
                                                        El resultado se almacena en la variable
@faltan
                                                        */
                                                        exec
sp_flujo_autoriza_cuenta_autorizadores
@numempleado,@idempresa,@c_idstatus,@folio,'cf',@disponibilidad,@faltan output
                                                        /*
                                                        Si si faltan personas por autorizar la
solicitud, entonces se almacena la autorizacion
                                                        del autorizador y las autorizaciones de
las personas que le han delegado su autoridad
                                                        */
                                                        if @faltan > 0
                                                        begin
                                                                exec sp_registra_historial
@folio,@numempleado,@c_idstatus,@observacion
                                                                if @c_idstatus = 'ep'
                                                                begin
                                                                        exec
sp_flujo_autoriza_cuenta_autorizadores
@numempleado,@idempresa,@c_idstatus,@folio,'cf',@disponibilidad,@faltan output
                                                                        if @faltan <= 0
                                                                        /*
                                                                        Si el estado es
presupuesto y ya no falta nadie por autorizar se inserta se registra que este ppto ya se comprometio
                                                                        La razon por la que el
estado es ep y ya no falta nadie por autorizar es porque hay disponibilidad de ppto
                                                                        */
                                                                        begin
                                                                                exec
[dbo].sp_ppto_inserta_modifica @folio
                                                                        end
                                                                end

                                                        end
```

```
                                                set @leyenda_existe = 'si'
                                        end
                                        else
                                        begin
                                                set @leyenda_existe = 'no'
                                        end

                                        fetch next from cursor_1
                                        into
                                                @c_idstatus
                                end

                        close           cursor_1
                        deallocate      cursor_1

                end
                /*
                Se determina cual es el siguiente status de la solicitud
                Ya sea que avance al siguiente o permanezca en el mismo
                */
                if @desde_final = 1
                begin
                        set @idstatus = (select
[dbo].fn_obtiene_ultimo_status_anterior(@idempresa,@idtipo_operacion,'at'))
                        update solicitud set idstatus = @idstatus where folio = @folio
                end

                exec sp_flujo_autoriza
@numempleado,@idempresa,@idstatus,@folio,@idtipo_operacion,@disponibilidad
        end
        /*
        Si se trata de un rechazo:
        */
        else
        begin
                /*
                Se obtiene el status de rechazo que le corresponde al status enviado
                Se actualiza la solicitud
                Se registra el rechazo en el historial
                Se le envia al solicitante un mail avisandole que su solicitud ha sido rechazada
                */
                set @c_idstatus = (select idstatus_rechazo from status_rechazo where idstatus =
@idstatus)
                update solicitud set idstatus = @c_idstatus where folio = @folio
                exec sp_registra_historial @folio,@numempleado,@c_idstatus,@observacion


                set @mail_origen = (
                        select
                        email
                        from usuario
                        where   idempresa = @idempresa and
                                numempleado = @numempleado
                )

                set @mail_destino = (
                        select
                        email
                        from usuario as us
                        inner join solicitud as so on us.numempleado = so.numempleado
                        where   us.idempresa = so.idempresa and
```

```
                                        so.idempresa = @idempresa and
                                        so.folio = @folio
                        )

                        set @mensaje = (
                                select
                                mensaje
                                from mensaje_status
                                where   idempresa = @idempresa and
                                        idstatus = @c_idstatus
                        )
                        if @mensaje is null or @mensaje = ''
                        begin
                                set @mensaje = 'No definido'
                        end

                        set @body =     'Rechazo por : ' +
[dbo].fn_obtiene_nombre(@idempresa,@numempleado) + '<br>' +
                                        'Folio : ' + rtrim(cast(@folio as char)) + '<br>' +
                                        'Mensaje : ' + rtrim(@mensaje) + '<br>' +
                                        'Motivo de rechazo : ' + rtrim(@observacion)

                        exec [dbo].sp_send_cdosysmail @mail_origen,@mail_origen,'Solicitud
Rechazada',@body
                end
                /*
                Si la variable commit = 0, se presenta al autorizador como quedarian los registros
                si efectua la autorizacion o el rechazo
                Se presenta la informacion nueva, es decir, la que se generaria
                Para esto se utiliza la variable @idhistorial_supuesto, para no presentar
                registros existentes
                */
                if @commit = 0
                begin
                        select
                                st.descripcion_2 as autorizacion,
                        case
                                when @numempleado = hd.numempleado_delega then 'PROPIA'
                                when @numempleado <> hd.numempleado_delega then 'DELEGADA'
                        end as tipo_autorizacion,
                                rtrim(us.nombre) + ' ' + rtrim(us.paterno) +' ' + rtrim(us.materno) as
responsable
                        from solicitud as an
                        inner join historial_solicitud as ha on an.folio = ha.folio
                        inner join historial_delegacion_autoridad_solicitud as hd on ha.idhistorial =
hd.idhistorial
                        inner join usuario as us on hd.numempleado_delega = us.numempleado
                        inner join status as st on ha.idstatus = st.idstatus
                        where   an.idempresa = us.idempresa and
                                an.folio = @folio and
                                ha.idhistorial >= @idhistorial_supuesto
                        order by ha.idhistorial asc
                end
        end

        if @commit = 1
        begin
                if @autorizacion_o_rechazo = 'a'
                begin
                        set @c_idstatus = (select idstatus from solicitud where folio = @folio)
                        exec  [dbo].sp_reporte_consulta_autorizadores @folio,@c_idstatus,'i',1,0
                end
```

```
                commit transaction trans_registra_autorizacion
                print 'Transaction commited'
        end
        if @commit <> 1
        begin
                rollback transaction trans_registra_autorizacion
                print 'Transaction rolled-back'
        end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_flujo_autoriza_revisa_autorizadores_status    Script Date: 02/07/2008
04:36:22 p.m. ******/




/*
Inicia
Nombre:
sp_flujo_autoriza_revisa_autorizadores_status



Funcionalidad:
Revisa que el autorizador o quienes le hayan delegado su autoridad existan como autorizador(es)


Entradas:
Numero de empresa en la que se esta trabajando
Folio de la solicitud
Numero de empleado autorizador
Status que se autoriza
Bandera de existencia de empleado (parametro de salida)

Proceso:
Revisa si el autorizador o quienes le delegaron su autoridad estan en la posibilidad de autorizar ese status

Salida:
Bandera de existencia de autorizador
        (0 = no puede autorizar)
        (1 = si puede autorizar)

Ejemplo:
```

```
declare
@empleado tinyint
set @empleado = 0
exec sp_flujo_autoriza_revisa_autorizadores_status 3,1,1,'ej', @empleado output
print 'empleado numero ' + cast(@empleado as char)
Termina
*/

CREATE procedure [dbo].sp_flujo_autoriza_revisa_autorizadores_status
(
	@idempresa int,		--Empresa en la que se esta trabajando
	@folio int,		--Folio de la solicitud
	@numempleado int,	--Numero de empleado que autoriza la solicitud
	@idstatus char(2),	--Status que se autoriza
	@disponibilidad tinyint,
	@existe int output	--Si ese autorizador (o los que le transifieron la autoridad) existe(n) para
ese folio en ese status
)
as
begin

	/*
	Se crean las tablas temporales
	*/
	create table #temporal_1 (autorizador int, idempresa int)
	create table #temporal_2 (existe int)

	/*
	En #temporal_1 se almacenan los usuarios que pueden autorizar, es decir
	el empleado autorizador y las personas que le han delegado su autoridad
	*/
	insert into #temporal_1
	select @numempleado,@idempresa
	union
	select numempleado_delega,@idempresa
	from transferencia_autoridad
	where	numempleado_delegado = @numempleado
		and idempresa = @idempresa
		and fecha_ini <= [dbo].fn_obtiene_fecha_hoy(getdate())
		and fecha_fin >= [dbo].fn_obtiene_fecha_hoy(getdate())
		and idstatus = 'a'

	/*
	Todos tienen derecho a mandar a autorizar una solicitud
	*/
	if @idstatus = 'nu'
	begin
		insert into #temporal_2
		select @numempleado as existe
	end
	/*
	Si se autoriza el status ep, los autorizadores son los responsables de los centros
	de costo entre los que se haya distribuido la solicitud
	Esos autorizadores tienen que existir en los registros de #temporal_1
	Eso delimita si un usuario tien el poder de autorizar este status
	*/
	if @idstatus = 'ep'
		if @disponibilidad = 1
		begin
			insert into #temporal_2
			select distinct @numempleado as existe
		end
```

```
		else
		begin
			insert into #temporal_2
			select distinct isnull(ca.numempleado,0) as existe
			from distribucion_solicitud as di
			inner join centro_costo as ce on rtrim(di.ccostos) = rtrim(ce.idccostos)
			inner join centro_costo_autoriza as ca on rtrim(ce.idccostos) = rtrim(ca.idccostos)
			where	di.folio = @folio and
				di.idempresa = ce.idempresa and
				ce.idempresa = ca.idempresa and
				di.idempresa = @idempresa and
				ca.numempleado in (select autorizador from #temporal_1 where idempresa
= @idempresa)
		end
	/*
	Si se autoriza el status ej, el autorizador es el jefe inmediato
	Esos autorizadores tienen que existir en los registros de #temporal_1
	Eso delimita si un usuario tien el poder de autorizar este status
	*/
	if @idstatus = 'ej'
		begin
			insert into #temporal_2
			select distinct isnull(us.num_jefe,0) as existe
			from solicitud as an
			inner join usuario as us on an.numempleado = us.numempleado
			where	an.idempresa = us.idempresa and
				an.folio = @folio and
				an.idempresa = @idempresa and
				us.num_jefe in (select autorizador from #temporal_1 where idempresa =
@idempresa)
		end

	/*
	Si se autoriza el status ea, los autorizadores tienen que ser responsables
	del area a la que pertenezca el concepto de gasto o usuarios designados como
	responsables cuando un usuario haga cualquier solicitud
	Esos autorizadores tienen que existir en los registros de #temporal_1
	Eso delimita si un usuario tien el poder de autorizar este status
	*/
	if @idstatus = 'ea'
		begin
			insert into #temporal_2
			select distinct isnull(aa.numempleado,0) as existe
			from detalle_solicitud as de
			inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
			inner join concepto_gasto_area as ca on cg.idconcepto = ca.idconcepto
			inner join area_responsabilidad as ar on ca.idarea = ar.idarea
			inner join area_responsabilidad_autoriza as aa on ar.idarea = aa.idarea
			inner join usuario as us on aa.numempleado = us.numempleado
			where	de.idempresa = cg.idempresa and
				cg.idempresa = ca.idempresa and
				ca.idempresa = ar.idempresa and
				ar.idempresa = aa.idempresa and
				aa.idempresa = us.idempresa and
				cg.idstatus = 'a' and
				ar.idstatus = 'a' and
				us.idstatus = 'a' and
				de.folio = @folio and
				de.idempresa = @idempresa and
				aa.numempleado in (select autorizador from #temporal_1 where idempresa
= @idempresa)
			union
```

```
                        select distinct isnull(ua.autorizador,0) as existe
                        from solicitud as an
                        inner join usuario as u1 on an.numempleado = u1.numempleado
                        inner join usuario_area_autoriza as ua on u1.numempleado = ua.numempleado
                        inner join usuario as u2 on ua.autorizador = u2.numempleado
                        where   an.idempresa = u1.idempresa and
                                u1.idempresa = ua.idempresa and
                                ua.idempresa = u2.idempresa and
                                an.idempresa = @idempresa and
                                u2.idstatus = 'a' and
                                an.folio = @folio and
                                ua.autorizador in (select autorizador from #temporal_1 where idempresa =
@idempresa)
                                order by existe desc

                end
        /*
        Si se autoriza el status ed, los autorizadores son los directores
        Esos autorizadores tienen que existir en los registros de #temporal_1
        Eso delimita si un usuario tien el poder de autorizar este status
        */
        if @idstatus = 'ed'
                begin
                        insert into #temporal_2
                        select distinct isnull(ua.autorizador,0) as existe
                        from solicitud as an
                        inner join usuario as u1 on an.numempleado = u1.numempleado
                        inner join usuario_direccion_autoriza as ua on u1.numempleado = ua.numempleado
                        inner join usuario as u2 on ua.autorizador = u2.numempleado
                        where   an.idempresa = u1.idempresa and
                                u1.idempresa = ua.idempresa and
                                ua.idempresa = u2.idempresa and
                                an.idempresa = @idempresa and
                                u2.idstatus = 'a' and
                                an.folio = @folio and
                                ua.autorizador in (select autorizador from #temporal_1 where idempresa =
@idempresa)
                                order by existe desc
                end
        /*
        Si se autoriza el status ec, los autorizadores tienen que ser los responsables
        de contabilidad dentro de esa empresa
        Esos autorizadores tienen que existir en los registros de #temporal_1
        Eso delimita si un usuario tien el poder de autorizar este status
        */
        if @idstatus = 'ec'
                begin
                        insert into #temporal_2
                        select distinct isnull(ca.numempleado,0) as existe
                        from solicitud as an
                        inner join contabilidad_autoriza as ca on an.idempresa = ca.idempresa

                        where   an.folio = @folio and                                   an.idempresa =
@idempresa and
                                ca.numempleado in (select autorizador from #temporal_1 where idempresa
= @idempresa)
                end

        /*
        Una vez que esos autorizadores con esa autoridad estan registrados en #temporal_2
        Se hace una breve evaluacion para saber si esa tabla no esta vacia
        Si esta vacia quiere decir que ni el autorizador ni las personas que le delegaron
```

```
	su autoridad tienen la facultad para autorizar ese status
	Si la tabla no esta vacia, quiere decir que el autorizador si puede autorizar, ya
	sea con su nombre o bajo el nombre de quienes le delegaron su autoridad
	Este valor se guarda en la variable @existe
	*/
	set @existe = (
		select top 1 existe
		from #temporal_2
		where existe is not null
	)
	/*
	Se eliminan las tablas temporales
	*/
	drop table #temporal_1
	drop table #temporal_2

	/*
	Se regresa el valor 1 o 0 dependiendo de si pueden autorizar o no
	*/
	if @existe is null or @existe = 0
	begin
		set @existe = 0
	end
	else
	begin
		set @existe = 1
	end

	return
end


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_flujo_autorizacion_inserta_modifica    Script Date: 02/07/2008
04:36:25 p.m. ******/




/*
Inicia
Nombre:
sp_flujo_autorizacion_inserta_modifica



Funcionalidad:
```

Este procedimiento inserta el orden en que sera llevado a cabo el flujo de autorizacion en la empresa

Entradas:
Numero de empresa en la que se esta trabajando
Tipo de operacion para el que aplicara dicho flujo de autorizacion
Status a insertar
Accion a seguir
	i = insertar un nuevo status en el flujo
	o = obtener los status que no han sido seleccionados e insertados en el flujo
	n = despliega los status en el orden en que aparecen en el flujo

Proceso:
Una u otra accion dependiendo de la variable @proviene

Salida:
Status insertado o
Status disponible o
Orden de status en el flujo

Ejemplo:
exec sp_flujo_autorizacion_inserta_modifica 1, 1,'ep',i
Termina
*/

```
CREATE procedure [dbo].sp_flujo_autorizacion_inserta_modifica
(
	@idempresa int,
	@idtipo_operacion int,
	@idstatus char(2),
	@proviene char(1)
)
as
begin
	declare
	@ultimo_status char(2),
	@idflujo int,
	@idstatus_sig char(2)

	begin transaction flujo_autorizacion

	/*
	Esto es para insertar un nuevo status en el flujo
	*/
	if @proviene = 'i'
	begin
		set @ultimo_status = (
			select idstatus
			from flujo_autoriza
			where	idstatus_sig = 'at' and
				idempresa = @idempresa and
				idtipo_operacion = @idtipo_operacion
		)

		if @ultimo_status is not null
		begin
			update flujo_autoriza
			set idstatus_sig = @idstatus
			where	idempresa = @idempresa and
				idtipo_operacion = @idtipo_operacion and
```

```
                                        idstatus = @ultimo_status

                                set @idflujo = (select isnull(max(idflujo),0) + 1 from flujo_autoriza)

                                insert into flujo_autoriza
                                (
                                idflujo,
                                idempresa,
                                idtipo_operacion,
                                idstatus,
                                idstatus_sig
                                )
                                values
                                (
                                @idflujo,
                                @idempresa,
                                @idtipo_operacion,
                                @idstatus,
                                'at'
                                )
                        end
                        else
                        begin
                                set @idflujo = (select isnull(max(idflujo),0) + 1 from flujo_autoriza)

                                insert into flujo_autoriza
                                (
                                idflujo,
                                idempresa,
                                idtipo_operacion,
                                idstatus,
                                idstatus_sig
                                )
                                values
                                (
                                @idflujo,
                                @idempresa,
                                @idtipo_operacion,
                                'nu',
                                'at'
                                )
                        end
                end

                /*
                Esto es para dibujar el select y ordenar los status en el orden en que se quiera que
                funcione el flujo de autorizacion
                */
                if @proviene = 'o'
                begin
                        select
                        st.idstatus,
                        st.descripcion_1
                        from status as st
                        left outer join flujo_autoriza as fa on st.idstatus = fa.idstatus and fa.idempresa = @idempresa
and fa.idtipo_operacion = @idtipo_operacion
                        where   fa.idstatus is null and
                                substring(st.idstatus,1,1) = 'e'
                end

                /*
                Esto obtiene los nombres y el orden de los status que participan en el flujo
```

```
	*/
	if @proviene = 'n'
	begin
		select
		fa.idflujo,
		st.idstatus,
		st.descripcion_1,
		fa.idstatus_sig,

	[dbo].fn_obtiene_numero_solicitudes_por_status(@idempresa,@idtipo_operacion,lower(st.idstatus))
		from status as st
		inner join flujo_autoriza as fa on st.idstatus = fa.idstatus
		where	fa.idempresa = @idempresa and
			fa.idtipo_operacion = @idtipo_operacion
		order by fa.idflujo asc
	end

	/*
	Si se quiere eliminar un flujo
	*/
	if @proviene = 'd'
	begin

		set @idstatus_sig = (
			select
			fa.idstatus_sig
			from flujo_autoriza as fa
			where	fa.idempresa = @idempresa and
				fa.idtipo_operacion = @idtipo_operacion and
				fa.idstatus = @idstatus
		)

		update flujo_autoriza
		set
		idstatus_sig = @idstatus_sig
		where	idempresa = @idempresa and
			idtipo_operacion = @idtipo_operacion and
			idstatus_sig = @idstatus

		delete from flujo_autoriza
		where	idempresa = @idempresa and
			idtipo_operacion = @idtipo_operacion and
			idstatus = @idstatus
	end

	commit transaction flujo_autorizacion
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Stored Procedure dbo.sp_get_new_load_key    Script Date: 02/07/2008 04:36:23 p.m. ******/



/*
declare
@last_key_assigned as int
set @last_key_assigned = 0
exec sp_get_new_load_key 'banco',@last_key_assigned

print '@last_key_assigned = ' + cast(@last_key_assigned as char)

*/

CREATE PROC sp_get_new_load_key
                @load_description varchar(100) = 'Carga normal',
                @last_key_assigned int OUTPUT
AS

DECLARE @object_name varchar(50),
        @locked_status_flag char(2)

SET     @object_name='carga'

/* Obtener consecutivo para siguiente carga */

EXECUTE sp_rtrn_sequential
                @object_name,
                @last_key_assigned OUTPUT,
                @locked_status_flag OUTPUT,
                's'

/* Incrementar consecutivo e insertar nueva carga */
SET  @last_key_assigned=@last_key_assigned+1

INSERT INTO carga
                        (
                        llave_carga,
                        date_time_started,
                        load_description
                        )
                VALUES          (
                        @last_key_assigned,
                        GETDATE(),
                        @load_description
                        )

/* Actualizar tabla carga con nuevo consecutivo */
EXECUTE sp_update_object_sequential_key @object_name,@last_key_assigned
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_iva_catalogo    Script Date: 02/07/2008 04:36:24 p.m. ******/




/*
Inicia
Nombre:
sp_iva_catalogo



Funcionalidad:
Obtiene el catalogo de ivas

Entradas:
Accion a seguir
	a = presenta datos relevantes al administrador
	u = presenta datos relevantes a los usuarios

Proceso:
Obtiene el catalogo de ivas

Salida:
Catalogo de ivas

Ejemplo:
exec [dbo].sp_iva_catalogo 'u'
Termina
*/


CREATE procedure [dbo].sp_iva_catalogo
(
	@proviene char(1)
)
as
begin
	if @proviene = 'u'
	begin
		select
		rtrim(iv.identificador),
		iv.porcentaje,
		iv.descripcion
```

```
		from iva as iv
		where	iv.idstatus = 'a'
	end

	if @proviene = 'a'
	begin
		select
		rtrim(iv.identificador),
		iv.porcentaje,
		iv.descripcion,
		st.descripcion_1
		from iva as iv
		left outer join status as st on iv.idstatus = st.idstatus
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_kilometro_catalogo    Script Date: 02/07/2008 04:36:25 p.m. ******/




/*
Inicia
Nombre:
sp_kilometro_catalogo



Funcionalidad:
Obtiene el catalogo de origenes y destinos por empresa

Entradas:
Empresa en la que se esta trabajando
Identificador del kilometro (origen, destino)

Proceso:
Obtiene el catalogo de origenes y distancias

Salida:
Catalogo de origenes y distancias

Ejemplo:
exec [dbo].sp_kilometro_catalogo 1,1
```

```
Termina
*/

CREATE procedure [dbo].sp_kilometro_catalogo
(
	@idempresa int,
	@idkm int
)
as
begin
	if @idkm = 0
	begin
		select
		kl.idkm as idkm,
		kl.origen as origen,
		kl.destino as destino,
		kl.distancia as distancia
		from kilometro as kl
		where	kl.idempresa = @idempresa and
			kl.idstatus = 'a'
	end
	else
	begin
		select
		kl.idkm as idkm,
		kl.origen as origen,
		kl.destino as destino,
		kl.distancia as distancia
		from kilometro as kl
		where	kl.idempresa = @idempresa and
			kl.idstatus = 'a' and
			kl.idkm = @idkm
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_kilometro_inserta_modifica    Script Date: 02/07/2008 04:36:25 p.m.
******/




/*
Inicia
```

Nombre:
sp_kilometro_inserta_modifica

Funcionalidad:
Inserta, modifica o elimina un registro de destinos y distancias

Entradas:
Identificador del kilometro (origen, destino) en caso de que fuera una modificacion
Empresa en la que se esta trabajando
Origen
Destino
Distancia en KM
Status del registro
Accion a seguir
	i = insertar
	d = eliminar

Proceso:
Inserta o actualiza la tabla de origenes y destinos

Salida:
Registro insertado o eliminado

Ejemplo:
exec [dbo].sp_kilometro_inserta_modifica 1,1,'mexico','villahermosa','900','a','u'
Termina
*/

```
CREATE procedure [dbo].sp_kilometro_inserta_modifica
(
	@id int,
	@idempresa int,
	@origen varchar(50),
	@destino varchar(50),
	@distancia int,
	@idstatus char(2),
	@proviene char(1)
)
as
begin
	declare
	@idkm int

	if @proviene = 'i'
	begin
		if exists(
			select *
			from kilometro
			where	idempresa = @idempresa and
				origen = @origen and
				destino = @destino
		)
		begin
			update kilometro
			set
			distancia = @distancia
			where	idempresa = @idempresa and
				origen = @origen and
				destino = @destino
```

```
                end
                else
                begin
                        set @idkm = (select isnull(max(idkm),0) + 1 from kilometro)

                        insert into kilometro
                        (
                        idkm,
                        idempresa,
                        origen,
                        destino,
                        distancia,
                        idstatus
                        )
                        values
                        (
                        @idkm,
                        @idempresa,
                        @origen,
                        @destino,
                        @distancia,
                        @idstatus
                        )
                end
        end

        if @proviene = 'd'
        begin
                delete
                from kilometro
                where   idempresa = @idempresa and
                        idkm = @id
        end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_lock_object_control    Script Date: 02/07/2008 04:36:23 p.m. ******/
```

```
CREATE PROC sp_lock_object_control
                @object_name varchar(50)='dummy_object'

AS

DECLARE
        @last_locked_by varchar(50),
        @last_locked_at datetime


UPDATE object_control
        SET     locked_status_flag=1,
                last_locked_by=SESSION_USER,
                last_locked_at=getdate()
        WHERE           LTRIM([object_name]) = @object_name









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_mensaje_status_inserta_modifica    Script Date: 02/07/2008 04:36:25
p.m. ******/







/*
Inicia
Nombre:
sp_mensaje_status_inserta_modifica



Funcionalidad:
Inserta o actualiza los mensajes a enviar por correo cuando haya una solicitud en espera de autorizacion o
haya sido rechaza

Entradas:
```

Empresa en la que se esta trabajando
Status para el que se quiere el mensaje
Mensaje de correo a enviar

Proceso:
Verifica si existe un mensaje para ese status, si existe se actualiza si no existe se inserta

Salida:
Mensaje insertado o actualizado

Ejemplo:
exec sp_mensaje_status_inserta_modifica 1,'ep', 'Solicitud en espera de autorizacion de presupuesto'
Termina
*/

```
CREATE procedure [dbo].sp_mensaje_status_inserta_modifica
(
	@idempresa int,
	@idstatus char(2),
	@mensaje varchar(250)
)
as
begin
	declare
	@respuesta varchar(50)

	if exists(
		select
		*
		from mensaje_status
		where	idempresa = @idempresa and
			idstatus = @idstatus
	)
	begin
		update mensaje_status
		set
		mensaje = @mensaje
		where	idempresa = @idempresa and
			idstatus = @idstatus

		set @respuesta = 'Mensaje actualizado'

	end
	else
	begin
		insert into mensaje_status
		(
		idempresa,
		idstatus,
		mensaje
		)
		values
		(
		@idempresa,
		@idstatus,
		@mensaje
		)

		set @respuesta = 'Mensaje registrado'
	end

	select @respuesta as mensaje
```

```
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_menu_opciones_principales    Script Date: 02/07/2008 04:36:26 p.m.
******/




/*
Inicia
Nombre:
sp_menu_opciones_principales



Funcionalidad:
Obtiene la ubicacion principal de los objetos del menu asi como la descripcion de la ubicacion. Se utiliza para
dibujar el m

Entradas:
Empresa en la que se esta trabajando
Numero de empleado que esta firmado en la aplicacion

Proceso:
Obtiene las ubicaciones de los objetos del menu para pintar el menu principal

Salida:
Ubicaciones de los menus

Ejemplo:
exec [dbo].sp_menu_opciones_principales 1,123
Termina
*/

CREATE procedure [dbo].sp_menu_opciones_principales
(
        @idempresa int,
        @numempleado int
)
as
begin
        declare
        @idperfil int
```

```
	set @idperfil = (select idperfil from usuario where idempresa = @idempresa and numempleado =
@numempleado)

	select distinct
	ou.idubicacion,
	ou.descripcion
	from perfil_objeto as po
	inner join objeto as ob on po.idobjeto = ob.idobjeto
	inner join objeto_ubicacion as ou on ob.idubicacion = ou.idubicacion
	where	po.idempresa = @idempresa and
		po.idperfil = @idperfil
	order by ou.idubicacion

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_moneda_catalogo    Script Date: 02/07/2008 04:36:24 p.m. ******/



/*
Inicia
Nombre:
sp_moneda_catalogo



Funcionalidad:
Obtiene el catalogo de monedas

Entradas:
Sin entradas. Este catalago no depende de ninguna empresa

Proceso:
Obtiene el catalogo de monedas

Salida:
Catalogo de ivas

Ejemplo:
exec [dbo].sp_moneda_catalogo 'a'
Termina
*/
```

```
CREATE procedure [dbo].sp_moneda_catalogo
(
	@proviene char(1)
)
as
begin
	if @proviene = 'u'
	begin
		select
		mn.idmoneda,
		mn.descripcion
		from moneda as mn
		where	mn.idstatus = 'a'
	end

	if @proviene = 'a'
	begin
		select
		mn.idmoneda,
		mn.identificador,
		mn.descripcion,
		st.descripcion_1
		from moneda as mn
		left outer join status as st on mn.idstatus = st.idstatus
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_monto_limite_catalogo    Script Date: 02/07/2008 04:36:27 p.m.
******/




/*
Inicia
Nombre:
sp_monto_limite_catalogo


Funcionalidad:
```

Obtiene el catalogo de montos limite

Entradas:
Empresa en la que se esta trabajando
Tipo de operacion en la que aplicara la politica
Identificador del monto limite en caso de que sea quiera obtener un registro especifico

Proceso:
Obtiene el catalogo de monto limite

Salida:
Catalogo de montos limite

Ejemplo:
exec [dbo].sp_monto_limite_catalogo 1,'anticipo',0
Termina
*/

```
CREATE procedure [dbo].sp_monto_limite_catalogo
(
	@idempresa int,
	@tipo_operacion varchar(50),
	@idmonto_limite int
)
as
begin
	declare
	@idtipo_operacion int

	if @idmonto_limite = 0
	begin
		set @idtipo_operacion = (select
[dbo].fn_obtiene_tipo_operacion(@idempresa,rtrim(@tipo_operacion)))

		select
		ml.idmonto_limite as idmonto_limite,
		ni.descripcion as nivel,
		cg.descripcion as concepto_gasto,
		te.descripcion as territorio,
		ml.monto_limite as monto,
		mn.descripcion as moneda
		from monto_limite as ml
		inner join nivel as ni on ml.idnivel = ni.idnivel
		inner join concepto_gasto as cg on ml.idconcepto = cg.idconcepto
		inner join moneda as mn on ml.idmoneda = mn.idmoneda
		inner join territorio as te on ml.idterritorio = te.idterritorio
		where	ml.idempresa = ni.idempresa and
			ni.idempresa = cg.idempresa and
			ml.idempresa = @idempresa and
			cg.idstatus = 'a' and
			mn.idstatus = 'a' and
			ml.idtipo_operacion = @idtipo_operacion
	end

	else
	begin
		select
		ml.idmonto_limite,
		ml.idnivel,
		ml.idconcepto,
		lower(rtrim(tp.descripcion)),
		ml.idmoneda,
```

```
                    ml.idterritorio,
                    ml.monto_limite
                    from monto_limite as ml
                    inner join tipo_operacion as tp on ml.idtipo_operacion = tp.idtipo_operacion
                    where    ml.idempresa = tp.idempresa and
                             ml.idempresa = @idempresa and
                             ml.idmonto_limite = @idmonto_limite
          end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/*
Inicia
Nombre:
sp_monto_limite_inserta_modifica

Funcionalidad:
Inserta el monto limite maximo a solicitar por nivel, concepto de gasto, tipo de solicitud, empresa, moneda y
territorio

Entradas:
Identificador del registro en caso de que se trate de una modificacion o eliminacion
Identificador del nivel
Concepto de gasto
Tipo de solicitud
Empresa en la que se este trabajando
Moneda para la que se va a imponer el monto limite
Monto limite
Accion a seguir
          u = para la moneda en cuestion
          m = para todas las monedas utilizando el tipo de cambio vigente

Proceso:
Inserta, actualiza o elimina montos limite para una o varias monedas

Salida:
Tabla actualizada

Ejemplo:
exec [dbo].sp_monto_limite_inserta_modifica 0,1,1,1,1,1,1,5000.00,'m'
Termina
*/

CREATE procedure [dbo].sp_monto_limite_inserta_modifica
(
          @id int,
```

```
	@idnivel int,
	@idconcepto int,
	@tipo_operacion varchar(50),
	@idempresa int,
	@idmoneda int,
	@idterritorio int,
	@monto money,
	@proviene char(1) = 'u'	-- u = (Un registro, solo el de la moneda especificada)
				-- m = (Multiples registros, el de la moneda especificada se toma como
base
				--	y se calculan los de las demas monedas)
)
as
begin
	declare
	@idtipo_operacion int,
	@c_id int,
	@c_idnivel int,
	@c_idconcepto int,
	@c_tipo_operacion varchar(50),
	@c_idempresa int,
	@c_idmoneda int,
	@c_idterritorio int,
	@c_monto money,

	@idmonto_limite int

	set @idtipo_operacion = (select
[dbo].fn_obtiene_tipo_operacion(@idempresa,rtrim(@tipo_operacion)))

	if @proviene = 'u'
	begin
		if exists(
			select *
			from monto_limite
			where	idnivel = @idnivel and
				idconcepto = @idconcepto and
				idtipo_operacion = @idtipo_operacion and
				idempresa = @idempresa and
				idmoneda = @idmoneda and
				idterritorio = @idterritorio
		)
		begin
			update monto_limite
			set
			monto_limite = @monto
			where	idnivel = @idnivel and
				idconcepto = @idconcepto and
				idtipo_operacion = @idtipo_operacion and
				idempresa = @idempresa and
				idmoneda = @idmoneda and
				idterritorio = @idterritorio
		end
		else
		begin
			set @idmonto_limite = (select isnull(max(idmonto_limite),0) + 1 from monto_limite)

			insert into monto_limite
			(
			idmonto_limite,
			idnivel,
			idconcepto,
```

```
                        idtipo_operacion,
                        idempresa,
                        idmoneda,
                        idterritorio,
                        monto_limite
                        )
                        values
                        (
                        @idmonto_limite,
                        @idnivel,
                        @idconcepto,
                        @idtipo_operacion,
                        @idempresa,
                        @idmoneda,
                        @idterritorio,
                        @monto
                        )
                end
        end

        if @proviene = 'm'
        begin
                declare cursor_1 cursor for
                        select
                        @id as idmonto_limite,
                        @idnivel as idnivel,
                        @idconcepto as idconcepto,
                        @tipo_operacion as tipo_operacion,
                        @idempresa as idempresa,
                        idmoneda as idmoneda,
                        @idterritorio as idterritorio,
                        [dbo].fn_obtiene_tipo_cambio(@idmoneda,idmoneda,getdate(),@monto) as monto
                        from moneda
                        where idstatus = 'a'

                open cursor_1

                fetch next from cursor_1
                        into
                                @c_id,
                                @c_idnivel,
                                @c_idconcepto,
                                @c_tipo_operacion,
                                @c_idempresa,
                                @c_idmoneda,
                                @c_idterritorio,
                                @c_monto

                while @@fetch_status = 0
                        begin
                                if @c_monto > 0
                                begin
                                        exec [dbo].sp_monto_limite_inserta_modifica
@c_id,@c_idnivel,@c_idconcepto,@c_tipo_operacion,@c_idempresa,@c_idmoneda,@c_idterritorio,@c_mon
to,'u'
                                end


                                fetch next from cursor_1
                                into
                                        @c_id,
                                        @c_idnivel,
```

```
                                        @c_idconcepto,
                                        @c_tipo_operacion,
                                        @c_idempresa,
                                        @c_idmoneda,
                                        @c_idterritorio,
                                        @c_monto
                        end

                close           cursor_1
                deallocate      cursor_1

        end

        if @proviene = 'd'
        begin
                delete
                from monto_limite
                where   idempresa = @idempresa and
                        idmonto_limite = @id
        end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_muestra_datos_capturados_ver_detalle    Script Date: 02/07/2008
04:36:29 p.m. ******/




/*
Inicia
Nombre:
sp_muestra_datos_capturados_ver_detalle



Funcionalidad:
Muestra informacion acerca de datos capturados para un detalle

Entradas:
Folio de la solicitud
Opcion de mostrar informacion
        1 = mostrar datos que han sido capturados totalmente
        0 = mostrar datos en proceso de captura
```

```
Proceso:
Obtencion de informacion acerca de los detalles capturados

Salida:
Informacion de los detalles

Ejemplo:
exec sp_muestra_datos_capturados_ver_detalle 1,1
Termina
*/

CREATE procedure [dbo].sp_muestra_datos_capturados_ver_detalle
(
	@folio int,
	@datos_capturados tinyint
)
as
begin
	if @datos_capturados = 1
	begin
		select
		de.detalle,
		cg.descripcion,
		de.comentario,
		convert(char,de.fecha_gasto,103),
		de.subtotal,
		de.iva,
		de.propina,
		de.tua,
		de.total,
		de.inc_pol
		from solicitud as so
		inner join detalle_solicitud as de on so.folio = de.folio
		inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
		where	so.folio = @folio and
			so.idempresa = cg.idempresa
		order by de.detalle asc
	end

	else
	begin
		select
		de.detalle,
		cg.descripcion,
		case
			when len(de.comentario) < = 25 then de.comentario
			else left(de.comentario,23) + '...'
		end as comentario,
		'ver',
		de.total
		from solicitud as so
		inner join detalle_solicitud as de on so.folio = de.folio
		inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
		where	so.folio = @folio and
			so.idempresa = cg.idempresa
		order by de.detalle asc
	end
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_muestra_datos_grale    Script Date: 02/07/2008 04:36:22 p.m. ******/



/*
Inicia
Nombre:
sp_muestra_datos_grale


Funcionalidad:
Muestra los datos generales de una solicitud

Entradas:
Folio de la solicitud

Proceso:
Obtencion de informacion general de la solicitud

Salida:
Informacion general de la solicitud

Ejemplo:
exec sp_muestra_datos_grale 1
Termina
*/

CREATE procedure [dbo].sp_muestra_datos_grale
(
	@folio int
)
as
begin
	select top 1
	so.folio,
	us.nombre + ' ' + us.paterno,
	[dbo].fn_obtiene_total_por_folio(so.folio),
	m1.descripcion,
	m2.descripcion,
	st.descripcion_1,
	te.descripcion,
	tp.nombre,
	so.descripcion
	from solicitud as so
	inner join tipo_operacion as tp on tp.idtipo_operacion = so.idtipo_operacion
```

```
        left outer join detalle_solicitud as de on so.folio = de.folio
        inner join usuario as us on so.numempleado = us.numempleado
        inner join moneda as m1 on so.idmoneda = m1.idmoneda
        inner join moneda as m2 on so.idmoneda_pago = m2.idmoneda
        inner join status as st on so.idstatus = st.idstatus
        inner join territorio as te on so.idterritorio = te.idterritorio
        where   so.folio = @folio and
                so.idempresa = us.idempresa and
                so.idempresa = tp.idempresa
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_muestra_distribucion    Script Date: 02/07/2008 04:36:30 p.m. ******/




/*
Inicia
Nombre:
sp_muestra_distribucion



Funcionalidad:
Muestra los valores de distribucion entre los centros de costo

Entradas:
Folio del detalle

Proceso:
Obtencion de informacion de distribucion

Salida:
Informacion acerca de la distribucion de un detalle

Ejemplo:
exec sp_muestra_distribucion 1
Termina
*/

CREATE procedure [dbo].sp_muestra_distribucion
(
        @detalle int
```

```
)
as
begin
	select
	cc.identificador,
	cc.descripcion,
	sum(ds.porcentaje),
	sum(de.subtotal*ds.porcentaje/100) as subtotal,
	sum(de.iva*ds.porcentaje/100) as iva,
	sum(de.propina*ds.porcentaje/100) as propina,
	sum(de.tua*ds.porcentaje/100) as tua,
	sum(de.total*ds.porcentaje/100) as total,
	ds.distribucion
	from solicitud as so
	inner join detalle_solicitud as de on so.folio = de.folio
	inner join distribucion_solicitud as ds on de.detalle = ds.detalle
	inner join centro_costo as cc on ds.ccostos = cc.idccostos
	where	so.idempresa = cc.idempresa and
		de.detalle = @detalle and
		ds.porcentaje > 0
	group by cc.identificador, cc.descripcion,ds.distribucion
end


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_nivel_catalogo    Script Date: 02/07/2008 04:36:23 p.m. ******/




/*
Inicia
Nombre:
sp_nivel_catalogo



Funcionalidad:
Obtiene los niveles de responsabilidad vigentes para la empresa

Entradas:
Numero de empresa en la que se esta trabajando

Proceso:
Obtencion de los niveles por empresa

Salida:
Niveles
```

```
Ejemplo:
exec sp_nivel_catalogo 1
Termina
*/

CREATE procedure [dbo].sp_nivel_catalogo
(
	@idempresa int
)
as
begin
	select
	idnivel,
	descripcion
	from nivel
	where	idempresa = @idempresa
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_objeto_catalogo    Script Date: 02/07/2008 04:36:24 p.m. ******/



/*
Inicia
Nombre:
sp_objeto_catalogo



Funcionalidad:
Obtiene el catalogo de objetos del menu

Entradas:
No tiene entradas

Proceso:
Obtiene el catalogo de objetos del menu

Salida:
Catalogo de objetos

Ejemplo:
```

```
exec [dbo].sp_objeto_catalogo
Termina
*/


CREATE procedure [dbo].sp_objeto_catalogo
as
begin
	select
	idobjeto,
	descripcion
	from objeto
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_obtiene_datos_detalle    Script Date: 02/07/2008 04:36:22 p.m. ******/




/*
Inicia
Nombre:
sp_obtiene_datos_detalle


Funcionalidad:
Muestra los datos particulares del detalle de una solicitud

Entradas:
Numero de detalle de la solicitud

Proceso:
Obtiene algunos datos del detalle de las solicitudes

Salida:
Datos de un detalle de solicitud

Ejemplo:
exec sp_obtiene_datos_detalle 2
Termina
*/
```

```
CREATE procedure [dbo].sp_obtiene_datos_detalle
(
	@detalle int
)
as
begin
	select
	idconcepto,
	comentario,
	[dbo].sp_convierte_fecha_a_texto(convert(char,fecha_gasto,103)) as fecha_gasto,
	inc_pol
	from detalle_solicitud
	where	detalle = @detalle
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_obtiene_datos_grales    Script Date: 02/07/2008 04:36:28 p.m. ******/




/*
Inicia
Nombre:
sp_obtiene_datos_grales



Funcionalidad:
Muestra los datos generales de una solicitud

Entradas:
Folio de la solicitud

Proceso:
Obtiene algunos datos generales de las solicitudes

Salida:
Datos generales de una solicitud

Ejemplo:
exec sp_obtiene_datos_grales 2
Termina
```

```
*/

CREATE procedure [dbo].sp_obtiene_datos_grales
(
	@folio int
)
as
begin
	select
	so.idmoneda,
	so.idmoneda_pago,
	so.idterritorio,
	so.descripcion
	from solicitud as so
	where	so.folio = @folio
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_obtiene_depositos_por_empleado    Script Date: 02/07/2008
04:36:22 p.m. ******/




/*
Inicia
Nombre:
sp_obtiene_depositos_por_empleado



Funcionalidad:
Muestra los depositos a favor de la empresa hechos por los empleados

Entradas:
Empresa en la que se esta trabajando
Numero de empleado que hace la comprobacion de anticipo
Moneda en que esta hecha la comprobacion
Folio del deposito

Proceso:
Obtiene los datos del deposito

Salida:
```

Datos del deposito

Ejemplo:
exec [dbo].sp_obtiene_depositos_por_empleado 1,1,1,1
Termina
*/
```
CREATE procedure [dbo].sp_obtiene_depositos_por_empleado
(
	@idempresa int,
	@numempleado int,
	@idmoneda int,
	@folio int
)
as
begin
	/*
	Si no se trata de un folio especifico, se presentan todos los depositos
	que no ha sido utilizados en otras comprobaciones
	*/
	if @folio = 0
	begin
		select
		folio,
		referencia,
		convert(char, fecha, 103) as fecha,
		ba.nombre,
		isnull([dbo].fn_obtiene_tipo_cambio(de.idmoneda,@idmoneda,fecha,de.monto),0) as monto,
		mv.descripcion,
		isnull(de.monto,0),
		md.descripcion
		from deposito as de
		inner join banco as ba on de.idbanco = ba.idbanco
		inner join moneda as mv on @idmoneda = mv.idmoneda
		inner join moneda as md on de.idmoneda = md.idmoneda
		where	de.idempresa = @idempresa and
			de.numempleado = @numempleado and
			de.folio not in (select folio_deposito from detalle_deposito)
	end

	/*
	Se presentan todos los depositos
	*/
	if @folio <> 0
	begin
		select
		folio,
		referencia,
		convert(char, fecha, 103) as fecha,
		ba.nombre,
		isnull([dbo].fn_obtiene_tipo_cambio(de.idmoneda,@idmoneda,fecha,de.monto),0) as monto,
		mv.descripcion,
		isnull(de.monto,0),
		md.descripcion
		from deposito as de
		inner join banco as ba on de.idbanco = ba.idbanco
		inner join moneda as mv on @idmoneda = mv.idmoneda
		inner join moneda as md on de.idmoneda = md.idmoneda
		where	de.idempresa = @idempresa and
			de.numempleado = @numempleado
		end
end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_obtiene_folios    Script Date: 02/07/2008 04:36:29 p.m. ******/




/*
Inicia
Nombre:
sp_obtiene_folios



Funcionalidad:
Muestra los folios de las solicitudes que corresponden al status que se este buscando

Entradas:
Empresa en la que se esta trabajando
Numero de empleado que se firma en la aplicacion
Tipo de operacion (anticipo,pago,etc)
Tipo de status de operaciones que se van a buscar
        (n = solicitudes nuevas)
        (e = en espera de autorizacion)
        (r = rechazadas)

Proceso:
Obtencion de folios de solicitudes

Salida:
Numeros de folio

Ejemplo:
exec sp_obtiene_folios 12,123,'anticipo','e'
Termina
*/

CREATE procedure [dbo].sp_obtiene_folios
(
        @idempresa int,
        @numempleado int,
        @tipo_operacion varchar(50),
        @tipo_status char(1)
)
as
```

```
begin
	declare
	@idtipo_operacion int

	set @idtipo_operacion = (select [dbo].fn_obtiene_tipo_operacion(@idempresa,@tipo_operacion))

	select
	so.folio,
	convert(char,so.fecha_solicitud,103) as fecha_solicitud,
	case
		when len(so.descripcion) <= 25 then so.descripcion
		else left(so.descripcion,22) + '...'
	end as descripcion,
	sum(isnull(de.total,0)) as total,
	st.descripcion_1
	from solicitud as so
	left outer join detalle_solicitud as de on so.folio = de.folio
	inner join status as st on so.idstatus = st.idstatus
	where	substring(so.idstatus,1,1) = @tipo_status and
		so.idtipo_operacion = @idtipo_operacion and
		so.idempresa = @idempresa and
		so.numempleado = @numempleado
	group by so.folio,so.fecha_solicitud,so.descripcion,st.descripcion_1
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_obtiene_folios_relacionados    Script Date: 02/07/2008 04:36:30 p.m.
******/




/*
Inicia
Nombre:
sp_obtiene_folios_relacionados



Funcionalidad:
Muestra los folios que estan relacionados con otros y los folios que se relacionan con este

Entradas:
El folio del que queremos saber los relacionados con el y los folios con los que el se relaciona
```

```
Accion a seguir
	No se toma en cuenta esta variable

Proceso:
Obtencion de folios relacionados

Salida:
Numeros de folio

Ejemplo:
exec [dbo].sp_obtiene_folios_relacionados 5,'fr'
Termina
*/

CREATE procedure [dbo].sp_obtiene_folios_relacionados
(
	@folio int,
	@proviene char(2)
)
as
begin
	/*
	Esta relacion es uno a uno, es decir, un folio solo puede estar afectando a un folio
	Titulo = Este folio afecta a:
	*/
	select
	fp.folio_principal,
	tp.nombre,
	'P' as tipo_de_folio
	from folio_principal_folio_relacionado as fp
	inner join solicitud as so on fp.folio_principal = so.folio
	inner join tipo_operacion as tp on so.idtipo_operacion = tp.idtipo_operacion
	where	fp.folio_relacionado = @folio and
		so.idempresa = tp.idempresa
	/*
	Esta relacion es uno a muchos, es decir, un folio solo estar afectado por muchos folios
	Titulo = Este folio es afectado por:
	*/
	union
	select
	fp.folio_relacionado,
	tp.nombre,
	'R' as tipo_de_folio
	from folio_principal_folio_relacionado as fp
	inner join solicitud as so on fp.folio_relacionado = so.folio
	inner join tipo_operacion as tp on so.idtipo_operacion = tp.idtipo_operacion
	where	fp.folio_principal = @folio and
		so.idempresa = tp.idempresa
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
```

```
GO


/****** Object:  Stored Procedure dbo.sp_obtiene_historial    Script Date: 02/07/2008 04:36:29 p.m. ******/




/*
Inicia
Nombre:
sp_obtiene_historial



Funcionalidad:
Obtiene toda la historia de una solicitud en la aplicacion

Entradas:
Folio de la solicitud de la que se quiere obtener el historial

Proceso:
Obtencion del historial de la solicitud

Salida:
Numeros de folio

Ejemplo:
exec [dbo].sp_obtiene_historial 6
Termina
*/

CREATE procedure [dbo].sp_obtiene_historial
(
	@folio int
)
as
begin
	select
	idhistorial,
	rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as persona,
	st.idstatus,
	st.descripcion_2,
	convert(char,fecha,103) as fecha,
	convert(char,fecha,108) as fecha,
	observacion
	from solicitud as so
	inner join usuario as us on so.idempresa = us.idempresa
	inner join historial_solicitud as hs on so.folio = hs.folio
	inner join status as st on hs.idstatus = st.idstatus
	where	so.folio = @folio and
		hs.numempleado = us.numempleado
	order by hs.idhistorial asc

end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_obtiene_solicitudes_por_status    Script Date: 02/07/2008 04:36:28
p.m. ******/




/*
Inicia
Nombre:
sp_obtiene_solicitudes_por_status



Funcionalidad:
Obtiene solicitudes en funcion del status y el tipo de operacion proporcionados

Entradas:
Empresa en la que se esta trabajando
Tipo de solicitud que se quiere obtener
Status que se pretende buscar

Proceso:
Busca solicitudes del tipo y status proporcionados

Salida:
Folio de las solicitudes encontradas

Ejemplo:
exec [dbo].sp_obtiene_solicitudes_por_status 1,1,'ej'
Termina
*/

CREATE procedure [dbo].sp_obtiene_solicitudes_por_status
(
	@idempresa int,
	@idtipo_operacion int,
	@idstatus char(2)
)
as
begin
	select
	so.folio,
	st.descripcion_1,
	rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as solicitante
	from solicitud as so
```

```
	inner join status as st on so.idstatus = st.idstatus
	inner join usuario as us on so.numempleado = us.numempleado
	where	so.idempresa = @idempresa and
		so.idempresa = us.idempresa and
		so.idtipo_operacion = @idtipo_operacion and
		so.idstatus = @idstatus
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_obtiene_tipos_de_operacion    Script Date: 02/07/2008 04:36:23 p.m.
******/




/*
Inicia
Nombre:
sp_obtiene_tipos_de_operacion



Nota:
Este procedimiento se utiliza en las pantallas de flujo de autorizacion dentro de la aplicacion

Funcionalidad:
Obtiene los tipos de operacion vigentes por empresa

Entradas:
Numero de empresa en la que se esta trabajando
Accion a seguir
	u = muestra al usuario solo los tipos de operacion activos
	a = muestra al administrador los tipos de operacion para esa empresa

Proceso:
Obtencion de los tipos de operacion vigentes por empresa

Salida:
Tipos de operacion

Ejemplo:
exec sp_obtiene_tipos_de_operacion 1,'u'
Termina
s*/
```

```
CREATE procedure [dbo].sp_obtiene_tipos_de_operacion
(
	@idempresa int,
	@proviene char(1)
)
as
begin
	if @proviene = 'u'
	begin
		select
		idtipo_operacion,
		descripcion,
		nombre
		from tipo_operacion
		where	idempresa = @idempresa and
			idstatus = 'a'
	end

	if @proviene = 'a'
	begin
		select
		idtipo_operacion,
		descripcion,
		nombre
		from tipo_operacion
		where	idempresa = @idempresa
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_obtiene_tipos_operacion_por_empleado    Script Date: 02/07/2008
04:36:28 p.m. ******/




/*
Inicia
Nombre:
sp_obtiene_tipos_operacion_por_empleado


Funcionalidad:
```

```
Obtiene los tipos de operacion por empleado

Entradas:
Numero de empresa en la que se esta trabajando
Numero de empleado del que se quieren obtener las solicitudes

Proceso:
Busca las solicitudes que un empleado ha hecho

Salida:
Tipos de operacion

Ejemplo:
exec [dbo].sp_obtiene_tipos_operacion_por_empleado 1,123
Termina
*/

CREATE procedure [dbo].sp_obtiene_tipos_operacion_por_empleado
(
	@idempresa int,
	@numempleado int
)
as
begin
	if @numempleado <> 0
	begin
		select distinct
		tp.descripcion,
		tp.nombre
		from solicitud as so
		inner join tipo_operacion as tp on so.idtipo_operacion = tp.idtipo_operacion
		where	so.idempresa = tp.idempresa and
			so.idempresa = @idempresa and
			so.numempleado = @numempleado
	end

	if @numempleado = 0
	begin
		select
		rtrim(lower(descripcion)),
		nombre
		from tipo_operacion
		where	idempresa = @idempresa and
			idstatus = 'a'
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Stored Procedure dbo.sp_operaciones_perfil    Script Date: 02/07/2008 04:36:25 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_operaciones_perfil


Funcionalidad:
Inserta, actualiza o elimina el numero de solicitudes en proceso que un usuario, con base en su perfil, puede
tener

Entradas:
Numero de empresa en la que se esta trabajando
Identificador del perfil de usuario
Identificador del objeto del menu
Numero de solicitudes en proceso que ese perfil puede tener
Accion a seguir
        i = inserta o actualiza
        d = elimina registro

Proceso:
Inserta, actualiza o elimina registro

Salida:
Numero de solicitudes en proceso por perfil

Ejemplo:
exec [dbo].sp_operaciones_perfil 1,1,1,5,'i'
Termina
*/

CREATE procedure [dbo].sp_operaciones_perfil
(
        @idempresa int,
        @idperfil int,
        @idobjeto int,
        @op_proceso int,
        @proviene char(1)
)
as
begin
        declare
        @tipo_operacion varchar(50),
        @idtipo_operacion int

        set @tipo_operacion = (
                select tp.descripcion
                from objeto as ob
                inner join tipo_operacion as tp on lower(rtrim(ob.descripcion)) = lower(rtrim(tp.nombre))
                where   ob.idobjeto = @idobjeto and
                        tp.idempresa = @idempresa
        )
```

```
	set @idtipo_operacion = (select [dbo].fn_obtiene_tipo_operacion(@idempresa,@tipo_operacion))

	if @proviene = 'i'
	begin
		if exists(
			select
			1
			from operacion_proceso
			where	idempresa = @idempresa and
				idperfil = @idperfil and
				idtipo_operacion = @idtipo_operacion
		)
		begin
			update operacion_proceso
			set
			op_proceso = @op_proceso
			where	idempresa = @idempresa and
				idperfil = @idperfil and
				idtipo_operacion = @idtipo_operacion
		end
		else
		begin
			insert into operacion_proceso
			(
			idempresa,
			idperfil,
			idtipo_operacion,
			op_proceso
			)
			values
			(
			@idempresa,
			@idperfil,
			@idtipo_operacion,
			@op_proceso
			)
		end
	end

	if @proviene = 'd'
	begin
		delete from operacion_proceso
		where	idempresa = @idempresa and
			idperfil = @idperfil and
			idtipo_operacion = @idtipo_operacion
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Stored Procedure dbo.sp_perfil_catalogo    Script Date: 02/07/2008 04:36:23 p.m. ******/



/*
Inicia
Nombre:
sp_perfil_catalogo



Funcionalidad:
Obtiene los perfiles vigentes por empresa

Entradas:
Numero de empresa en la que se esta trabajando

Proceso:
Obtencion de los perfiles vigentes por empresa

Salida:
perfiles

Ejemplo:
exec sp_perfil_catalogo 1
Termina
*/

CREATE procedure [dbo].sp_perfil_catalogo
(
	@idempresa int
)
as
begin
	select
	idperfil,
	descripcion
	from perfil
	where	idempresa = @idempresa
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Stored Procedure dbo.sp_perfil_inserta_modifica    Script Date: 02/07/2008 04:36:23 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_perfil_inserta_modifica


Funcionalidad:
Inserta, actuliza o elimina perfiles de usuario

Entradas:
Identificador del perfil, en caso de que se trate de una modificacion o eliminacion
Empresa en la que se esta trabajando
Nombre del perfil
Accion a seguir
        i = insercion
        u = actualizacion
        d = eliminacion del perfil

Proceso:
Insercion, actualizacion o eliminacion de un registro

Salida:
Perfil insertado o eliminado

Ejemplo:
exec sp_perfil_inserta_modifica 0,1,'Perfil Nuevo','i'
Termina
*/

CREATE procedure [dbo].sp_perfil_inserta_modifica
(
        @id int,
        @idempresa int,
        @descripcion varchar(50),
        @proviene char(1)
)
as
begin


        declare
                @idperfil int

        if ltrim(rtrim(@descripcion)) = ''
        begin
                set @descripcion = 'no definida'
        end

        begin transaction perfil

        if @proviene = 'i'
        begin
```

```
		set @idperfil = (select isnull(max(idperfil),0) + 1 from perfil)

		insert into perfil
		(
		idperfil,
		idempresa,
		descripcion
		)
		values
		(
		@idperfil,
		@idempresa,
		upper(ltrim(rtrim(@descripcion)))
		)
	end

	if @proviene = 'u'
	begin
		update perfil
		set
		descripcion = upper(ltrim(rtrim(@descripcion)))
		where	idperfil = @id and
			idempresa = @idempresa

	end

	if @proviene = 'd'
	begin
		delete from perfil
		where	idperfil = @id and
			idempresa = @idempresa
	end


	commit transaction perfil


end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_perfil_objeto_catalogo    Script Date: 02/07/2008 04:36:22 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_perfil_objeto_catalogo



Nota:
Este stored procedure sirve para dibujar los objetos en el menu

Funcionalidad:
Obtiene el catalogo de objetos del menu

Entradas:
Ubicacion del menu

Proceso:
Busca todos los objetos que se encuentren en la ubicacion dada. Por ejemplo, si se proporciona 'solicitudes',
el procedimien

Salida:
Objetos del menu que se encuentren bajo una ubicacion dada

Ejemplo:
exec [dbo].sp_perfil_objeto_catalogo 'Reportes'
Termina
*/

CREATE procedure [dbo].sp_perfil_objeto_catalogo
(
	@ubicacion varchar(50)
)
as
begin
	declare
	@sql varchar(1000)

	if lower(rtrim(@ubicacion)) = 'solicitudes'
	begin
		set @sql = '
		select
		ob.idobjeto,
		ob.descripcion
		from objeto as ob
		inner join objeto_ubicacion as ou on ob.idubicacion = ou.idubicacion
		inner join tipo_operacion as tp on lower(rtrim(ob.descripcion)) = lower(rtrim(tp.nombre))
		where	rtrim(ou.descripcion) = "'+rtrim(@ubicacion)+'" and
			tp.idstatus = "a"
		'
	end
	else
	begin
		set @sql = '
		select
		ob.idobjeto,
		ob.descripcion
		from objeto as ob
		inner join objeto_ubicacion as ou on ob.idubicacion = ou.idubicacion
		where	rtrim(ou.descripcion) = "'+rtrim(@ubicacion)+'"
		order by ob.descripcion asc
		'
```

```
		end

		exec(@sql)
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_perfil_objeto_inserta_modifica    Script Date: 02/07/2008 04:36:25
p.m. ******/




/*
Inicia
Nombre:
sp_perfil_objeto_inserta_modifica



Funcionalidad:
Inserta o elimina registros que indican qué perfil de usuario puede hacer qué cosa

Entradas:
Identificador del perfil
Identificador de la empresa
Identificador del objeto del menu
Accion a seguir
		i = inserta relacion perfil objeto
		d = elimina relacion perfil objeto

Proceso:
Inserta o elimina una relacion perfil objeto

Salida:
Registro insertado o eliminado

Ejemplo:
exec [dbo].sp_perfil_objeto_inserta_modifica 1,1,1,'d'
Termina
*/

CREATE procedure [dbo].sp_perfil_objeto_inserta_modifica
(
		@idperfil int,
		@idempresa int,
```

```
	@idobjeto int,
	@proviene char(1),

	@commit tinyint = 1
)
as begin

	declare
	@existe tinyint,
	@mensaje varchar(250)

	begin transaction trans

	/*Si se pretende insertar un registro nuevo*/
	if @proviene = 'i'
	begin
		set @existe = 0

		/*
		Se busca si esa relacion objeto perfil existe
		*/
		set @existe = (
			select 1
			from perfil_objeto
			where	idperfil = @idperfil and
				idempresa = @idempresa and
				idobjeto = @idobjeto
		)

		if @existe = 1
		begin
			set @existe = 1
		end
		else
		begin
			set @existe = 0
		end

		/*
		Si la relacion no existe entonces se inserta
		*/
		if @existe = 0
		begin

			set dateformat dmy insert into perfil_objeto
			(
			idperfil,
			idempresa,
			idobjeto
			)
			values(
			@idperfil,
			@idempresa,
			@idobjeto
			)
			select ''
		end
		/*
		Si la relacion ya existe entonces no se hace la insercion
		*/
		if @existe = 1
		begin
```

```
                        select 'La relacion perfil / objeto del menu, ya existe'
                end
                else
                begin
                        select @mensaje
                end

        end

        /*
        EN ESTE PROCEDIMIENTO NO EXISTE LA ACTUALIZACION YA QUE UNA RELACION NO SE
PUEDE ACTUALIZAR,
        SINO ELIMINAR Y DAR DE ALTA OTRA
        */

        if @proviene = 'd'
        begin
                if @idobjeto <> 0
                begin
                        delete from perfil_objeto
                        where   idperfil = @idperfil and
                                idempresa = @idempresa and
                                idobjeto = @idobjeto
                end

        end


        if @commit = 1
        begin
                commit transaction trans
                print 'Transaction commited'

        end

        if @commit <> 1
        begin
                rollback transaction trans
                print 'Transaction rolled-back'
        end
end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_perfil_objeto_muestra_datos    Script Date: 02/07/2008 04:36:26 p.m.
******/
```

```
/*
Inicia
Nombre:
sp_perfil_objeto_muestra_datos



Funcionalidad:
Muestra los datos de un perfil dado

Entradas:
Numero de la empresa en la que se esta trabajando
Identificador del perfil
Accion a seguir
        n = presenta solo el nombre del perfil
        otro = presenta el objeto al que ese perfil tiene acceso, su nombre y el nombre del perfil

Proceso:
Presenta los datos o accesos de un perfil de usuario

Salida:
Datos o accesos de un perfil, segun sea la opcion

Ejemplo:
exec sp_perfil_objeto_muestra_datos 1,1,'n'
Termina
*/



CREATE procedure [dbo].sp_perfil_objeto_muestra_datos
(
        @idempresa int,
        @idperfil int,
        @proviene char(1)
)
as
begin
        begin

                if @proviene = 'n'
                begin
                        select
                        pe.descripcion
                        from perfil as pe
                        where   pe.idempresa = @idempresa and
                                pe.idperfil = @idperfil
                end
                else
                begin
                        select
                        ob.idobjeto,
                        ob.descripcion,
                        pe.descripcion
                        from perfil as pe
                        inner join perfil_objeto as po on pe.idperfil = po.idperfil and pe.idempresa =
po.idempresa
                        inner join objeto as ob on po.idobjeto = ob.idobjeto
```

```
                        where   pe.idempresa = @idempresa and
                                pe.idperfil = @idperfil
                end

        end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_ppto_detalle_inserta_modifica    Script Date: 02/07/2008 04:36:30
p.m. ******/




/*
Inicia
Nombre:
sp_ppto_detalle_inserta_modifica



Funcionalidad:
Registra el tipo de ppto que tiene gastado un detalle de una solicitud

Entradas:
Identificador del registro de ppto (en la tabla de presupuesto este numero identifica la cuenta contable, mes,
monto, etc)
Identificador del detalle dentro de la solicitud
Monto gastado o comprometido de ese detalle
Fecha de gasto
Accion a seguir
        i = insertar relacion
        d = eliminar relacion
Tipo de ppto
        c = comprometido, cuando esta en proceso de autorizacion
        g = gastado, cuanto la solicitud ha sido autorizada

Proceso:
Inserta o elimina una relacion de detalle ppto

Salida:
Relacion insertada o actualizada

Ejemplo:
```

```
Termina
*/

CREATE procedure [dbo].sp_ppto_detalle_inserta_modifica
(
	@idppto int,
	@detalle int,
	@monto money,
	@fecha datetime,
	@proviene char(1),
	@tipo char(1)
)
as
begin
	if @proviene = 'i'
	begin
		insert into ppto_detalle
		(
		idreg,
		idppto,
		detalle,
		monto,
		tipo,
		fecha
		)
		select isnull(max(idreg),0) + 1,
		@idppto,
		@detalle,
		@monto,
		@tipo,
		@fecha
		from ppto_detalle

	end

	if @proviene = 'd'
	begin
		delete from ppto_detalle
		where	detalle = @detalle
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_ppto_inserta_modifica    Script Date: 02/07/2008 04:36:31 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_ppto_inserta_modifica



Funcionalidad:
Inserta en la tabla ppto_detalle los gastos comprometidos de una solicitud

Entradas:
Folio de la solicitud de la que se va a comprometer ppto

Proceso:
Verifica si hay ppto disponible para los gastos de esa solicitud. Si hay, compromete el ppto. Si no hay,
entonces crea un registro en la tabla de ppto y compromete los gastos de esa solicitud

Salida:
Presupuesto comprometido de solicitud

Ejemplo:
exec sp_ppto_inserta_modifica 5
Termina
*/


CREATE procedure [dbo].sp_ppto_inserta_modifica
(
	@folio int
)
as
begin
	declare
	@idempresa int,
	@idmoneda int,

	@c_detalle int,
	@c_fecha_gasto datetime,
	@c_idconcepto int,
	@c_ccostos int,
	@c_porcentaje_distribucion float,
	@c_tipo varchar(12),
	@c_porcentaje_ded int,
	@c_ccontable char(6),
	@c_total money,

	@ppto_idmoneda int,
	@total_detalle_a_moneda_ppto money,

	@idppto int

	begin transaction compromete_ppto

	set @idempresa = (select idempresa from solicitud where folio = @folio)
	set @idmoneda = (select idmoneda from solicitud where folio = @folio)

	/*
	Creacion de tabla temporal
```

```
*/
create table #temp(
	detalle int,
	fecha_gasto datetime,
	idconcepto int,
	ccostos int,
	porcentaje_distribucion float,
	tipo varchar(12),
	porcentaje_ded numeric,
	ccontable char(6),
	total money
)

/*
Se guardan en esa tabla temporal los gastos de la solicitud (de todos sus detalles)
A traves del procedimiento sp_ppto_obtiene_gastos
*/
insert into #temp
	exec [dbo].sp_ppto_obtiene_gastos @folio

/*
Se obtienen los resultados de los gastos de la solicitud y se meten a un cursor
para estudiarlos individualmente
*/
declare cursor_ppto cursor for
	select * from #temp

open cursor_ppto

fetch next from cursor_ppto
	into
		@c_detalle,
		@c_fecha_gasto,
		@c_idconcepto,
		@c_ccostos,
		@c_porcentaje_distribucion,
		@c_tipo,
		@c_porcentaje_ded,
		@c_ccontable,
		@c_total

while @@fetch_status = 0
	begin
		/*
		Se inicializa la variable @idppto en nulo
		*/
		set @idppto = null

		/*
		Se selecciona el registro en donde existe la combinacion
		empresa, cuenta contable, año, mes y centro de costo que corresponda a los
		datos del detalle de la solicitud
		*/
		set @idppto = (
			select idppto
			from presupuesto
			where	idempresa = @idempresa and
				rtrim(ccontable) = @c_ccontable and
				anio = year(@c_fecha_gasto) and
				mes = month(@c_fecha_gasto) and
				identidad = @c_ccostos
		)
```

```
			/*
			Si si existe el registro quiere decir que si hay ppto dado de alta
			y se procede a actualizar ese registro con los nuevos datos del detalle de la
solicitud
			*/
			if @idppto is not null
			begin
				set @ppto_idmoneda = (select
[dbo].fn_obtiene_ppto_idmoneda(@idempresa,@c_ccontable,@c_fecha_gasto,@c_ccostos))
				set @total_detalle_a_moneda_ppto = (select
[dbo].fn_obtiene_tipo_cambio(@idmoneda,@ppto_idmoneda,@c_fecha_gasto,@c_total))

				/*
				Se actualiza el ppto con base en la moneda en que estaba dado de alta
ese registro
				*/
				update presupuesto
				set
				ppto_comprometido = isnull(ppto_comprometido,0) +
isnull(@total_detalle_a_moneda_ppto,0)
				where	idempresa = @idempresa and
					rtrim(ccontable) = @c_ccontable and
					anio = year(@c_fecha_gasto) and
					mes = month(@c_fecha_gasto) and
					identidad = @c_ccostos

				/*
				Una vez que se actualizo la tabla de ppto, se compromete en la tabla
ppto_detalle
				*/
				exec [dbo].sp_ppto_detalle_inserta_modifica
@idppto,@c_detalle,@total_detalle_a_moneda_ppto,@c_fecha_gasto,'i','c'
			end

			/*
			Si no existe el registro entonces se crea
			*/
			if @idppto is null
			begin
				set @idppto = (select isnull(max(idppto),0) + 1 from presupuesto)
				insert into presupuesto
				(
				idppto,
				idempresa,
				ccontable,
				anio,
				mes,
				ppto_presupuestado,
				ppto_gastado,
				ppto_comprometido,
				idmoneda,
				identidad
				)
				values
				(
				@idppto,
				@idempresa,
				@c_ccontable,
				year(@c_fecha_gasto),
				month(@c_fecha_gasto),
				0,
				0,
```

```
                                        @c_total,
                                        @idmoneda,
                                        @c_ccostos
                                        )

                                        /*
                                        Y se compromete el nuevo ppto con base en los datos de la solicitud
                                        */
                                        exec [dbo].sp_ppto_detalle_inserta_modifica
@idppto,@c_detalle,@c_total,@c_fecha_gasto,'i','c'
                                end

                                fetch next from cursor_ppto
                                into
                                        @c_detalle,
                                        @c_fecha_gasto,
                                        @c_idconcepto,
                                        @c_ccostos,
                                        @c_porcentaje_distribucion,
                                        @c_tipo,
                                        @c_porcentaje_ded,
                                        @c_ccontable,
                                        @c_total
                        end

        close           cursor_ppto
        deallocate      cursor_ppto

        drop table #temp

        commit transaction compromete_ppto

end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_ppto_obtiene_gastos    Script Date: 02/07/2008 04:36:30 p.m. ******/




/*
Inicia
Nombre:
sp_ppto_obtiene_gastos
```

```
Funcionalidad:
Obtiene los gastos totales relevantes para el ppto por solicitud
Solo se toma en cuenta el subtotal y la propina para la evaluacion de ppto
Se divide en gastos deducibles y no deducibles

Entradas:
Folio de la solicitud de la que se quieren obtener los gastos

Proceso:
Obtiene los gastos deducibles y no deducibles de las solicitudes

Salida:
Informacion relevante para calcular el ppto necesario de las solicitudes

Ejemplo:
exec sp_ppto_obtiene_gastos 5
Termina
*/

CREATE procedure [dbo].sp_ppto_obtiene_gastos
(
	@folio int
)
as
begin
	select
	de.detalle,
	de.fecha_gasto,
	de.idconcepto,
	di.ccostos,
	di.porcentaje as porcentaje_distribucion,
	'Deducible' as tipo,
	cg.porcentaje_deducible,
	cg.ccontable_deducible,
	(de.subtotal + de.propina)*(di.porcentaje/100)*(cg.porcentaje_deducible/100) as total
	from detalle_solicitud as de
	inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
	inner join distribucion_solicitud as di on de.detalle = di.detalle
	where	de.folio = @folio and
		de.idempresa = cg.idempresa and
		cg.idempresa = di.idempresa and
		cg.porcentaje_deducible > 0
	union
	select
	de.detalle,
	de.fecha_gasto,
	de.idconcepto,
	di.ccostos,
	di.porcentaje as porcentaje_distribucion,
	'No deducible' as tipo,
	cg.porcentaje_no_deducible,
	cg.ccontable_no_deducible,
	(de.subtotal + de.propina)*(di.porcentaje/100)*(cg.porcentaje_no_deducible/100) as total
	from detalle_solicitud as de
	inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
	inner join distribucion_solicitud as di on de.detalle = di.detalle
	where	de.folio = @folio and
		de.idempresa = cg.idempresa and
		cg.idempresa = di.idempresa and
```

```
                    cg.porcentaje_no_deducible > 0
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_proveedor_catalogo    Script Date: 02/07/2008 04:36:26 p.m. ******/



/*
Inicia
Nombre:
sp_proveedor_catalogo



Funcionalidad:
Obtiene el catalogo de proveedores

Entradas:
Numero de empresa
Accion a seguir
        u = presenta informacion relevante a los usuario
        a = presenta informacion relevante a los administradores

Proceso:
Obtiene el catalogo de proveedores

Salida:
Catalogo de proveedores

Ejemplo:
exec [dbo].sp_proveedor_catalogo 1,'u'
Termina
*/



CREATE procedure [dbo].sp_proveedor_catalogo
(
        @idempresa int,
        @proviene char(1) = 'u'
)
as
begin
```

```
	if @proviene = 'u'
	begin
		select
		pr.idproveedor,
		pr.nombre
		from proveedor as pr
		inner join proveedor_empresa as pe on pr.idproveedor = pe.idproveedor
		where	pe.idstatus = 'a' and
			pe.idempresa = @idempresa
	end

	if @proviene = 'a'
	begin
		select
		pr.idproveedor,
		pr.nombre,
		pr.rfc,
		pr.telefono,
		pr.mail,
		pr.identificador,
		st.descripcion_1
		from proveedor as pr
		inner join proveedor_empresa as pe on pr.idproveedor = pe.idproveedor
		inner join status as st on pe.idstatus = st.idstatus
		where	pe.idempresa = @idempresa
	end


end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_proveedor_concepto_catalogo    Script Date: 02/07/2008 04:36:28
p.m. ******/






/*
Inicia
Nombre:
sp_proveedor_concepto_catalogo
```

```
Funcionalidad:
Obtiene la relacion de a qué proveedores se les pueden cargar qué conceptos de gasto

Entradas:
Numero de empresa

Identificador del proveedor

Proceso:
Busca los conceptos de gasto que aplican para ese proveedor

Salida:
Conceptos de gasto

Ejemplo:
exec [dbo].sp_proveedor_concepto_catalogo 1,2
Termina
*/


CREATE procedure [dbo].sp_proveedor_concepto_catalogo
(
	@idempresa int,
	@idproveedor int
)
as
begin
	select
	cg.idconcepto,
	cg.descripcion
	from proveedor_concepto as pc
	inner join concepto_gasto as cg on pc.idconcepto = cg.idconcepto
	where	pc.idempresa = cg.idempresa and
		pc.idempresa = @idempresa and
		pc.idproveedor = @idproveedor
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_proveedor_detalle_catalogo    Script Date: 02/07/2008 04:36:26 p.m.
******/




/*
```

Inicia
Nombre:
sp_proveedor_detalle_catalogo

Funcionalidad:
Obtiene datos del proveedor dado

Entradas:
Numero de empresa en la que se esta trabajando
Identificador del proveedor

Proceso:
Obtiene los datos de detalle del proveedor dado

Salida:
Datos del proveedor

Ejemplo:
exec sp_proveedor_detalle_catalogo 1,1
Termina
*/

```
CREATE procedure [dbo].sp_proveedor_detalle_catalogo
(
	@idempresa int,
	@idproveedor int
)
as
begin
	select
	em.identificador,
	em.nombre,
	em.rfc,
	em.calle,
	em.numero,
	em.interior,
	em.colonia,
	em.delegacion,
	em.estado,
	em.pais,
	em.cp,
	em.telefono,
	em.fax,
	lower(em.mail) as email,
	em.serv_producto,
	bn.nombre,
	em.no_cuenta,
	st.descripcion_1
	from proveedor as em
	left outer join banco as bn on em.idbanco = bn.idbanco
	inner join proveedor_empresa as pe on em.idproveedor = pe.idproveedor
	inner join status as st on pe.idstatus = st.idstatus
	where	pe.idempresa = @idempresa and
		em.idproveedor = @idproveedor

end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_recupera_password    Script Date: 02/07/2008 04:36:26 p.m. ******/



/*
Inicia
Nombre:
sp_recupera_password


Funcionalidad:
Envia un correo electronico al usuario que ha olvidado su clave de acceso

Entradas:
Numero de empresa en la que se esta trabajando
Numero de empleado que olvido su clave
Tipo de usuario (interno o externo)

Proceso:
Obtiene el correo electronico de la persona con los datos proporcionados y envia un correo a esa direccion de
correo con la

Salida:
Correo electronico

Ejemplo:
exec sp_recupera_password 1,1,1
Termina
*/

CREATE procedure [dbo].sp_recupera_password
(
        @idempresa int,
        @numempleado int,
        @idtipo_usuario int
)
as
begin
        declare
        @password varchar(10),
        @email varchar(50),
        @origen varchar(50),
        @titulo varchar(30),
        @body varchar(4000)
```

```
	begin transaction trans
		/*
		Se revisa que exista un usuario con los datos proporcionados
		*/
		if exists(
			select numempleado
			from usuario
			where	idempresa = @idempresa and
				numempleado = @numempleado and
				idtipo_usuario = @idtipo_usuario
		)
		/*
		Si existe se obtiene su correo electronico y si clave de acceso
		*/
		begin
			set @email = (
				select
				email
				from usuario
				where	idempresa = @idempresa and
					numempleado = @numempleado and
					idtipo_usuario = @idtipo_usuario
			)

			set @password = (
				select
				[dbo].user_pwd(rtrim(clave_acceso),2)
				from usuario
				where	idempresa = @idempresa and
					numempleado = @numempleado and
					idtipo_usuario = @idtipo_usuario
			)

			set @origen	= 'Administracion'
			set @titulo	= 'Recuperacion de correo'
			set @body	= 'Su clave de acceso ha sido recuperada,
					  le sugerimos acceder el sistema y cambiar la clave cuanto antes
<br>
					  Clave : ' + @password

			/*
			Se envia un correo electronico
			*/
			exec [dbo].sp_send_cdosysmail @origen,@email,@titulo,@body

			select 'Se_le_ha_enviado_en_correo_electronico'
		end
		else
		/*
		Si no se encuentra un usuario con esos datos
		Se avisa al usuario del problema
		*/
		begin
			select
'No_se_encontro_registro_alguno_con_esos_datos,_no_se_envio_ningun_correo'
		end

	commit transaction trans

end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_registra_historial    Script Date: 02/07/2008 04:36:31 p.m. ******/



/*
Inicia
Nombre:
sp_registra_historial


Funcionalidad:
Registra la autorizacion o el rechazo en nombre de los verdaderos autorizadores (delegadores de autoridad)


Entradas:
Folio de la solicitud
Numero de empleado autorizador
Status que se autoriza
Observaciones acerca de la autorizacion o el rechzo

Proceso:
Se registra la autorizacion del autorizador. Se registra la autorizacion en nombre de quienes le delegaron la
autoridad

Salida:
Registros de autorizacion o rechazo

Ejemplo:
exec sp_registra_historial 1,3,'ea','observacion'
Termina
*/

CREATE procedure [dbo].sp_registra_historial
(
        @folio int,                     --Folio de la solicitud
        @numempleado int,               --Numero de empleado que autoriza
        @idstatus char(2),              --Status que se autoriza
        @observacion varchar(250)       --Observacion o comentario acerca de la autorizacion o el
rechazo
)
as
begin
        declare
```

```
	@idhistorial int,			--Id del historial
	@idempresa int,				--Numero de empresa en la que se trabaja
	@ultimo_id int,				--Id del ultimo rechazo de la solicitud
	@c_numempleado int,			--Cursor que almacenara el numero de los verdaderos
autorizadores
	@idstatus_rechazo char(2),		--Status de rechazo del status enviado
	@idstatus_a_comp char(2)		--Status que se usara para comparar que no se haya autorizado o
rechazado
						--bajo ese status

	/*
	Se asigna proximo idhistorial para el registro y se inserta el registro
	*/
	set @idhistorial = (select isnull(max(idhistorial),0) + 1 from historial_solicitud)
	insert into historial_solicitud
	(
	idhistorial,
	folio,
	numempleado,
	idstatus,
	fecha,
	observacion
	)
	values
	(
	@idhistorial,
	@folio,
	@numempleado,
	upper(@idstatus),
	getdate(),
	upper(@observacion)
	)

	/*
	Se obtiene el status de autorizacion, el status de rechazo y le empresa en la que se trabaja
	*/
	set @idstatus  = (select idstatus from status_rechazo where idstatus = @idstatus or idstatus_rechazo
= @idstatus)
	set @idstatus_rechazo  = (select idstatus_rechazo from status_rechazo where idstatus = @idstatus)
	set @idempresa = (select idempresa from solicitud where folio = @folio)

	if substring(@idstatus,1,1) = 'r'
	begin
		set @ultimo_id = 0
		set @idstatus = @idstatus
		set @idstatus_a_comp = @idstatus_rechazo
	end
	else
	begin
		set @ultimo_id = (select [dbo].fn_flujo_autoriza_obtiene_ultimo_id_rechazo(@folio))
		set @idstatus = @idstatus
		set @idstatus_a_comp = @idstatus
	end

	if @idstatus = 'nu'
	begin
		insert into historial_delegacion_autoridad_solicitud
		select  distinct @idhistorial, @numempleado
	end

	/*
	Se crea una tabla temporal en donde se almacenaran los numeros de empleado que han autorizado
```

```
	para ese status a partir del ultimo rechazo
	Esto se hace para que los autorizadores no se dupliquen en sus autorizaciones
	*/
	create table #temporal (autorizador int)
	insert into #temporal
	select hd.numempleado_delega
	from historial_solicitud as ha
	inner join historial_delegacion_autoridad_solicitud as hd on ha.idhistorial = hd.idhistorial
	where	ha.folio = @folio
		and ha.idstatus = @idstatus_a_comp
		and ha.idhistorial > @ultimo_id

	/*
	Se obtienen los autorizadores, es decir, el autorizador y las personas que le han
	delegado su autoridad
	*/
	declare cursor_2 cursor for
		select @numempleado
		union
		select numempleado_delega
		from transferencia_autoridad
		where	numempleado_delegado = @numempleado
			and idempresa = @idempresa
			and fecha_ini <= [dbo].fn_obtiene_fecha_hoy(getdate())
			and fecha_fin >= [dbo].fn_obtiene_fecha_hoy(getdate())
			and idstatus = 'a'


	open cursor_2


	fetch next from cursor_2
		into
			@c_numempleado

	while @@fetch_status = 0
		begin	/*
			Se verifica que las personas que tienen poder para autorizar la
			solicitud se encuentren en la tabla #temporal, es decir, que el autorizador o quienes
le
			delegaron su autoridad se encuentren entre los responsables
			Cuando esto se cumple, se registra en el historial, la persona que hizo la
autorizacion
			y en la tabla de historial_delegacion_autoridad... el nombre bajo el cual se hizo
			la autorizacion que puede ser el mismo que el del autorizador o de otra persona
			en caso de haber delegacion de autoridad
			*/
			if @idstatus = 'ep'
			begin
				insert into historial_delegacion_autoridad_solicitud
				select  distinct @idhistorial, @c_numempleado
				from distribucion_solicitud as di
				inner join centro_costo as ce on rtrim(di.ccostos) = rtrim(ce.idccostos)
				inner join centro_costo_autoriza as ca on rtrim(ce.idccostos) =
rtrim(ca.idccostos)
				where	di.folio = @folio  and
					di.idempresa = ce.idempresa and
					ce.idempresa = ca.idempresa and
					ca.numempleado = @c_numempleado and
					di.idempresa = @idempresa and
					ca.numempleado not in (select autorizador from #temporal)
			end
```

```
if @idstatus = 'ej'
begin
	insert into historial_delegacion_autoridad_solicitud
	select  distinct @idhistorial, us.num_jefe
	from solicitud as an
	inner join usuario as us on an.numempleado = us.numempleado
	where	an.idempresa = us.idempresa and
		an.folio = @folio and
		an.idempresa = @idempresa and
		us.num_jefe = @c_numempleado and
		us.num_jefe not in (select autorizador from #temporal)
end

if @idstatus = 'ea'
begin
	insert into historial_delegacion_autoridad_solicitud
	select  distinct @idhistorial, aa.numempleado
	from detalle_solicitud as de
	inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
	inner join concepto_gasto_area as ca on cg.idconcepto = ca.idconcepto
	inner join area_responsabilidad as ar on ca.idarea = ar.idarea
	inner join area_responsabilidad_autoriza as aa on ar.idarea = aa.idarea
	inner join usuario as us on aa.numempleado = us.numempleado
	where	de.idempresa = cg.idempresa and
		cg.idempresa = ca.idempresa and
		ca.idempresa = ar.idempresa and
		ar.idempresa = aa.idempresa and
		aa.idempresa = us.idempresa and
		cg.idstatus = 'a' and
		ar.idstatus = 'a' and
		us.idstatus = 'a' and
		aa.numempleado = @c_numempleado and
		de.folio = @folio and
		de.idempresa = @idempresa and
		aa.numempleado not in (select autorizador from #temporal)
	union
	select  distinct @idhistorial, ua.autorizador
	from solicitud as an
	inner join usuario as u1 on an.numempleado = u1.numempleado
	inner join usuario_area_autoriza as ua on u1.numempleado =
ua.numempleado
	inner join usuario as u2 on ua.autorizador = u2.numempleado
	where	an.idempresa = u1.idempresa and
		u1.idempresa = ua.idempresa and
		ua.idempresa = u2.idempresa and
		an.idempresa = @idempresa and
		u2.idstatus = 'a' and
		ua.autorizador = @c_numempleado and
		an.folio = @folio and
		ua.autorizador not in (select autorizador from #temporal)

end

if @idstatus = 'ed'
begin
	insert into historial_delegacion_autoridad_solicitud
	select  distinct @idhistorial, ua.autorizador
	from solicitud as an
	inner join usuario as u1 on an.numempleado = u1.numempleado
inner join usuario_direccion_autoriza as ua on u1.numempleado = ua.numempleado
	inner join usuario as u2 on ua.autorizador = u2.numempleado
```

```
                                        where   an.idempresa = u1.idempresa and
                                                u1.idempresa = ua.idempresa and
                                                ua.idempresa = u2.idempresa and
                                                an.idempresa = @idempresa and
                                                u2.idstatus = 'a' and
                                                ua.autorizador = @c_numempleado and
                                                an.folio = @folio and
                                                ua.autorizador not in (select autorizador from #temporal)
                                end

                                if @idstatus = 'ec'
                                begin
                                        insert into historial_delegacion_autoridad_solicitud
                                        select distinct @idhistorial, ca.numempleado
                                        from solicitud as an
                                        inner join contabilidad_autoriza as ca on an.idempresa = ca.idempresa
                                        where   an.folio = @folio and
                                                an.idempresa = @idempresa and
                                                ca.numempleado = @c_numempleado and
                                                ca.numempleado not in (select autorizador from #temporal)

                                end

                                fetch next from cursor_2
                                into
                                        @c_numempleado
                        end

                close           cursor_2
                deallocate      cursor_2

                /*
                Se elimina la tabla temporal
                */
                drop table #temporal

end


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_reporte_consulta_autorizadores    Script Date: 02/07/2008 04:36:28
p.m. ******/




/*
Inicia
```

Nombre:
sp_reporte_consulta_autorizadores

Funcionalidad:
Se obtienen los autorizadores pendientes de una solicitud y se les envia un correo electronico o se despliegan dependiendo de los parametros de entrada

Entradas:
Folio de la solicitud
Status del que se quieren encontrar los autorizadores
Accion a seguir
	p = autorizadores pendientes a partir del status dado
	i = autorizadores pendientes solo del status dado
Accion a seguir
	0 = no mandar email a autorizadores
	1 = mandar email a autorizadores
Accion a seguir
	0 = no mostrar los autorizadores
	1 = mostrar los autorizadores

Proceso:
Busca los autorizadores pendientes de la solicitud

Salida:
Registros de autorizacion o rechazo

Ejemplo:
exec sp_reporte_consulta_autorizadores 1,'nu','p',0,1
Termina
*/

```
CREATE procedure [dbo].sp_reporte_consulta_autorizadores
(
	@folio int,
	@idstatus char(2),
	@proviene char(1) = 'p', -- p = pendientes a partir del status dado
				 -- i = pendidentes solo del status dado

	@email tinyint = 0,	 -- 0 = no mandar email
				 -- 1 = si mandar email

	@mostrar tinyint = 0	 -- 0 = no mostrar los destinatarios del mail
				 -- 1 = mostrar los destinatarios del mail
)
as
begin
	declare
	@idtipo_operacion int,
	@idempresa int,
	@idflujo_a_partir int,
	@ultimo_id int,
	@solicitante int,

	@c_idflujo int,
	@c_idstatus char(2),
	@c_status varchar(50),
	@c_responsable int,
	@c_nombre varchar(50),

	@origen varchar(50),
```

```
	@destino varchar(50),
	@tipo_operacion varchar(50),
	@status varchar(250),
	@body varchar(4000)

	set @idtipo_operacion = (select idtipo_operacion from solicitud where folio = @folio)
	set @idempresa = (select idempresa from solicitud where folio = @folio)

	create table #temporal(
		idstatus char(2),
		idflujo int
	)

	/*
	Se inserta en la tabla temporal 1 el flujo a seguir de la solicitud
	*/
	insert into #temporal
	select
	idstatus,
	idflujo
	from flujo_autoriza
	where	idempresa = @idempresa and
		idtipo_operacion = @idtipo_operacion
	order by idflujo


	create table #temporal2(
		folio int,
		idflujo int,
		idstatus char(2),
		status varchar(50),
		responsable int,
		nombre varchar(50)
	)

	create table #temporal3(
		idflujo int,
		idstatus char(2),
		status varchar(50),
		responsable int,
		nombre varchar(50)
	)

	/*
	Se inserta en la tabla temporal 2 los autorizadores de todos los niveles
	*/
	insert into #temporal2
	select distinct
	@folio,
	te.idflujo,
	'ep' as idstatus,
	st.descripcion_1 as status,
	isnull(ca.numempleado,0) as responsable,
	rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as nombre
	from distribucion_solicitud as di
	inner join centro_costo as ce on rtrim(di.ccostos)  = rtrim(ce.idccostos)
	inner join centro_costo_autoriza as ca on rtrim(ce.idccostos) = rtrim(ca.idccostos)
	inner join #temporal as te on 'ep'  COLLATE DATABASE_DEFAULT = te.idstatus COLLATE
DATABASE_DEFAULT
	inner join usuario as us on ca.numempleado = us.numempleado
	inner join status as st on te.idstatus COLLATE DATABASE_DEFAULT  = st.idstatus COLLATE
DATABASE_DEFAULT
```

```
	where	di.folio = @folio and
		di.idempresa = ce.idempresa  and
		ce.idempresa  = ca.idempresa  and
		ca.idempresa  = us.idempresa

	union

	select distinct
	@folio,
	te.idflujo,
	'ej' as idstatus,
	st.descripcion_1 as status,
	isnull(us.num_jefe,0) as responsable,
	rtrim(u2.nombre) + ' ' + rtrim(u2.paterno) + ' ' + rtrim(u2.materno) as nombre
	from solicitud as an
	inner join usuario as us on an.numempleado = us.numempleado
	inner join #temporal as te on 'ej' COLLATE DATABASE_DEFAULT  = te.idstatus COLLATE
DATABASE_DEFAULT
	left outer join usuario as u2 on us.num_jefe = u2.numempleado
	inner join status as st on te.idstatus COLLATE DATABASE_DEFAULT  = st.idstatus COLLATE
DATABASE_DEFAULT
	where	an.idempresa  = us.idempresa  and
		us.idempresa  = u2.idempresa  and
		an.folio = @folio


	union

	select distinct
	@folio,
	te.idflujo,
	'ea' as idstatus,
	st.descripcion_1 as status,
	isnull(aa.numempleado,0) as responsable,
	rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as nombre
	from detalle_solicitud as de
	inner join concepto_gasto as cg on de.idconcepto = cg.idconcepto
	inner join concepto_gasto_area as ca on cg.idconcepto = ca.idconcepto
	inner join area_responsabilidad as ar on ca.idarea = ar.idarea
	inner join area_responsabilidad_autoriza as aa on ar.idarea = aa.idarea
	inner join usuario as us on aa.numempleado = us.numempleado
	inner join #temporal as te on 'ea' COLLATE DATABASE_DEFAULT  = te.idstatus COLLATE
DATABASE_DEFAULT
	inner join status as st on te.idstatus COLLATE DATABASE_DEFAULT  = st.idstatus COLLATE
DATABASE_DEFAULT
	where	de.idempresa  = cg.idempresa  and
		cg.idempresa   = ca.idempresa  and
		ca.idempresa   = ar.idempresa  and
		ar.idempresa  = aa.idempresa  and
		aa.idempresa   = us.idempresa  and
		cg.idstatus  COLLATE DATABASE_DEFAULT = 'a'  COLLATE DATABASE_DEFAULT and
		ar.idstatus  COLLATE DATABASE_DEFAULT = 'a'  COLLATE DATABASE_DEFAULT and
		us.idstatus  COLLATE DATABASE_DEFAULT = 'a' COLLATE DATABASE_DEFAULT  and
		de.folio = @folio

	union

	select distinct
	@folio,
	te.idflujo,
	'ea' as idstatus,
	st.descripcion_1 as status,
```

```
	isnull(ua.autorizador,0) as responsable,
	rtrim(u2.nombre) + ' ' + rtrim(u2.paterno) + ' ' + rtrim(u2.materno) as nombre
	from solicitud as an
	inner join usuario as u1 on an.numempleado = u1.numempleado
	inner join usuario_area_autoriza as ua on u1.numempleado = ua.numempleado
	inner join usuario as u2 on ua.autorizador = u2.numempleado
	inner join #temporal as te on 'ea'  COLLATE DATABASE_DEFAULT = te.idstatus COLLATE
DATABASE_DEFAULT
	inner join status as st on te.idstatus COLLATE DATABASE_DEFAULT  = st.idstatus COLLATE
DATABASE_DEFAULT
	where	an.idempresa  = u1.idempresa   and
		u1.idempresa   = ua.idempresa   and
		ua.idempresa   = u2.idempresa  and
		u2.idstatus  COLLATE DATABASE_DEFAULT = 'a'  COLLATE DATABASE_DEFAULT and
		an.folio = @folio

	union

	select distinct
	@folio,
	te.idflujo,
	'ed' as idstatus,
	st.descripcion_1 as status,
	isnull(ua.autorizador,0) as responsable,
	rtrim(u2.nombre) + ' ' + rtrim(u2.paterno) + ' ' + rtrim(u2.materno) as nombre
	from solicitud as an
	inner join usuario as u1 on an.numempleado = u1.numempleado
	inner join usuario_direccion_autoriza as ua on u1.numempleado = ua.numempleado
	inner join usuario as u2 on ua.autorizador = u2.numempleado
	inner join #temporal as te on 'ed'  COLLATE DATABASE_DEFAULT = te.idstatus COLLATE
DATABASE_DEFAULT
	inner join status as st on te.idstatus COLLATE DATABASE_DEFAULT  = st.idstatus COLLATE
DATABASE_DEFAULT
	where	an.idempresa  = u1.idempresa   and
		u1.idempresa   = ua.idempresa   and
		ua.idempresa   = u2.idempresa   and
		u2.idstatus  COLLATE DATABASE_DEFAULT = 'a'  COLLATE DATABASE_DEFAULT and
		an.folio = @folio

	union

	select distinct
	@folio,
	te.idflujo,
	'ec' as idstatus,
	st.descripcion_1 as status,
	isnull(ca.numempleado,0) as responsable,
	rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as nombre
	from solicitud as an
	inner join contabilidad_autoriza as ca on an.idempresa = ca.idempresa
	inner join #temporal as te on 'ec' COLLATE DATABASE_DEFAULT  = te.idstatus COLLATE
DATABASE_DEFAULT
	inner join usuario as us on ca.numempleado = us.numempleado
	inner join status as st on te.idstatus  COLLATE DATABASE_DEFAULT = st.idstatus COLLATE
DATABASE_DEFAULT
	where	an.folio = @folio and
		ca.idempresa    =  us.idempresa
	order by te.idflujo

	/*
	Si es status es diferente de vacio o no es rechazado o la solicitud no es nueva
	*/
```

```
	if @idstatus = '' or substring(@idstatus,1,1) = 'r' or substring(@idstatus,1,1) = 'n'
	begin
		/*
		Se inserta en la tabla temporal 3
		los autorizadores que no hayan autorizado a partir del ultimo rechazo de la solicitud
		*/
		insert into #temporal3
		select
		te.idflujo,
		te.idstatus,
		te.status,
		te.responsable,
		te.nombre
		from #temporal2 as te
		where te.responsable not in (
			select hd.numempleado_delega
			from historial_solicitud as ha
			inner join historial_delegacion_autoridad_solicitud as hd on ha.idhistorial =
hd.idhistorial
			where	ha.folio = @folio
				and ha.idstatus   COLLATE DATABASE_DEFAULT = te.idstatus
COLLATE DATABASE_DEFAULT
				and ha.idhistorial > @ultimo_id
		)

	end
	else
	/*
	Si el statu de la solicitud es vacio o rechazado o nuevo
	Se presentan todos los autorizadores
	*/
	begin
		set @idflujo_a_partir = (select min(idflujo) from #temporal2 where idstatus = @idstatus)
		set @ultimo_id = (select [dbo].fn_flujo_autoriza_obtiene_ultimo_id_rechazo(@folio))

		/*
		La letra p o i definen a partir de cuando tomar en cuenta el flujo
		p = autorizadores pendientes a partir del flujo dado
		i = solo los autorizadores de ese status
		*/
		if @proviene = 'p'
		begin
			insert into #temporal3
			select
			te.idflujo,
			te.idstatus,
			te.status,
			te.responsable,
			te.nombre
			from #temporal2 as te
			where te.responsable not in (
				select hd.numempleado_delega
				from historial_solicitud as ha
				inner join historial_delegacion_autoridad_solicitud as hd on ha.idhistorial =
hd.idhistorial
				where	ha.folio = @folio
					and ha.idstatus  COLLATE DATABASE_DEFAULT  = te.idstatus
COLLATE DATABASE_DEFAULT
					and ha.idhistorial > @ultimo_id
			) and
			idflujo >= @idflujo_a_partir
			/*
```

```
                        Aqui se define la diferencia de la p o la i
                        */
                end

                if @proviene = 'i'
                begin
                        insert into #temporal3
                        select
                        te.idflujo,
                        te.idstatus,
                        te.status,
                        te.responsable,
                        te.nombre
                        from #temporal2 as te
                        where te.responsable not in (
                                select hd.numempleado_delega
                                from historial_solicitud as ha
                                inner join historial_delegacion_autoridad_solicitud as hd on ha.idhistorial =
hd.idhistorial
                                where   ha.folio  = @folio
                                        and ha.idstatus  COLLATE DATABASE_DEFAULT  = te.idstatus
COLLATE DATABASE_DEFAULT
                                        and ha.idhistorial > @ultimo_id
                        ) and
                        idflujo = @idflujo_a_partir
                        /*
                        Aqui se define la diferencia de la p o la i
                        */
                end
        end

        /*
        Si se manda correo o no
        */
        if @email = 1
        begin
                declare cursor_1 cursor for
                        select * from #temporal3
                open cursor_1

                fetch next from cursor_1
                        into
                                @c_idflujo,
                                @c_idstatus,
                                @c_status,
                                @c_responsable,
                                @c_nombre

                while @@fetch_status = 0
                        begin
                                set @solicitante = (select numempleado from solicitud where folio =
@folio)
                                set @origen = (select [dbo].fn_obtiene_email(@idempresa,@solicitante))
                                set @destino = (select
[dbo].fn_obtiene_email(@idempresa,@c_responsable))
                                set @tipo_operacion = (select nombre from tipo_operacion where
idempresa = @idempresa and idtipo_operacion = @idtipo_operacion )
                                set @status = (
                                        select
                                        mensaje
                                        from mensaje_status
                                        where    idempresa = @idempresa and
```

```
                                        idstatus = @c_idstatus
                                )

                                if @status is null
                                begin
                                        set @status = 'No definido'
                                end

                                set @body =     'Solicitante : ' +
[dbo].fn_obtiene_nombre(@idempresa,@solicitante) + '<br>' +
                                        'Folio : ' + rtrim(cast(@folio as char)) + '<br>' +
                                        'Tipo de operacion : ' + @tipo_operacion  + '<br>' +
                                        'Mensaje : ' + rtrim(@status)

                                CREATE TABLE #tmp([id] int)
                                insert into #tmp
                                exec [dbo].sp_send_cdosysmail @origen,@destino,'Autorizaciones
Pendientes',@body
                                drop table #tmp

                                print 'Mail Enviado'
                                print 'De : ' + @origen
                                print 'A : ' + @destino
                                print 'Asunto : Autorizaciones Pendientes'
                                print 'Cuerpo : ' + @body

                        fetch next from cursor_1
                                into
                                        @c_idflujo,
                                        @c_idstatus,
                                        @c_status,
                                        @c_responsable,
                                        @c_nombre
                        end

                close           cursor_1
                deallocate      cursor_1

        end

        /*
        Si se muestran los datos o no
        */
        if @mostrar = 1
        begin
                select * from #temporal3
        end

        drop table #temporal
        drop table #temporal2
        drop table #temporal3

end

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
```

```
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_revisa_disponibilidad_ppto    Script Date: 02/07/2008 04:36:31 p.m. ******/




/*
Inicia
Nombre:
sp_revisa_disponibilidad_ppto



Funcionalidad:
Obtiene si para una solicitud hay ppto disponible

Entradas:
Folio de la solicitud
Variable en donde se guardara el resultado de la operacion
	1 = si hay ppto disponible
	0 = no hay ppto disponible

Proceso:
Obtiene los gastos de la solicitud y revisa la tabla de ppto para ver si hay ppto disponible para esa solicitud

Salida:
0 = no hay ppto disponible
1 = si hay ppto disponible

Ejemplo:
declare
@disponibilidad tinyint
exec [dbo].sp_revisa_disponibilidad_ppto 3, @disponibilidad output
print 'hay ppto : ' + rtrim(cast(@disponibilidad as char))

Termina
*/

CREATE procedure [dbo].sp_revisa_disponibilidad_ppto
(
	@folio int,
	@disponibilidad_ppto tinyint output
)
as
begin
	declare
	@idempresa int,
	@idmoneda int,

	@c_detalle int,
	@c_fecha_gasto datetime,
	@c_idconcepto int,
	@c_ccostos int,
	@c_porcentaje_distribucion float,
```

```
@c_tipo varchar(12),
@c_porcentaje_ded int,
@c_ccontable char(6),
@c_total money,

@ppto_presupuestado money,
@ppto_gastado money,
@ppto_comprometido money,
@ppto_disponible money,
@ppto_idmoneda int,

@total_detalle_a_moneda_ppto money

set @idempresa = (select idempresa from solicitud where folio = @folio)
set @idmoneda = (select idmoneda from solicitud where folio = @folio)

begin transaction valida_ppto

create table #temp(
	detalle int,
	fecha_gasto datetime,
	idconcepto int,
	ccostos int,
	porcentaje_distribucion float,
	tipo varchar(12),
	porcentaje_ded numeric,
	ccontable char(6),
	total money
)

insert into #temp
	exec [dbo].sp_ppto_obtiene_gastos @folio

declare cursor_ppto cursor for
	select * from #temp

open cursor_ppto

fetch next from cursor_ppto
	into
		@c_detalle,
		@c_fecha_gasto,
		@c_idconcepto,
		@c_ccostos,
		@c_porcentaje_distribucion,
		@c_tipo,
		@c_porcentaje_ded,
		@c_ccontable,
		@c_total

while @@fetch_status = 0
	begin
		/*
		print 'idempresa : ' + rtrim(cast(@idempresa as char))
		print 'ccontable : ' + rtrim(@c_ccontable)
		print 'anio : ' + rtrim(cast(year(@c_fecha_gasto) as char))
		print 'mes : ' + rtrim(cast(month(@c_fecha_gasto) as char))
		print 'idmoneda : ' + rtrim(cast(@idmoneda as char))
		print 'ccostos : ' + rtrim(cast(@c_ccostos as char))
		*/

		if @idempresa is null or rtrim(@idempresa) = ''
```

```
                        begin
                                --print 'La empresa no pueder ser vacio'
                                set @disponibilidad_ppto = 0
                                break
                        end

                        if @c_ccontable is null or rtrim(@c_ccontable) = ''
                        begin
                                --print 'La cuenta contable no pueder ser vacio'
                                set @disponibilidad_ppto = 0
                                break
                        end

                        if exists(
                                select *
                                from presupuesto
                                where   idempresa = @idempresa and
                                        rtrim(ccontable) = @c_ccontable and
                                        anio = year(@c_fecha_gasto) and
                                        mes = month(@c_fecha_gasto) and
                                        identidad = @c_ccostos
                        )
                        begin
                                set @ppto_idmoneda = (select
[dbo].fn_obtiene_ppto_idmoneda(@idempresa,@c_ccontable,@c_fecha_gasto,@c_ccostos))
                                set @ppto_presupuestado = (select
[dbo].fn_obtiene_ppto_presupuestado(@idempresa,@c_ccontable,@c_fecha_gasto,@c_ccostos))
                                set @ppto_gastado = (select
[dbo].fn_obtiene_ppto_gastado(@idempresa,@c_ccontable,@c_fecha_gasto,@c_ccostos))
                                set @ppto_comprometido = (select
[dbo].fn_obtiene_ppto_comprometido(@idempresa,@c_ccontable,@c_fecha_gasto,@c_ccostos))
                                set @ppto_disponible = @ppto_presupuestado - (@ppto_gastado +
@ppto_comprometido)
                                set @total_detalle_a_moneda_ppto = (select
[dbo].fn_obtiene_tipo_cambio(@idmoneda,@ppto_idmoneda,@c_fecha_gasto,@c_total))
                        end
                        else
                        begin
                                set @ppto_idmoneda = @idmoneda
                                set @ppto_presupuestado = 0
                                set @ppto_gastado = 0
                                set @ppto_comprometido = @c_total
                                set @ppto_disponible = 0
                                set @total_detalle_a_moneda_ppto = @c_total
                                set @disponibilidad_ppto = 0
                                break
                        end
                        /*
                        print 'Moneda del presupuesto : '    + rtrim(cast(@ppto_idmoneda as char))
                        print 'Presupuestado : '             + rtrim(cast(@ppto_presupuestado as char))
                        print 'Gastado : '                   + rtrim(cast(@ppto_gastado as char))
                        print 'Comprometido : '              + rtrim(cast(@ppto_comprometido as char))
                        print 'Disponible : '                        + rtrim(cast(@ppto_disponible as char))

                        print 'total de detalle : ' + rtrim(cast(@c_total as char))
                        print 'total detalle convertido a moneda : ' +
rtrim(cast(@total_detalle_a_moneda_ppto as char))
                        */

                        if @total_detalle_a_moneda_ppto <= @ppto_disponible
                        begin
                                --print 'si hay ppto'
```

```
                        set @disponibilidad_ppto = 1
                end

                if @total_detalle_a_moneda_ppto > @ppto_disponible
                begin
                        --print 'no hacer nada'
                        set @disponibilidad_ppto = 0
                        break
                end



                --print '---------------------------------'

                fetch next from cursor_ppto
                into
                        @c_detalle,
                        @c_fecha_gasto,
                        @c_idconcepto,
                        @c_ccostos,
                        @c_porcentaje_distribucion,
                        @c_tipo,
                        @c_porcentaje_ded,
                        @c_ccontable,
                        @c_total
        end

        close           cursor_ppto
        deallocate      cursor_ppto

        drop table #temp

        commit transaction valida_ppto

        return(@disponibilidad_ppto)

end


SET QUOTED_IDENTIFIER OFF


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_revisa_distribucion_por_folio    Script Date: 02/07/2008 04:36:31 p.m.
******/
```

```
/*
Inicia
Nombre:
sp_revisa_distribucion_por_folio



Funcionalidad:
Se evalua si un folio esta distribuido al 100% entre los centros de costo

Entradas:
Folio de la solicitud

Proceso:
Evaluacion de distribucion de una solicitud

Salida:
Caracter de validez
        (0 = no esta correctamente distribuido)
        (1 = esta correctamente distribuido)

Ejemplo:
exec sp_revisa_distribucion_por_folio 1
Termina
*/

CREATE procedure [dbo].sp_revisa_distribucion_por_folio
(
        @folio int
)
as
begin
        declare
        @distribucion_valida tinyint

        /*
        Aunque sean muchos detalles dentro de una solicitud, por el hecho de estar ordenados por detalle,
        si un detalle no esta distribuido correctamente el resultado sera como sigue:

        vale    vale    vale
        ----    ----    ----
        0       0       1
        1       0       1
        1       1       1

        De esta manera cuando el primer registro es 1, la distribucion es valida, cuando es 0
        la distribucion no es valida
        */
        set @distribucion_valida = (
                select top 1
                case
                        when sum(isnull(di.porcentaje,0)) <= 0      then 0
                        when sum(isnull(di.porcentaje,0)) < 100     then 0
                        when sum(isnull(di.porcentaje,0)) = 100     then 1
                end as vale
                from solicitud as so
                left outer join detalle_solicitud as de on so.folio = de.folio
                left outer join distribucion_solicitud as di on de.detalle = di.detalle
                where   so.folio = @folio
                group by di.detalle
```

```
                order by vale asc
        )

        select @distribucion_valida as distribucion_valida

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_revisa_operaciones_en_proceso    Script Date: 02/07/2008 04:36:28 p.m. ******/




/*
Inicia
Nombre:
sp_revisa_operaciones_en_proceso


Funcionalidad:
Obtiene los siguientes datos:
Cuantas solicitudes puede tener el usuario pendientes en proceso antes de pedir una nueva
Cuantas tiene pendientes
Cuantas tiene en proceso
Cuantas tiene rechazadas
Cuantas puede pedir

La operacion es una formula matematica como la que sigue: Posibilidad = politica - (pendientes + proceso + rechazadas)

Entradas:
Empresa en la que se trabaja
Numero de empleado que se firmo a la aplicacion
Tipo de operacion que quiere solicitar

Proceso:
Saber si el usuario puede hacer una solicitud

Salida:
Numeros de cuantas solicitudes tiene el usuario en cada rubro

Ejemplo:
```

```
exec [dbo].sp_revisa_operaciones_en_proceso 13,123,'anticipo'
Termina
*/

CREATE procedure [dbo].sp_revisa_operaciones_en_proceso
(
	@idempresa int,
	@numempleado int,
	@tipo_operacion varchar(50)
)
as
begin
	declare
	@idperfil int,
	@idtipo_operacion int,
	@operaciones_politica int,
	@operaciones_pendientes int,
	@operaciones_rechazadas int,
	@operaciones_proceso int,
	@restante int

	set @idperfil = (select idperfil from usuario where idempresa = @idempresa and numempleado =
@numempleado)
	set @idtipo_operacion = (select [dbo].fn_obtiene_tipo_operacion(@idempresa,@tipo_operacion))
	set @operaciones_politica = (
		select isnull(op_proceso,0)
		from operacion_proceso
		where	idempresa = @idempresa and
			idperfil = @idperfil and
			idtipo_operacion = @idtipo_operacion
	)
	set @operaciones_pendientes = (
		select count(folio)
		from solicitud
		where	substring(idstatus,1,1) = 'n' and
			idempresa = @idempresa and
			numempleado = @numempleado and
			idtipo_operacion = @idtipo_operacion
	)
	set @operaciones_rechazadas = (
		select count(folio)
		from solicitud
		where	substring(idstatus,1,1) = 'r' and
			idempresa = @idempresa and
			numempleado = @numempleado and
			idtipo_operacion = @idtipo_operacion
	)
	set @operaciones_proceso = (
		select count(folio)
		from solicitud
		where	substring(idstatus,1,1) = 'e' and
			idempresa = @idempresa and
			numempleado = @numempleado and
			idtipo_operacion = @idtipo_operacion
	)
	set @restante = @operaciones_politica - (@operaciones_pendientes + @operaciones_rechazadas +
@operaciones_proceso)

	select	@operaciones_politica as politica,
		@operaciones_pendientes as pendientes,
		@operaciones_rechazadas as rechazadas,
		@operaciones_proceso as proceso,
```

```
                @restante as restante

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_rtrn_sequential    Script Date: 02/07/2008 04:36:23 p.m. ******/




/*
EXEC sp_rtrn_sequential 'carga'
*/

CREATE PROC sp_rtrn_sequential
                @object_name varchar(50)='dummy_object',
                @last_key_assigned int = 0 OUTPUT,
                @locked_status_flag tinyint = 0 OUTPUT,
                @option char(1) = 'v'
AS

        DECLARE
                @last_locked_by varchar(50),
                @last_locked_at datetime

        SELECT
        @last_key_assigned=last_key_assigned,
        @locked_status_flag=locked_status_flag,
        @last_locked_by=last_locked_by,
        @last_locked_at=last_locked_at
        FROM [dbo].[object_control]
        WHERE           LTRIM([object_name]) = @OBJECT_NAME

        IF @option = 'v'
        BEGIN
                SELECT
                @last_key_assigned AS last_key_assigned,
                @locked_status_flag AS locked_status_flag,
                @last_locked_by AS last_locked_by,
                @last_locked_at AS last_locked_at
        END
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_send_cdosysmail    Script Date: 02/07/2008 04:36:23 p.m. ******/




/*
Inicia
Nombre:
sp_send_cdosysmail



Funcionalidad:
Envia correos electronicos a partir de la libreria CDOSYS contenida por defecto en Windows 2000 Server y en
Windows 2003 Server

Entradas:
Origen
Destino
Titulo del correo
Cuerpo del correo
Direccion IP del servidor Web de donde se enviaran los correos

Proceso:
Envio del correo

Salida:
Correo enviado o codigo de erros

Ejemplo:
exec [dbo].sp_send_cdosysmail 'origen@origen.com','destino@destino.com','Titulo','Cuerpo del correo'
Termina
*/

CREATE PROCEDURE [dbo].[sp_send_cdosysmail]
            @From varchar(100) ,
            @To varchar(100) ,
            @Subject varchar(100)='',
            @Body varchar(4000)='',
            @smtpserver varchar(15) = '192.168.1.105'
          /************************************************************************
          This stored procedure takes the parameters and sends an e-mail.
```

```
        All the mail configurations are hard-coded in the stored procedure.
        Comments are added to the stored procedure where necessary.
        References to the CDOSYS objects are at the following MSDN Web site:
        http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cdosys/html/_cdosys_messaging.asp

        **********************************************************************/
          AS
          Declare @iMsg int
          Declare @hr int
          Declare @source varchar(255)
          Declare @description varchar(500)
          Declare @output varchar(1000)


        --************* Create the CDO.Message Object ************************
          EXEC @hr = sp_OACreate 'CDO.Message', @iMsg OUT
          IF @hr <>0
            BEGIN
              SELECT @hr
              INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OACreate')
              EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
              IF @hr = 0
                BEGIN
                  SELECT @output = '  Source: ' + @source
                  PRINT  @output
                  SELECT @output = '  Description: ' + @description
                  PRINT  @output
INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To, @Subject, @Body,
@iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OACreate')
RETURN
                END
              ELSE
                BEGIN
                  PRINT '  sp_OAGetErrorInfo failed.'
                  RETURN
                END
            END

        --***************Configuring the Message Object ******************
        -- This is to configure a remote SMTP server.
        -- http://msdn.microsoft.com/library/default.asp?url=/library/en-
us/cdosys/html/_cdosys_schema_configuration_sendusing.asp
          EXEC @hr = sp_OASetProperty @iMsg,
'Configuration.fields("http://schemas.microsoft.com/cdo/configuration/sendusing").Value','2'
          IF @hr <>0
            BEGIN
              SELECT @hr
              INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OASetProperty sendusing')
              EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
              IF @hr = 0
                BEGIN
                  SELECT @output = '  Source: ' + @source
                  PRINT  @output
                  SELECT @output = '  Description: ' + @description
                  PRINT  @output
INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To, @Subject, @Body,
@iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OASetProperty sendusing')
GOTO send_cdosysmail_cleanup
                END
              ELSE
```

```
                BEGIN
                  PRINT '  sp_OAGetErrorInfo failed.'
                  GOTO send_cdosysmail_cleanup
                END
              END
          -- This is to configure the Server Name or IP address.
          -- Replace MailServerName by the name or IP of your SMTP Server.
            EXEC @hr = sp_OASetProperty @iMsg,
'Configuration.fields("http://schemas.microsoft.com/cdo/configuration/smtpserver").Value', @smtpserver
            IF @hr <>0
              BEGIN
                SELECT @hr
                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OASetProperty smtpserver')
                EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
                IF @hr = 0
                  BEGIN
                    SELECT @output = '  Source: ' + @source

            PRINT  @output

                    SELECT @output = '  Description: ' + @description
                    PRINT  @output
                        INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OASetProperty
smtpserver')
GOTO send_cdosysmail_cleanup
                  END
                ELSE
                  BEGIN
                    PRINT '  sp_OAGetErrorInfo failed.'
                    GOTO send_cdosysmail_cleanup
                  END
              END

          -- Save the configurations to the message object.
            EXEC @hr = sp_OAMethod @iMsg, 'Configuration.Fields.Update', null
            IF @hr <>0
              BEGIN
                SELECT @hr
                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OASetProperty Update')
                EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
                IF @hr = 0
                  BEGIN
                    SELECT @output = '  Source: ' + @source
                    PRINT  @output
                    SELECT @output = '  Description: ' + @description
                    PRINT  @output
                        INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OASetProperty
Update')
                        GOTO send_cdosysmail_cleanup
                  END
                ELSE
                  BEGIN
                    PRINT '  sp_OAGetErrorInfo failed.'
                    GOTO send_cdosysmail_cleanup
                  END
              END

          -- Set the e-mail parameters.
```

```
            EXEC @hr = sp_OASetProperty @iMsg, 'To', @To
            IF @hr <>0
              BEGIN
                SELECT @hr
                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OASetProperty To')
                EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
                IF @hr = 0
                  BEGIN
                    SELECT @output = '  Source: ' + @source
                    PRINT  @output
                    SELECT @output = '  Description: ' + @description
                    PRINT  @output
                        INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OASetProperty
To')
GOTO send_cdosysmail_cleanup
                  END
                ELSE
                  BEGIN
                    PRINT '  sp_OAGetErrorInfo failed.'
                    GOTO send_cdosysmail_cleanup
                  END
              END

            EXEC @hr = sp_OASetProperty @iMsg, 'From', @From
            IF @hr <>0
              BEGIN
                SELECT @hr
                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OASetProperty From')
                EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
                IF @hr = 0
                  BEGIN
                    SELECT @output = '  Source: ' + @source
                    PRINT  @output
                    SELECT @output = '  Description: ' + @description
                    PRINT  @output
                        INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OASetProperty
From')
GOTO send_cdosysmail_cleanup
                  END
                ELSE
                  BEGIN
                    PRINT '  sp_OAGetErrorInfo failed.'
                    GOTO send_cdosysmail_cleanup
                  END
              END

            EXEC @hr = sp_OASetProperty @iMsg, 'Subject', @Subject
            IF @hr <>0
              BEGIN
                SELECT @hr
                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OASetProperty Subject')
                EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
                IF @hr = 0
                  BEGIN
                    SELECT @output = '  Source: ' + @source
                    PRINT  @output
                    SELECT @output = '  Description: ' + @description
```

```
                PRINT  @output
                    INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OASetProperty
Subject')
GOTO send_cdosysmail_cleanup
                END
              ELSE
                BEGIN
                  PRINT '  sp_OAGetErrorInfo failed.'
                  GOTO send_cdosysmail_cleanup
                END
            END

          -- If you are using HTML e-mail, use 'HTMLBody' instead of 'TextBody'.
            EXEC @hr = sp_OASetProperty @iMsg, 'HTMLBody', @Body
            IF @hr <>0
              BEGIN
                SELECT @hr
                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OASetProperty TextBody')
                EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
                IF @hr = 0
                  BEGIN
                    SELECT @output = '  Source: ' + @source
                    PRINT  @output
                    SELECT @output = '  Description: ' + @description
                    PRINT  @output
                        INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OASetProperty
TextBody')
GOTO send_cdosysmail_cleanup
                END
              ELSE
                BEGIN
                  PRINT '  sp_OAGetErrorInfo failed.'
                  GOTO send_cdosysmail_cleanup
                END
            END

            EXEC @hr = sp_OAMethod @iMsg, 'Send', NULL

            IF @hr <>0
              BEGIN
                SELECT @hr
                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OAMethod Send')

                EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT
                IF @hr = 0
                  BEGIN
                    SELECT @output = '  Source: ' + @source
                    PRINT  @output
                    SELECT @output = '  Description: ' + @description
                    PRINT  @output
                        INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for sp_OAMethod
Send')

GOTO send_cdosysmail_cleanup
                END
              ELSE
                BEGIN
```

```
                PRINT '  sp_OAGetErrorInfo failed.'
                GOTO send_cdosysmail_cleanup
              END
            END



        -- Do some error handling after each step if you have to.
        -- Clean up the objects created.
send_cdosysmail_cleanup:
        If (@iMsg IS NOT NULL) -- if @iMsg is NOT NULL then destroy it
        BEGIN
                EXEC @hr=sp_OADestroy @iMsg

                -- handle the failure of the destroy if needed
                IF @hr <>0
                BEGIN
                        select @hr
            INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, 'Failed at sp_OADestroy')
                        EXEC @hr = sp_OAGetErrorInfo NULL, @source OUT, @description OUT

                        -- if sp_OAGetErrorInfo was successful, print errors
                        IF @hr = 0
                        BEGIN
                                SELECT @output = '  Source: ' + @source
                            PRINT  @output
                            SELECT @output = '  Description: ' + @description
                            PRINT  @output
                                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid,
@From, @To, @Subject, @Body, @iMsg, @hr, @source, @description, @output, 'sp_OAGetErrorInfo for
sp_OADestroy')
                        END

                        -- else sp_OAGetErrorInfo failed
                        ELSE
                        BEGIN
                                PRINT '  sp_OAGetErrorInfo failed.'
                            RETURN
                        END
                END
        END
        ELSE
        BEGIN
                PRINT ' sp_OADestroy skipped because @iMsg is NULL.'
                INSERT INTO [dbo].[cdosysmail_failures] VALUES (getdate(), @@spid, @From, @To,
@Subject, @Body, @iMsg, @hr, @source, @description, @output, '@iMsg is NULL, sp_OADestroy skipped')
            RETURN
        END


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Stored Procedure dbo.sp_solicitud_inserta_modifica    Script Date: 02/07/2008 04:36:28 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_solicitud_inserta_modifica


Funcionalidad:
Inserta, modifica o elimina una solicitud

Entradas:
Folio de la solicitud
Empresa en la que se esta trabajando
Numero de empleado que hace la solicitud
Tipo de operacion
Status de la solicitud
Moneda de la solicitud
Moneda en la que se pretende se haga el pago
Territorio del gasto de la solicitud
Descripcion
Opcion de accion
        (i = inserta)
        (d = elimina)
Variable commit

Proceso:
Dependiendo de la variable @proviene, el procedimiento, modifica o elimina

Salida:
Operacion de insercion, actualizacion o eliminacion

Ejemplo:
exec sp_solicitud_inserta_modifica 0,12,123,'anticipo','nu',1,1,1,'Descripcion','i'
Termina
*/


CREATE procedure [dbo].sp_solicitud_inserta_modifica
(
        @id int,
        @idempresa int,
        @numempleado int,
        @tipo_operacion varchar(50),
        @idstatus char(2),
        @idmoneda int,
        @idmoneda_pago int,
        @idterritorio int,
        @descripcion varchar(250),
        @proviene char(1),

        @commit tinyint = 1
```

```
)
as
begin
	declare
	@idtipo_operacion int,
	@folio int

	set @idtipo_operacion = (select
[dbo].fn_obtiene_tipo_operacion(@idempresa,rtrim(@tipo_operacion)))


	begin transaction trans


	if @proviene = 'i'
	begin
		set @folio = (select [dbo].fn_solicitud_obtiene_folio_consecutivo())

		set dateformat dmy insert into solicitud
		(
		folio,
		idtipo_operacion,
		idempresa,
		numempleado,
		idstatus,
		idmoneda,
		idmoneda_pago,
		idterritorio,
		descripcion,
		fecha_solicitud
		)
		values(
		@folio,
		@idtipo_operacion,
		@idempresa,
		@numempleado,
		@idstatus,
		@idmoneda,
		@idmoneda_pago,
		@idterritorio,
		upper(left(@descripcion,250)),
		getdate()
		)

		select @folio as folio
	end


	if @proviene = 'u'
	begin
		set dateformat dmy
		update solicitud
		set
		idmoneda = @idmoneda,
		idmoneda_pago = @idmoneda_pago,
		idterritorio = @idterritorio,
		fecha_solicitud = getdate(),
		descripcion = upper(left(@descripcion,250))
		where folio = @id
	end

	if @proviene = 'd'
```

```
		begin
			delete from solicitud where folio = @id
		end



		if @commit = 1
		begin
			commit transaction trans
			print 'Transaction commited'

		end

		if @commit <> 1
		begin
			rollback transaction trans
			print 'Transaction rolled-back'
		end



end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_territorio_catalogo    Script Date: 02/07/2008 04:36:23 p.m. ******/



/*
Inicia
Nombre:
sp_territorio_catalogo



Funcionalidad:
Obtiene los territorios vigentes por empresa

Entradas:
No tiene entradas

Proceso:
Obtencion de los territorios vigentes
```

```
Salida:
Territorios

Ejemplo:
exec sp_territorio_catalogo
Termina
*/

CREATE procedure [dbo].sp_territorio_catalogo
as
begin
	select
	idterritorio,
	descripcion
	from territorio
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_tipo_cambio_catalogo    Script Date: 02/07/2008 04:36:26 p.m. ******/



/*
Inicia
Nombre:
sp_tipo_cambio_catalogo


Funcionalidad:
Presenta el catalogo de tipos de cambio

Entradas:
Fecha en la que se quiere localizar tipos de cambio (opcional)

Proceso:
Busca el tipo de cambio de la fecha proporcionada

Salida:
Tipos de cambio

Ejemplo:
Termina
```

```
*/

CREATE procedure [dbo].sp_tipo_cambio_catalogo
(
	@fecha_tipo_cambio char(11)
)
as
begin
	declare
	@fecha char(10)

	if @fecha_tipo_cambio <> '0'
	begin
		set @fecha = (select [dbo].sp_convierte_texto_a_fecha(@fecha_tipo_cambio))

		select
		mn1.descripcion as moneda_origen,
		mn2.descripcion as moneda_destino,
		convert(char,tc.fecha,103) as fecha_tipo_cambio,
		isnull(tc.tipo_cambio,0) as tipo_cambio
		from tipo_cambio as tc
		inner join moneda as mn1 on tc.idmoneda_origen = mn1.idmoneda
		inner join moneda as mn2 on tc.idmoneda_destino = mn2.idmoneda
		where	convert(char,tc.fecha,103) = @fecha
		order by tc.fecha desc, mn1.descripcion
	end

	if @fecha_tipo_cambio = '0'
	begin
		select top 30
		mn1.descripcion as moneda_origen,
		mn2.descripcion as moneda_destino,
		convert(char,tc.fecha,103) as fecha_tipo_cambio,
		isnull(tc.tipo_cambio,0) as tipo_cambio
		from tipo_cambio as tc
		inner join moneda as mn1 on tc.idmoneda_origen = mn1.idmoneda
		inner join moneda as mn2 on tc.idmoneda_destino = mn2.idmoneda
		order by tc.fecha desc, mn1.descripcion
	end


end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_tipo_concepto_catalogo    Script Date: 02/07/2008 04:36:23 p.m.
******/
```

```
/*
Inicia
Nombre:
sp_tipo_concepto_catalogo



Nota:
Este procedimiento se obtiene el catalogo de tipos de concepto por empresa

Funcionalidad:
Obtiene los tipos de concepto por empresa

Entradas:
Numero de empresa en la que se esta trabajando
Accion a seguir
        u = presenta los datos relevantes para los usuarios
        a = presenta los datos relevantes para los administradores

Proceso:
Obtencion de los tipos de concepto por empresa

Salida:
Tipos de concepto

Ejemplo:
exec sp_tipo_concepto_catalogo 1,'a'
Termina
*/

CREATE procedure [dbo].sp_tipo_concepto_catalogo
(
        @idempresa int,
        @proviene char(1)
)
as
begin
        if @proviene = 'u'
        begin
                select
                idtipo_concepto,
                descripcion
                from tipo_concepto
                where    idempresa = @idempresa
                order by descripcion asc
        end

        if @proviene = 'a'
        begin
                select
                idtipo_concepto,
                descripcion,
                isnull(boleto_avion,0) as boleto_avion,
                isnull(kilometraje,0) as kilometraje,
                isnull(precio_unitario_cantidad,0) as precio_unitario_cantidad,
                isnull(iva,0) as iva,
                isnull(tua,0) as tua,
```

```
                isnull(propina,0) as propina,
                isnull(dias,0) as dias,
                isnull(comensales,0) as comensales,
                isnull(deposito,0) as deposito
                from tipo_concepto
                where    idempresa = @idempresa
                order by descripcion asc
        end

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_muestra_datos_capturados_ver_detalle    Script Date: 02/07/2008
04:36:29 p.m. ******/



/****** Object:  Stored Procedure dbo.sp_tipo_concepto_define    Script Date: 02/07/2008 04:36:26 p.m. ******/




/*
Inicia
Nombre:
sp_tipo_concepto_define



Funcionalidad:
Obtiene el las caracteristicas que se le pediran al usuario cuando seleccione un concepto de gasto. Ademas
obtiene los valores correspondientes a esas caracteristica en caso de existir un detalle capturado para ese
concepto de gast

Entradas:
Numero de empresa en la que se esta trabajando
Numero de empleado que hace la solicitud
Concepto de gasto
Moneda de la solicitud
Territorio de la solicitud
Tipo de operacion (anticipo,pago,compra,etc)
Numero de detalle de la solicitud

Proceso:
Obtiene las caracteristicas que el usuario debera introducir y los valores correspondientes en caso de que se
trate de un detalle capturado previamente
```

Salida:
Caracteristicas del concepto de gasto y valores capturados para el mismo

Ejemplo:
exec [dbo].sp_tipo_concepto_define 3,3,1,6,1,'anticipo',1
Termina
*/

```
CREATE procedure [dbo].sp_tipo_concepto_define
(
	@idempresa int,
	@numempleado int,
	@idconcepto int,
	@idmoneda int,
	@idterritorio int,
	@tipo_operacion varchar(50),
	@detalle int
)
as
begin
	declare
	@idnivel int,
	@idtarjeta_corporativa tinyint,
	@monto_limite money,
	@idtipo_operacion int

	/*
	Se obtiene el nivel del usuario
	*/
	set @idnivel = (select idnivel from usuario where idempresa = @idempresa and numempleado =
@numempleado)
	/*
	Se obtiene si el usuario tiene tarjeta corporativa
	*/
	set @idtarjeta_corporativa = (0)
	/*
	Se obtiene el tipo de operacion
	*/
	set @idtipo_operacion =(select
[dbo].fn_obtiene_tipo_operacion(@idempresa,rtrim(@tipo_operacion)))
	/*
	Se obtiene el monto limite al que tiene permitido el usuario dependiendo de su nivel, la moneda, el
	territorio de gasto, el concepto de gasto seleccionado y si tiene o no tarjeta corporativa
	*/
	set @monto_limite = (select
[dbo].fn_obtiene_monto_limite(@idnivel,@idconcepto,@idempresa,@idmoneda,@idterritorio,@idtipo_operaci
on))


	select top 1

	tc.boleto_avion as ask_boleto_avion,
	[dbo].fn_detalle_obtiene_origen_boleto_avion(@detalle) as origen,
	[dbo].fn_detalle_obtiene_destino_boleto_avion(@detalle) as destino,
	[dbo].fn_detalle_obtiene_aerolinea_boleto_avion(@detalle) as aerolinea,
	[dbo].fn_detalle_obtiene_hora_salida_boleto_avion(@detalle) as hora_salida,
	/*
	Datos relativos al kilometraje:
	El concepto exige kilometraje?
	Costo por kilometro
```

```
Identificador del origen/destino
Numero de viajes
*/
tc.kilometraje as ask_kilometraje,
[dbo].fn_detalle_obtiene_total_kilometros (@idempresa,@detalle) as total_kilometros,
[dbo].fn_obtiene_costo_por_kilometro(@idempresa,@idmoneda) as costo_por_kilometro,
[dbo].fn_detalle_obtiene_idkm_por_detalle (@idempresa,@detalle) as idkm,
[dbo].fn_detalle_obtiene_no_viajes_por_detalle(@idempresa,@detalle) as no_viajes,

/*
Datos relativos al precio unitario y cantidad de un articulo:
El concepto exige precio y cantidad?
Precio unitario
Numero de unidades
*/
tc.precio_unitario_cantidad as ask_precio_unitario_cantidad,
[dbo].fn_detalle_obtiene_precio_unitario (@detalle) as precio_unitario,
[dbo].fn_detalle_obtiene_unidades(@detalle) as unidades,

/*
Subtotal
*/
[dbo].fn_obtiene_subtotal_por_detalle(@detalle) as subtotal,

/*
Datos relativos al IVA:
El concepto exige IVA?
IVA
Clave del IVA utilizado
*/
tc.iva as ask_iva,
[dbo].fn_obtiene_iva_por_detalle(@detalle) as iva,
[dbo].fn_obtiene_iva_valor_por_detalle(@detalle) as idiva,

/*
Datos relativos a la propina:
El concepto exige propina?
Propina
*/
tc.propina as ask_propina,
[dbo].fn_obtiene_propina_por_detalle(@detalle) as propina,

/*
Datos relativos al TUA:
El concepto exige TUA?
TUA
*/
tc.tua as ask_tua,
[dbo].fn_obtiene_tua_por_detalle(@detalle) as tua,

/*
Total
*/
[dbo].fn_obtiene_total_por_detalle(@detalle) as total,

/*
Datos relativos a los dias:
El concepto exige especificar dias?
Numero de dias
*/
tc.dias as ask_dias,
[dbo].fn_detalle_obtiene_dias(@detalle) as dias,
```

```
		/*
		Datos relativos a los comensales:
		El concepto exige especificar comensales?
		Numero de comensales
		*/
		tc.comensales as ask_comensales,
		[dbo].fn_detalle_obtiene_comensales(@detalle) as comensales,

		/*
		Datos relativos a los depositos:
		El concepto exige asociar un deposito?
		Numero de comensales
		*/
		tc.deposito as ask_deposito,
		[dbo].fn_detalle_obtiene_folio_deposito(@detalle) as folio_deposito,

		/*
		Datos relativos a las politicas corporativas:
		Subtotal diario
		*/
		round((
		[dbo].fn_obtiene_subtotal_por_detalle(@detalle) /
		[dbo].fn_detalle_obtiene_dias(@detalle) /
		[dbo].fn_detalle_obtiene_comensales(@detalle)
		),2) as subtotal_diario,

		/*
		Datos relativos a las politicas corporativas:
		Monto limite permitido
		*/
		@monto_limite as monto_limite

		from concepto_gasto as cg
		inner join concepto_gasto_tipo_concepto as ct on cg.idconcepto = ct.idconcepto
		inner join tipo_operacion as tp on ct.idtipo_operacion = tp.idtipo_operacion
		inner join tipo_concepto as tc on ct.idtipo_concepto = tc.idtipo_concepto
		where	cg.idempresa = ct.idempresa and
			ct.idempresa = tc.idempresa and
			tc.idempresa = tp.idempresa and
			cg.idempresa = @idempresa and
			cg.idstatus = 'a' and
			tp.idtipo_operacion = @idtipo_operacion and
			tp.idstatus = 'a' and
			cg.idconcepto = @idconcepto
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

/****** Object: Stored Procedure dbo.sp_tipo_concepto_inserta_modifica   Script Date: 02/07/2008 04:36:23 p.m. ******/

/*
Inicia
Nombre:
sp_tipo_concepto_inserta_modifica

Funcionalidad:
Inserta o modifica tipos de concepto

Entradas:
Identificador del tipo de concepto, en caso de que se trate de una modificacion o eliminacion
Empresa en la que se esta trabajando
Nombre del tipo de concepto
Se exige boleto de avion
	1 = si
	0 = no
Se exige kilometraje
	1 = si
	0 = no
Se exige precio unitario y cantidad
	1 = si
	0 = no
Se exige IVA
	1 = si
	0 = no
Se exige TUA
	1 = si
	0 = no
Se exige propina
	1 = si
	0 = no
Se exige dias
	1 = si
	0 = no
Se exige comensales
	1 = si
	0 = no
Se exige deposito a favor de la empresa
	1 = si
	0 = no
Accion a seguir
	 i = insertar tipo de concepto
	u = actualizar tipo de concepto

Proceso:
Insercion o actualizacion de tipos de concepto

Salida:
Tipo de concepto insertado o actualizado

Ejemplo:

```
exec [dbo].sp_tipo_concepto_inserta_modifica 0,1,'Tipo de concepto',1,1,10,0,01,1,1,'i'
Termina
*/

CREATE procedure [dbo].sp_tipo_concepto_inserta_modifica
(
	@id int,
	@idempresa int,
	@descripcion varchar(50),
	@boleto_avion tinyint,
	@kilometraje tinyint,
	@precio_unitario_cantidad tinyint,
	@IVA tinyint,
	@TUA tinyint,
	@propina tinyint,
	@dias tinyint,
	@comensales tinyint,
	@deposito tinyint,

	@proviene char(1)
)
as
begin


	declare
		@idtipo_concepto int

	begin transaction tipo_concepto

	if @proviene = 'i'
	begin
		set @idtipo_concepto = (select isnull(max(idtipo_concepto),0) + 1 from tipo_concepto)

		insert into tipo_concepto
		(
		idempresa,
		idtipo_concepto,
		descripcion,
		boleto_avion,
		kilometraje,
		precio_unitario_cantidad,
		iva,
		tua,
		propina,
		dias,
		comensales,
		deposito
		)
		values
		(
		@idempresa,
		@idtipo_concepto,
		@descripcion,
		@boleto_avion,
		@kilometraje,
		@precio_unitario_cantidad,
		@iva,
		@tua,
		@propina,
		@dias,
		@comensales,
```

```
		@deposito
		)
	end

	if @proviene = 'u'
	begin
		update tipo_concepto
		set
		descripcion = @descripcion,
		boleto_avion = @boleto_avion,
		kilometraje = @kilometraje,
		precio_unitario_cantidad = @precio_unitario_cantidad,
		iva = @iva,
		tua = @tua,
		propina = @propina,
		dias = @dias,
		comensales = @comensales,
		deposito = @deposito
		where	idtipo_concepto = @id and
			idempresa = @idempresa

	end


	commit transaction tipo_concepto


end




GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_tipo_concepto_muestra_datos    Script Date: 02/07/2008 04:36:24
p.m. ******/




/*
Inicia
Nombre:
sp_tipo_concepto_muestra_datos



Funcionalidad:
Muestra los datos particulares de un tipo de concepto
```

Entradas:
Numero de la empresa en la que se esta trabajando
Tipo de concepto

Proceso:
Presenta los datos particulares de un tipo de concepto

Salida:
Lo que ese tipo de concepto pide a los usuarios a la hora de seleccionar un concepto de gasto que pertenece a el

Ejemplo:
exec sp_tipo_concepto_muestra_datos 1,2
Termina
*/

```
CREATE procedure [dbo].sp_tipo_concepto_muestra_datos
(
	@idempresa int,
	@idtipo_concepto int
)
as
begin
	select top 1
	tc.idtipo_concepto,
	tc.descripcion,
	isnull(tc.boleto_avion,0) as boleto_avion,
	isnull(tc.kilometraje,0) as kilometraje,
	isnull(tc.precio_unitario_cantidad,0) as precio_unitario_cantidad,
	isnull(tc.iva,0) as iva,
	isnull(tc.tua,0) as tua,
	isnull(tc.propina,0) as propina,
	isnull(tc.dias,0) as dias,
	isnull(tc.comensales,0) as comensales,
	isnull(tc.deposito,0) as deposito
	from tipo_concepto as tc
	where	tc.idempresa = @idempresa and
		tc.idtipo_concepto = @idtipo_concepto
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_tipo_usuario_catalogo    Script Date: 02/07/2008 04:36:24 p.m. ******/
```

```
/*
Inicia
Nombre:
sp_tipo_usuario_catalogo



Funcionalidad:
Obtiene el catalogo de tipos de usuario

Entradas:
No tiene entradas

Proceso:
Obtencion de los tipos de usuario

Salida:
Tipos de usuario

Ejemplo:
exec sp_tipo_usuario_catalogo
Termina
*/

CREATE procedure [dbo].sp_tipo_usuario_catalogo
as
begin
	select
	idtipo_usuario,
	descripcion
	from tipo_usuario
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_transferencia_catalogo    Script Date: 02/07/2008 04:36:28 p.m.
******/




/*
Inicia
Nombre:
```

sp_transferencia_catalogo

Funcionalidad:
Transferencia de autoridad de un empleado a otro

Entradas:
Identificador del registro de transferencia
Empresa en la que se esta trabajando
Numero de empleado que transfiere su autoridad
Accion a seguir
- e = obtiene los datos de un registro especifico
- u = obtener solo el registro vigente
- h = obtiene el historial de transferencias de autoridad

Proceso:
Obtiene datos de las transfrencias realizadas dependiendo de los parametros de entrada

Salida:
Datos de las transferencias

Ejemplo:
exec [dbo].sp_transferencia_catalogo 1,1,123,'e'
Termina
*/

```
CREATE procedure [dbo].sp_transferencia_catalogo
(
	@idtransferencia int,
	@idempresa int,
	@numempleado int,
	@proviene char(1)
)
as
begin
	/*
	Si se quiere obtener los datos especificos de un registro para modificarlo
	*/
	if @proviene = 'e'
	begin
		select
		idtransferencia,
		[dbo].sp_convierte_fecha_a_texto(convert(char,fecha_ini,103)),
		[dbo].sp_convierte_fecha_a_texto(convert(char,fecha_fin,103)),
		numempleado_delegado,
		observacion
		from transferencia_autoridad
		where	idempresa = @idempresa and
			idtransferencia = @idtransferencia
	end

	/*
	Si se quiere ver solo el registro vigente
	*/
	if @proviene = 'u'
	begin
		select top 1
		tr.idtransferencia,
		rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as nombre,
		convert(char,tr.fecha,103) as fecha_registro,
		convert(char,tr.fecha_ini,103) as fecha_ini,
```

```
		convert(char,tr.fecha_fin,103) as fecha_fin,
		case
			when len(tr.observacion) = 20 then left(tr.observacion,17) + '...'
			else tr.observacion
		end as observacion
		from transferencia_autoridad as tr
		inner join usuario as us on tr.numempleado_delegado = us.numempleado
		where	tr.idempresa = @idempresa and
			tr.idempresa = us.idempresa and
			tr.idstatus = 'a' and
			fecha_fin >= [dbo].fn_obtiene_fecha_hoy(getdate()) and
			tr.numempleado_delega = @numempleado
		order by tr.idtransferencia desc


	end

	/*
	Si se quiere ver solo el historial de todas las transferencias de autoridad
	*/
	if @proviene = 'h'
	begin
		select
		tr.idtransferencia,
		rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as nombre,
		convert(char,tr.fecha,103) as fecha_registro,
		convert(char,tr.fecha_ini,103) as fecha_ini,
		convert(char,tr.fecha_fin,103) as fecha_fin,
		tr.observacion,
		st.descripcion_1
		from transferencia_autoridad as tr
		inner join usuario as us on tr.numempleado_delegado = us.numempleado
		inner join status as st on tr.idstatus = st.idstatus
		where	tr.idempresa = @idempresa and
			tr.idempresa = us.idempresa and
			tr.numempleado_delega = @numempleado
		order by tr.idtransferencia asc


	end

end


GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO



/****** Object:  Stored Procedure dbo.sp_update_object_sequential_key    Script Date: 02/07/2008 04:36:24
p.m. ******/
```

```
CREATE PROC sp_update_object_sequential_key
                @object_name varchar(50) ='dummy_object',
                @last_key_assigned int
AS

UPDATE object_control
        SET     last_key_assigned=@last_key_assigned,
                locked_status_flag=0,
                last_locked_by=SESSION_USER,
                last_locked_at=getdate()
        WHERE           LTRIM([object_name]) = @object_name

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

/****** Object:  Stored Procedure dbo.sp_usuario_area_catalogo    Script Date: 02/07/2008 04:36:28 p.m.
******/

/*
Inicia
Nombre:
sp_usuario_area_catalogo

Funcionalidad:
Obtiene los autorizadores de area que tienen los empleados
```

```
Entradas:
Empresa en la que se esta trabajando
Numero de empleado del que se quieren obtener los autorizadores de area

Proceso:
Obtiene el nombre de los autorizadores por usuario

Salida:
Nombre de los autorizadores

Ejemplo:
exec [dbo].sp_usuario_area_catalogo 1,123
Termina
*/

CREATE procedure [dbo].sp_usuario_area_catalogo
(
	@idempresa int,
	@numempleado int
)
as
begin
	select
	ua.autorizador,
	rtrim(us.nombre) + ' ' + rtrim(us.paterno) + ' ' + rtrim(us.materno) as autorizador
	from usuario as us
	inner join usuario_area_autoriza as ua on ua.autorizador = us.numempleado
	where	us.idempresa = ua.idempresa and
		ua.idempresa = @idempresa and
		ua.numempleado = @numempleado

end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO


/****** Object:  Stored Procedure dbo.sp_usuario_catalogo    Script Date: 02/07/2008 04:36:28 p.m. ******/



/*
Inicia
Nombre:
sp_usuario_catalogo
```

```
Funcionalidad:
Obtiene el catalogo de usuarios

Entradas:
Empresa en la que se trabaja
Accion a seguir
	u = muestra los usuarios activos a excepcion del administrador por defecto (1)
	i = muestra los usuarios internos
	e = muestra los usuarios externos
	a = ver los datos de un usuario particular

Proceso:
Obtiene el catalogo de usuarios

Salida:
Catalogo de usuarios

Ejemplo:
exec [dbo].sp_usuario_catalogo 1,123,'a'
Termina
*/


CREATE procedure [dbo].sp_usuario_catalogo
(
	@idempresa int,
	@proviene char(1) = 'u',
	@numempleado int = 0
)
as
begin
	if @proviene = 'u'
	begin
		select
		numempleado,
		nombre + ' ' + paterno + ' ' + materno
		from usuario
		where	idstatus = 'a' and
			idempresa = @idempresa and
			numempleado != 1
	end

	/*
	Cuando se quieren ver los usuarios internos
	*/
	if @proviene = 'i'
	begin
		select
		numempleado,
		nombre + ' ' + paterno + ' ' + materno
		from usuario
		where	idempresa = @idempresa and
			idtipo_usuario = 1 --Usuario Interno
	end

	/*
	Cuando se quieren ver los usuarios externos
	*/
	if @proviene = 'e'
	begin
		select
		numempleado,
```

```
		nombre + ' ' + paterno + ' ' + materno
		from usuario
		where	idempresa = @idempresa and
			idtipo_usuario = 2 --Usuario Externo
	end

	/*
	Cuando se quieren ver los datos de un empleado en particular
	*/
	if @proviene = 'a' and @numempleado != 0
	begin
		select
		us.numempleado,
		ba.nombre,
		us.cta_cheques,
		us.idtipo_usuario,
		us.idnivel,
		us.idperfil,
		us.idstatus,
		us.puesto,
		us.ccostos,
		us.rfc,
		us.email,
		us.num_jefe,
		di.autorizador,
		us.idbanco,
		rtrim(us.nombre),
		rtrim(us.paterno),
		rtrim(us.materno)
		from usuario as us
		left outer join banco as ba on us.idbanco = ba.idbanco
		left outer join usuario_direccion_autoriza as di on us.numempleado = di.numempleado
		where	us.idempresa = @idempresa and
			us.numempleado = @numempleado
	end
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/*
Inicia
Nombre:
sp_valida_datos_cabeceros_solicitudes

Fecha de ultima modificacion:
25 de octubre del 2005

Nota:
```

Este sp se usa cuando se van a relacionar dos folios. Y se utiliza para validar que estos datos sean los mismos entre el folio principal y el folio relacionado

Funcionalidad:
Obtiene datos cabeceros importantes dependiendo del tipo de solicitud

Entradas:
Folio de la solicitud de la que se quieren obtener los datos

Proceso:
Obtiene diferentes datos dependiendo de la solicitud de que se trate

Salida:
Datos de la solicitud

Ejemplo:
exec [dbo].sp_valida_datos_cabeceros_solicitudes 2
Termina
*/

```
CREATE procedure [dbo].sp_valida_datos_cabeceros_solicitudes
(
	@folio int
)
as
begin
	declare
	@idtipo_operacion as int,
	@tipo_operacion varchar(50),
	@idempresa as int,

	@idmoneda as int,
	@idterritorio as int,
	@idproveedor as int

	set @idtipo_operacion = (select idtipo_operacion from solicitud where folio = @folio)
	set @idempresa = (select idempresa from solicitud where folio = @folio)
	set @tipo_operacion = (select descripcion from tipo_operacion where idtipo_operacion =
@idtipo_operacion and idempresa = @idempresa)

	set @idmoneda = (select idmoneda from solicitud where folio = @folio)
	set @idterritorio = (select idterritorio from solicitud where folio = @folio)

	set @idproveedor = 0

	/*
	Si es una solicitud de pago se obtienen los siguientes datos
	*/
	if rtrim(ltrim(@tipo_operacion)) = 'pago'
	begin
		set @idproveedor = (select idproveedor from pago where folio = @folio)
	end

	/*
	Si es una solicitud de compra se obtienen los siguientes datos
	*/
	if rtrim(ltrim(@tipo_operacion)) = 'compra'
	begin
		set @idproveedor = (select idproveedor from compra where folio = @folio)
	end

	/*
```

```
	Si es una solicitud de reservacion se obtienen los siguientes datos
	*/
	if rtrim(ltrim(@tipo_operacion)) = 'reservacion'
	begin
		set @idproveedor = (select idproveedor from reservacion where folio = @folio)
	end

	select
	@idmoneda as idmoneda,
	@idterritorio as idterritorio,
	@idproveedor as idproveedor
end



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Trigger dbo.tr_ppto_detalle_elimina    Script Date: 02/07/2008 04:36:35 p.m. ******/




/*
Inicia
Nombre:
tr_ppto_detalle_elimina



Funcionalidad:
Cuando un detalle de solicitud se elimina, se elimina el presupuesto gastado o comprometido para ese detalle.

Proceso:
Se actualizan los registros comprometidos o gastados de un detalle.

Termina
*/

CREATE TRIGGER [dbo].tr_ppto_detalle_elimina
ON [dbo].ppto_detalle
after delete
as
	declare
	@monto money,
	@tipo char(1),
	@c_idreg int,
	@c_idppto int,
```

```
@c_detalle int,
@c_tipo char(1),
@c_monto money

declare ppto_detalle cursor for
	select idreg,idppto,detalle,tipo,monto from deleted

	open ppto_detalle

	fetch next from ppto_detalle
		into
			@c_idreg,
			@c_idppto,
			@c_detalle,
			@c_tipo,
			@c_monto

	while @@fetch_status = 0
		begin

			if @c_tipo = 'g'
			begin
				update presupuesto
				set
				ppto_gastado = isnull(ppto_gastado,0) - @c_monto
				where idppto = @c_idppto

			end

			if @c_tipo = 'c'
			begin
				update presupuesto
				set
				ppto_comprometido = isnull(ppto_comprometido,0) - @c_monto
				where idppto = @c_idppto

			end

			fetch next from ppto_detalle
			into
				@c_idreg,
				@c_idppto,
				@c_detalle,
				@c_tipo,
				@c_monto
		end

close		ppto_detalle
deallocate	ppto_detalle
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Trigger dbo.tr_ppto_detalle_actualiza    Script Date: 02/07/2008 04:36:36 p.m. ******/



/*
Inicia
Nombre:
tr_ppto_detalle_actualiza



Funcionalidad:
Cuando una relación de detalle presupuesto se actualiza, se actualiza la tabla de presupuesto para indicar
que un presuuesto ha sido gastado. Por ejemplo, cuando una relación pasa a ser de tipo "g" (gastado), se
toma el monto de esta relacion y en la tabla de presupuesto se elimina ese monto del comprometido y se
actualiza la columna de gastado.

Proceso:
Se actualiza la tabla de presupuesto dependiendo de la actualización de la solicitud.

Termina
*/

CREATE TRIGGER [dbo].tr_ppto_detalle_actualiza
ON [dbo].ppto_detalle
after update
as
        declare
        @monto money,
        @tipo char(1),
```

```
	@c_idreg int,
	@c_idppto int,
	@c_detalle int


	declare ppto_detalle cursor for
		select idreg,idppto,detalle from inserted

		open ppto_detalle

		fetch next from ppto_detalle
			into
				@c_idreg,
				@c_idppto,
				@c_detalle

		while @@fetch_status = 0
			begin
				set @monto = (select monto from ppto_detalle where idreg = @c_idreg)
				set @tipo = (select tipo from ppto_detalle where idreg = @c_idreg)

				/*
				Si una solicitud se actualiza a "g" (gastado)
				Se actualiza la columna de ppto_gastado para que incluya este nuevo
monto
				Se elimina ese monto de la columna ppto_comprometido
				*/
				if @tipo = 'g'
				begin
					update presupuesto
					set
					ppto_gastado = isnull(ppto_gastado,0) + @monto
					where idppto = @c_idppto

					update presupuesto
					set
					ppto_comprometido = isnull(ppto_comprometido,0) - @monto
					where idppto = @c_idppto
				end

				/*
				Si una solicitud se actualiza a "c" (comprometido)
				Se elimina ese monto de la columna ppto_gastado
				Se actualiza la columna de ppto_comprometido para que incluya este
nuevo monto
				*/
				if @tipo = 'c'
				begin
					update presupuesto
					set
					ppto_gastado = isnull(ppto_gastado,0) - @monto
					where idppto = @c_idppto

					update presupuesto
					set
					ppto_comprometido = isnull(ppto_comprometido,0) + @monto
					where idppto = @c_idppto
				end

				fetch next from ppto_detalle
				into
					@c_idreg,
```

```
                                        @c_idppto,
                                        @c_detalle
                        end

        close           ppto_detalle
        deallocate      ppto_detalle



GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO


/****** Object:  Trigger dbo.tr_solicitud_actualiza    Script Date: 02/07/2008 04:36:35 p.m. ******/



/*
Inicia
Nombre:
tr_solicitud_actualiza



Funcionalidad:
Registra el presupuesto gastado o comprometido por solicitud.

Proceso:
Evalúa el status al que se actualizó la solicitud. Si la solicitud se actualizó a un estado de "Solicitud
Actualizada" (at) el prespuesto de esa solicitud se ha gastado, entonces se registra el caracter "g" (gastado)
para esa solicitud. Cualquier otro estatus diferente de AT, excepto un status de rechazo sugiere que la
solicitud se encuentra en proceso y, el presupuesto, comprometido. Cuando una solicitud se rechaza, se
elimina su presupuesto comprometido o gastado.

Termina
*/

/*
update solicitud set idstatus = 'ec' where folio = 1
update solicitud set idstatus = 'ea' where folio = 5
delete from historial_solicitud where folio = 5 and idstatus = 'ep'
*/

CREATE TRIGGER [dbo].tr_solicitud_actualiza
ON [dbo].solicitud
after update
as
        declare
        @c_idstatus char(2),
```

```
	@c_folio int,
	@c_idempresa int,
	@c_idtipo_operacion int,
	@c_idstatus_anterior char(2)

	/*
	Se obtienen los datos recién actualizados para la solicitud
	*/
	set @c_idstatus = (select idstatus from inserted)
	set @c_folio = (select folio from inserted)
	set @c_idempresa = (select idempresa from inserted)
	set @c_idtipo_operacion = (select idtipo_operacion from inserted)

	/*
	Si se ha actualizado el campo status
	*/
	if update(idstatus)
		/*
		Se evalua cual fue el status anterior al status actualizado
		*/
		begin
			set @c_idstatus_anterior = (
				select idstatus
				from flujo_autoriza
				where	idempresa = @c_idempresa and
					idtipo_operacion = @c_idtipo_operacion and
					idstatus_sig = @c_idstatus
			)

			/*
			Si el estado anterior fue ppto, es decir, si recién se acaba de terminar el rol de
			autorización de prespuesto y el presupuesto de la solicitud no ha sido
comprometido:
			*/
			if @c_idstatus_anterior = 'ep' and
[dbo].fn_ppto_obtiene_folio_en_ppto_detalle(@c_folio) = 0
			begin
				/*
				Se compromete el prespuesto
				*/
				exec [dbo].sp_ppto_inserta_modifica @c_folio
			end

			/*
			Si la autorización se ha actualizado con un estado de rechazo, es decir, alguien la
acaba de
			rechazar, entonces el presupuesto comprometido se elimina
			*/
			if rtrim(substring(@c_idstatus,1,1)) = 'r'
			begin
				delete from ppto_detalle
				where	detalle in (select detalle from detalle_solicitud where folio =
@c_folio)
			end

			/*
			Si la solicitud ya termino el flujo de autorizacion y el monto de la solicitud no ha sido
			presupuestado, entonces se registra ese presupuesto comprometido.
			*/
			if rtrim(@c_idstatus) = 'at' and
[dbo].fn_ppto_obtiene_folio_en_ppto_detalle(@c_folio) = 0
			begin
```

```
                        exec [dbo].sp_ppto_inserta_modifica @c_folio
                end

                /*
                Si la solicitud ha terminado el flujo entonces su monto se considera un presupuesto
gastado
                */
                if rtrim(@c_idstatus) = 'at'
                begin
                        update ppto_detalle
                        set tipo = 'g'
                        where     detalle in (select detalle from detalle_solicitud where folio =
@c_folio)
                end

        end
```

```
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
/****** Object:  Trigger dbo.tr_solicitud_elimina    Script Date: 02/07/2008 04:36:35 p.m. ******/
```

```
/*
Inicia
Nombre:
tr_solicitud_actualiza
```

Funcionalidad:
Cuando una solicitud se elimina, se elimina el presupesto comprometido o gastado para esa solicitud.

Proceso:
Se eliminan los registros de de la solicitud eliminada de la tabla que contiene la esta relación.

```
Termina
*/


CREATE TRIGGER [dbo].tr_solicitud_elimina
ON [dbo].solicitud
after delete
as
        declare
        @folio int

        set @folio = (select folio from inserted)

        delete from ppto_detalle
        where     detalle in (select detalle from detalle_solicitud where folio = @folio)









GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

Anexo 2. Código de la aplicación.

```
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../logout/logout.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -           'Fecha de ultima modificacion       : 5 de julio 2007           'Funcionalidad
general                     : Pagina de termino de sesion%><!--         #include
file="../includes/muestr_boton.asp"--><!--    Este archivo incluye la funcion que los botones--><%
        session("idempresa") = 0  session("numempleado") = 0%><html>      <head>          <title>Sesion
Terminada</title>           <link rel="stylesheet" href="../stylesheet/style.css">          <script
src="../includes/no_right_mouse.js"></script>                   <script language=javascript>                    <!--
                    function pagina_inicio(){                                   //Funcion que redirecciona a la
pagina de login                                 document.form.action = '../login.asp?logout=logout';
                document.form.submit();                     }                                   //Se evalua
si la pagina fue abierta por otra y se pintan las opciones                                  //correspondientes a
cada caso                                   if(window.opener){
        texto.innerHTML = 'Favor de cerrar esta ventana y volver a firmarse';
        boton.innerHTML = '<input class=boton type=button value=cerrar onclick=window.close();>';
                    }                                   else{
        texto.innerHTML = 'Favor de volver a firmarse';
        boton.innerHTML = '<input class=boton type=button value="Pagina de Login"
onclick=pagina_inicio();>';                                 }                   }                   -->
        </script></head> <body onload=javascript:tiene_opener();>   <script
src="../includes/no_right_mouse.js"></script>           <form name="form" method="post">
        <table width=50.5% border=0>                    <tr>                                <td
class=tabla_nivel_2 align=center>Sesion Terminada</td>                  </tr>           </table>
        <table width=50% border=0 class=contenedor>                 <tr>
        <td class=texto_nivel_2 align=center>Su sesion ha sido terminada
        <div id=texto>                                  <!--Aqui se pinta un boton dependiendo de si la
ventana abierta es nueva o no-->                            </div>
        </td>                   </tr>                   <tr>            </table>            <table
width=50.5% border=0>                           <td align=center>                               <div
id=boton>                                       <!--Aqui se pinta un boton dependiendo de si la ventana
abierta es nueva o no-->                            </div>                          </td>
                </tr>           </table> </form>  </body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../reporte_autoriza/repor_autor_autor.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -           'Fecha de ultima modificacion       : 5 de julio 2007           'Funcionalidad
general                     : Pagina que muestra el flujo de autorizacion               '
                                    pendiente para cada solicitud%><!--         #include
file="../connect/conne_open.asp"--><!--     Este archivo contiene la conexion a la BD--><!--     #include
file="../includes/catal_centr_costo.asp"--><!--         Este archivo incluye la funcion que muestra el catalogo
de centros de costo--><!-- #include file="../includes/muestr_boton.asp"--><!--    Este archivo incluye la funcion
que los botones--><!--      #include file="../includes/forma_fecha.asp"--><!--      Este archivo incluye la funcion
que da formato a las fechas--><%
        '=============================================================================
===============                 'Estas variables son comunes a todas las paginas                dim
numempleado                                     'Numero de empleado que esta firmado
en la aplicacion            dim idempresa                                               'Numero de
empresa en la que se esta trabajando                numempleado = request("numempleado")
        idempresa = request("idempresa")
        '=============================================================================
===============
        '=============================================================================
```

```
==========                    'Estas variables se usan nada mas en las paginas de solicitudes
        dim folio                                                    'Numero de folio de la solicitud
                folio = request("folio")
        '=============================================================================
==========
        '=============================================================================
==========                    'Estas variables se usan nada mas en las paginas de solicitudes
        dim status                                                   'Status de la solicitud
                status = request("status")
        '=============================================================================
==========
        '=============================================================================
==========                    'Esta variable evalua si se muestra o no el menu              dim
ver_menu                ver_menu = request("ver_menu")
        '=============================================================================
==========                    function datos_grales(folio,clase)                    'Esta funcion
presenta los datos generale de la solicitud                  dim sql                 dim res
        sql = "exec [dbo].sp_muestra_datos_grale "&folio&" "                set res = connect.execute(sql)
                        if not res.eof then                     response.Write "<table
width=""90%"">"                         response.Write " <tr class=tabla_nivel_2>"
        response.Write "        <td colspan=2>Flujo de Autorizacion</td>"
        response.Write " </tr>"                         response.Write " <tr class="&clase&">"
                        response.Write "        <td>Tipo de solicitud</td>"
        response.Write "        <td>"&res(7)&"</td>"                                    response.Write "
        </tr>"                  response.Write " <tr class="&clase&">"
        response.Write "        <td>Folio</td>"                         response.Write "
        <td>"&res(0)&"</td>"                            response.Write " </tr>"
        response.Write " <tr class="&clase&">"                          response.Write "
        <td>Solicitante</td>"                           response.Write "        <td>"&res(1)&"</td>"
                        response.Write " </tr>"                         response.Write " <tr
class="&clase&">"                       response.Write "        <td>Status</td>"
                response.Write "        <td>"&res(5)&"</td>"
        response.Write " </tr>"                         end if          end function            function
main_function(folio,status)             'Esta funcion presenta los autorizadores pendientes por solicitud
                        'Recibe:                 'folio = folio de la solicitud                 'status =
status a partir del cual se presentaran los autorizadores                       'pendientes
        dim sql                 dim res                 dim arRes                       dim t_arRes
                        dim count                       sql = "exec sp_reporte_consulta_autorizadores
"&folio&","'&status&"','p',0,1"                 set res = connect.execute(sql)                  if
not res.eof then                        arRes = res.getrows                             t_arRes =
ubound(arRes,2)                 else                            t_arRes = -1                    end
if                      response.Write "<tr>"                   response.Write " <td colspan=2
align=center><hr></td>"                 response.Write "</tr>"                  if t_arRes >= 0 then
                                        response.Write "<tr class=tabla_nivel_3>"
                response.Write " <td colspan=1>"                                        response.Write "
        Status"                                 response.Write " </td>"
        response.Write " <td colspan=1>"                                        response.Write "
        Responsable"                                    response.Write " </td>"
                response.Write "</tr>"                          for count = 0 to t_arRes
                        response.Write "<tr class=texto_nivel_2>"
        response.Write " <td colspan=1>"                                        response.Write
        arRes(2,count)                                  response.Write " </td>"
                response.Write " <td colspan=1>"                                        response.Write
                arRes(4,count)                          response.Write " </td>"
                        response.Write "</tr>"                  next                    else
                                response.Write "<tr class=aviso>"
        response.Write " <td colspan=2 align=center>"                                   response.Write "
        Todavia no se tiene un flujo definido, favor de consultar con el administrador"
        response.Write " </td>"                         response.Write "</tr>"                  end
if                      response.Write " <tr>"                  response.Write "        <td
colspan=2 align=center><hr></td>"               response.Write " </tr>"
        response.Write " <tr>"                  response.Write "        <td colspan=2 align=center>"
```

```
                                                                                'Funcion que muestra
los botones de operacion
        'Localizacion = "../includes/muestr_boton.asp"                          response.Write "
        </td>"                          response.Write " </tr>"                 response.Write "</table>"
        end function%><html>    <head>          <title>Flujo de Autorizaci&oacute;n</title>
        <script src="../includes/no_right_mouse.js"></script>           <link rel="stylesheet"
href="../stylesheet/style.css">                 <script language=javascript>                <!--                -->
                </script></head> <body>  <%if ver_menu<>0 then%>        <!--#include
file=../menu/menu.asp-->  <%end if%>    <form name="form" method="post">                <input
type=hidden name=idempresa              value=<%=idempresa%>>           <input type=hidden
name=numempleado                value=<%=numempleado%>>         <input type=hidden
name=ver_menu           value=<%=ver_menu%>>            <%                      call
datos_grales(folio,"texto_nivel_2")                     'Funcion que da al usuario la opcion de anexar un
archivo a la solicitud                  'Localizacion = esta misma pagina           %>      </form>
        </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../reporte_autoriza/repor_autor_usuar.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -           'Fecha de ultima modificacion       : 9 de agosto 2007
        'Funcionalidad general                  : Pagina mediante la cual el usuario revisa el estado de
todos                   '                                                               sus documentos
dentro de la aplicacion%><!--       #include file="../connect/conne_open.asp"--><!--       Este archivo contiene
la conexion a la BD--><!-- #include file=../includes/funci_pagin.asp--><!--        Este archivo incluye la funcion
que le da la paginacion al reporte--><!--       #include file="../includes/catal_moned.asp"--><!--      Este archivo
incluye la funcion que presenta el catalogo de monedas--><!--#include
file="../includes/catal_opera_usuar.asp"--><!--         Este archivo incluye la funcion que muestra al usuario el
tipo de operacion a seleccionar--><!--          #include file="../includes/catal_statu.asp"--><!--       Este archivo
incluye la funcion que muestra al usuario los diferentes status para buscar su solicitud--><!--       #include
file="../includes/catal_prove.asp"--><!--       Este archivo incluye la funcion que presenta el catalogo de
proveedores--><!--          #include file="../includes/muestr_texto.asp"--><!--        Este archivo incluye la funcion
que presenta una caja de texto--><!--           #include file="../includes/muestr_fecha.asp"--><!--       Este archivo
incluye la funcion que presenta las cajas de texto para que se seleccione una fecha--><!--         #include
file="../includes/muestr_boton.asp"--><!--      Este archivo incluye la funcion que los botones--><!--#include
file="../includes/forma_fecha.asp"--><!--       Este archivo incluye la funcion que da formato a las fechas--><%

        '=================================================================================
===============                 'Estas variables son comunes a todas las paginas            dim
numempleado                                             'Numero de empleado que esta firmado
en la aplicacion            dim idempresa                                               'Numero de
empresa en la que se esta trabajando                numempleado = request("numempleado")
        idempresa = request("idempresa")
        '=================================================================================
===============
        '=================================================================================
===============                 'Estas variables se usa para saber de donde proviene la peticion de esta
pagina, se usa en               'la mayoria de las paginas      dim proviene                proviene =
request("proviene")
        '=================================================================================
===============                 'Numero de registros de busqueda por hoja, maximo 50            dim
no_reg          no_reg = request("no_reg")              if trim(no_reg) = "" then
        no_reg = 50             else                    no_reg = no_reg         end if
        '=================================================================================
===============                 'Tipo de operacion, en caso de que se busque una especifica     dim
tipo_operacion          tipo_operacion = request("tipo_operacion")
        '=================================================================================
===============                 dim rst         dim intPageSize,PageIndex,TotalPages            dim
RecordCount,RecordNumber,Count              dim intPrev,intNext             function
presenta_resultado(tipo_operacion,idempresa,numempleado)            'Funcion que hace la
busqueda dinamica dependiendo de lo             'introducido en la forma de consulta
```

```
                dim sql                 dim inner_pago                  dim inner_reservacion
                dim inner_compra                        dim anexo_tipo_operacion                dim
anexo_proveedor                 dim anexo_factura                       dim anexo_depto
        dim anexo_moneda                        dim anexo_fecha                 dim anexo_estado
                        dim anexo_folio                 dim backcolor                   if
trim(tipo_operacion) <> "0" then                        anexo_tipo_operacion = " and
so.idtipo_operacion = [dbo].fn_obtiene_tipo_operacion("&idempresa&","'&trim(tipo_operacion)&"')"
                        select case tipo_operacion                                      case "pago"
        :                                               inner_pago = " inner join pago as pa on so.folio
= pa.folio "                                            if request("idproveedor") <> 0 then
                                        anexo_proveedor = " and pa.idproveedor =
"&request("idproveedor")&" "                                    else
                                anexo_proveedor = ""                                    end
if                                              if trim(left(replace(request("factura"),"'",""),10)) <> "" then
                                                anexo_factura = " and pa.no_factura like
'%"&left(replace(request("factura"),"'",""),10)&"%' "                                   else
                                                anexo_factura = ""
                end if                                  case "reservacion", "comprobacion_reservacion"
        :                                               inner_reservacion = " inner join reservacion as
re on so.folio = re.folio "                                     if request("idproveedor") <> 0 then
                                                anexo_proveedor = " and re.idproveedor =
"&request("idproveedor")&" "                                    else
                                anexo_proveedor = ""                                    end
if                                      case "compra"   :
        inner_compra = " inner join compra as co on so.folio = co.folio "
                if request("idproveedor") <> 0 then
        anexo_proveedor = " and co.idproveedor = "&request("idproveedor")&" "
                        else                                                    anexo_proveedor = ""
                                                end if                                  if
trim(left(replace(request("depto"),"'",""),50)) <> "" then
        anexo_depto = " and co.depto like '%"&replace(left(replace(request("depto"),"'",""),50)," ","%")&"%' "
                                                else
        anexo_depto = ""                                        end if                  end
select                  else                                    anexo_tipo_operacion = ""       end
if                      if request("idmoneda") <> 0 then                        anexo_moneda = "
and so.idmoneda = "&request("idmoneda")&" "                     else
        anexo_moneda = ""                       end if                  if request("fecha_ini") <> ""
and request("fecha_fin") <> "" then                     anexo_fecha =                   "
and convert(char,[dbo].sp_convierte_texto_a_fecha("'&request("fecha_ini")&"'),103) <=
convert(char,so.fecha_solicitud,103) "                  anexo_fecha = anexo_fecha & " and
convert(char,[dbo].sp_convierte_texto_a_fecha("'&request("fecha_fin")&"'),103) >=
convert(char,so.fecha_solicitud,103) "                  else                                    anexo_fecha
= ""                    end if                  if request("idstatus") <> "0" then
        select case lcase(trim(request("idstatus")))                                    case "n" :
anexo_estado = " and substring(so.idstatus,1,1) = 'n' "                                 case "e" :
anexo_estado = " and substring(so.idstatus,1,1) = 'e' "                                 case "r"  :
anexo_estado = " and substring(so.idstatus,1,1) = 'r' "                                 case "a" :
anexo_estado = " and ltrim(rtrim(so.idstatus)) in ('at','tp','pg') "                    end select
                        else                            anexo_estado = ""                       end if
                        if trim(request("folio")) <> "" then                    anexo_folio = " and so.folio =
"&request("folio")&" "                  else                                    anexo_folio = ""
        end if                  sql = sql & " select "                          sql = sql & " so.folio, "
                        sql = sql & " 'Historial', "            sql = sql & "
[dbo].fn_obtiene_total_por_folio(so.folio) as monto, "          sql = sql & " mn.identificador as
moneda, "               sql = sql & " convert(char,so.fecha_solicitud,103) as fecha_solicitud, "
                        sql = sql & " case "                    sql = sql & "   when len(so.descripcion) >=
15 then left(so.descripcion,12) + '...' "               sql = sql & "   else so.descripcion "
                        sql = sql & " end as descripcion, "             sql = sql & " st.idstatus, "
        sql = sql & " st.descripcion_1 "                sql = sql & " from solicitud as so "
        sql = sql & " inner join status as st on so.idstatus = st.idstatus "                    sql = sql &
  inner_pago            sql = sql &     inner_reservacion                       sql
= sql &   inner_compra          sql = sql & " where so.idempresa = "&idempresa&" and "
```

```
                sql = sql & "                so.numempleado = "&numempleado&" "                sql
= sql &         anexo_tipo_operacion                sql = sql &
        anexo_proveedor                 sql = sql &                 anexo_factura
        sql = sql &                     anexo_depto                 sql = sql &
        anexo_moneda                    sql = sql &                 anexo_fecha
        sql = sql &                     anexo_estado                sql = sql &
        anexo_folio                     sql = sql &         " order by so.folio asc"            set
Rst=server.CreateObject("ADODB.recordset")              Rst.Open sql,application("odbc"),3,1,1
                        if Rst.EOF then                             Response.Write
"<tr><td colspan=5 align=center class=aviso>"                       response.Write " No
se encontraron registros con esos parametros de busqueda"
        response.Write " </td></tr>"                    else
        intpageSize=cint(no_reg)                        PageIndex=request("PageIndex")
                        if PageIndex="" then PageIndex=1
        RecordCount=Rst.RecordCount                             RecordNumber=(intPageSize
* PageIndex) - intPageSize                       Rst.PageSize =intPageSize
                        Rst.AbsolutePage = PageIndex
        TotalPages=Rst.PageCount                                intPrev=PageIndex - 1
                        intNext=PageIndex +1                                Count=1
                        response.Write "<tr>"
        response.Write " <td colspan=8><hr></td>"                           response.Write
"</tr>"                         response.Write "<tr class=tabla_nivel_3>"
                response.Write " <td>Folio</td>"                            response.Write "
        <td>Historial</td>"             response.Write " <td>Monto</td>"
                        response.Write " <td>Moneda</td>"
        response.Write " <td>Fecha de solicitud</td>"
        response.Write " <td>Descripcion</td>"                              response.Write "
        <td>Estado</td>"                        response.Write " <td>Informacion
relevante</td>"                         response.Write "</tr>"
        while not Rst.EOF and Count<=intPageSize                            if Count mod
2 <>0 then                              backcolor = "#dcdcdc"
                        else                                                backcolor =
"white"                                 end if
        response.Write "<tr class=texto_nivel_2 bgcolor="&backcolor&">"
                response.Write " <td><a
href=javascript:ver_detalle("&rst.Fields(0)&","&rst.Fields(6)&");>"&Rst.Fields(0)&"</a></td>"
                        response.Write " <td><a
href=javascript:ver_historial("&rst.Fields(0)&");>"&Rst.Fields(1)&"</a></td>"
                response.Write " <td>"&formatcurrency(Rst.Fields(2),2)&"</td>"
                        response.Write " <td>"&Rst.Fields(3)&"</td>"
                response.Write " <td>"&formato_fecha(Rst.Fields(4))&"</td>"
                response.Write " <td>"&Rst.Fields(5)&"</td>"
        response.Write " <td>"&Rst.Fields(7)&"</td>"
        select case lcase(trim(mid(Rst.Fields(6),1,1)))
        case "n","e","r"    :
        response.Write " <td><a
href=javascript:autorizaciones_pendientes("&rst.Fields(0)&","&rst.Fields(6)&");>Autorizaciones
pendientes</a></td>"                                                    case "a"            :
                                                        response.Write " <td>Autorizaciones
OK</td>"                                                case "t"            :
                                                    response.Write " <td>Fecha de
importacion</td>"                                       case "p"            :
                                                    response.Write " <td>Datos de pago</td>"
                                    end select
        response.Write "</tr>"                                      Rst.MoveNext
                                Count=Count + 1                             wend
                        response.Write "<tr>"                               response.Write " <td
colspan=8><hr></td>"                                    response.Write "</tr>"
                call button(intPrev,intNext,TotalPages)
        Response.Write "</table>"                               set rst=nothing             end
if              end function            function busqueda(tipo_operacion,idempresa,numempleado)
                'Esta funcion presenta al usuario los datos de la busqueda                   'Detalle =
```

```
numero del detalle dentro de la solicitud                    response.Write "<table border=0 width=""90%""
align=center>"                    response.Write "</table>"          end function          function
datos_particulares(tipo_operacion,clase)                    'Funcion que presenta los datos de busqueda
particulares                    'en caso de que se haya seleccionado un tipo de operacion en particular
                    select case lcase(trim(tipo_operacion))                                        case
"reservacion","comprobacion_reservacion" :                                        response.Write "<tr>"
                                                response.Write "  <td></td>"
                                                call
catalogo_proveedor(idempresa,"proveedor","idproveedor",request("idproveedor"),clase)
                    response.Write "</tr>"                                        case "pago" :
                                                response.Write "<tr>"
          response.Write "  <td></td>"
                                                call
catalogo_proveedor(idempresa,"proveedor","idproveedor",request("idproveedor"),clase)
                                                response.Write "</tr>"
          response.Write "<tr>"                                        response.Write "  <td></td>"
                                                call
muestra_texto("No. Factura","factura",left(replace(request("factura"),""",""),10),clase,"12","10")
                                                response.Write "</tr>"                                        case "compra" :
                                                response.Write "<tr>"
          response.Write "  <td></td>"
                                                call
catalogo_proveedor(idempresa,"proveedor","idproveedor",request("idproveedor"),clase)
                                                response.Write "</tr>"
          response.Write "<tr>"                                        response.Write "  <td></td>"
                                                call
muestra_texto("Departamento solicitante","depto",left(replace(request("depto"),""",""),50),clase,"30","50")
                                                response.Write "</tr>"                                        case
"contrato" :                                        response.Write "<tr>"
                                                response.Write "  <td></td>"
                                                call
catalogo_proveedor(idempresa,"proveedor","idproveedor",request("idproveedor"),clase)
                                                response.Write "</tr>"                    end select          end function
                    function main_function()                    'Esta funcion presenta al usuario la posibilidad
de hacer busquedas                    'de solicitudes dentro de la aplicacion
          response.Write "<table border=0 width=""90.5%"" align=center>"
          response.Write "  <tr>"                    response.Write "          <td class=tabla_nivel_2
colspan=2>Reporte de Autorizaciones</td>"                    response.Write "  </tr>"
          response.Write "<table>"                    response.Write "<table border=0 width=""90%""
align=center class=contenedor>"                    response.Write "  <tr>"
          response.Write "          <td>"
                    call catalogo_operaciones_usuario(idempresa,numempleado,"Tipo de
operacion","tipo_operacion",tipo_operacion,"texto_nivel_2")                    response.Write "
          </td>"                    response.Write "  </tr>"                    response.Write "  <tr>"
                    response.Write "          <td>"
                                                call
catalogo_moneda("Moneda","idmoneda",request("idmoneda"),"texto_nivel_2","")
          response.Write "          </td>"                    response.Write "  </tr>"
          response.Write "  <tr>"                    response.Write "          <td>"
                                                call muestra_fecha("Fecha de
Inicio","fecha_ini",request("fecha_ini"),"texto_nivel_2")                    response.Write "
          </td>"                    response.Write "  </tr>"                    response.Write "  <tr>"
                    response.Write "          <td>"
                                                call muestra_fecha("Fecha de Fin","fecha_fin",request("fecha_fin"),"texto_nivel_2")
                                                response.Write "          </td>"                    response.Write "  </tr>"
                    response.Write "  <tr>"                    response.Write "          <td>"
                                                call
catalogo_status("Estado","idstatus","ra",request("idstatus"),"texto_nivel_2")
          response.Write "          </td>"                    response.Write "  </tr>"
          response.Write "  <tr>"                    response.Write "          <td>"
                                                call
muestra_texto("Folio","folio",request("folio"),"texto_nivel_2","12","10")                    response.Write "
```

```
        </td>"                          response.Write " </tr>"
                                        call datos_particulares(tipo_operacion,"texto_nivel_2")
        response.Write " <tr>"                          response.Write " <tr>"
        response.Write "        <td>"
                call muestra_texto("Registros por hoja","no_reg",no_reg,"texto_nivel_2","4","3")
        response.Write "        </td>"                  response.Write " </tr>"
        response.Write " <td></td>"                     response.Write "        <td colspan=2
align=center>"                                                                  call
muestra_boton("Buscar","buscar();")                     response.Write "                 "
                                                                        call
muestra_boton("Limpiar","limpiar();")
                'Funcion que muestra los botones de operacion
                                        'Localizacion = "../includes/muestr_boton.asp"
        response.Write "        </td>"                  response.Write " </tr>"
        response.Write "<table>"        end function%><html>    <head>          <title>Reporte de
Solicitudes</title>         <script src="../includes/open_windo.js"></script>           <script
src="../includes/valida_fechas.js"></script>            <script src="../includes/calen_picke.js"></script>
        <script src="../includes/no_right_mouse.js"></script>           <link rel="stylesheet"
href="../stylesheet/style.css">                 <script language=javascript>                    <!--
        function selecciona_tipo_operacion(){                                   //Funcion que  dispara la
seleccion de un tipo de operacion                       //especifica, dependiendo de esto, se presentan
unas u otras                                    //opciones
        document.form.idmoneda.value = '';                                      document.form.fecha_ini.value
= '';                                   document.form.fecha_fin.value = '';
        document.form.idstatus.value = '';                      document.form.folio.value = '';
                        document.form.no_reg.value = '';
        if(document.form.idproveedor)
        document.form.idproveedor.value = '';                                   if(document.form.depto)
                                        document.form.depto.value = '';
        if(document.form.factura)                               document.form.factura.value = '';
                        document.form.proviene.value = '';                              document.form.action
= 'repor_autor_usuar.asp';                      document.form.submit();                 }
                function limpiar(){                     //Funcion que limpia los campos de busqueda
                                        document.form.tipo_operacion.value = '';
        selecciona_tipo_operacion();                    }                       function
ver_detalle(folio,status){                      //Funcion de despliega el detalle de las solicitudes

        open_window('../solicitud/solic_ver_detal.asp?folio='+folio+'&proviene=US&status='+status+'&ver_me
nu=0','ver_detalle','width=600,height=300,scrollbars=yes,resizable=yes,status=yes');                    }
                        function selecciona_moneda(tipo_operacion,valor){               }
                function ver_historial(folio){                                  //Funcion que despliega el
historial de las solicitudes
        open_window('../historial/repor_histo.asp?folio='+folio+'','historial','width=900,height=300,scrollbars=y
es,resizable=yes,status=yes');                  }                               function
autorizaciones_pendientes(folio,status){                        //Funcion que obtiene las
autorizaciones pendientes
       open_window('repor_autor_autor.asp?folio='+folio+'&status='+status+'','repor_autorizaciones','width=6
00,height=300,scrollbars=yes,resizable=yes,status=yes');                        }
        function buscar(){                      //Funcion que dispara el proceso de busqueda
                        if((document.form.fecha_ini.value!='')&&(document.form.fecha_fin.value=='')){
                                alert('Favor de introducir las dos fechas de busqueda');
                        return false;                           }
        if((document.form.fecha_ini.value=='')&&(document.form.fecha_fin.value!='')){
                alert('Favor de introducir las dos fechas de busqueda');
        return false;                           }
        if((document.form.fecha_ini.value!='')&&(document.form.fecha_fin.value!='')){
                if(!valida_fechas(document.form.fecha_ini.value,document.form.fecha_fin.value)){
                                        alert('Fecha de inicio mayor a fecha de fin');
                        return false;                                           }
        }                               if(isNaN(document.form.folio.value)){
                alert('Valor no valido para el folio');
        document.form.folio.value = '';                                         document.form.folio.focus();
```

```
                                        return false;                                        }
                    if(isNaN(document.form.no_reg.value)||(document.form.no_reg.value==0)){
                              alert('Valor no valido para el numero de resultados deseado');
                    document.form.no_reg.value = 50;
          document.form.no_reg.focus();                                        return false;
                    }                                        document.form.proviene.value = 'B';
                    document.form.action = 'repor_autor_usuar.asp';
          document.form.submit();                                        }                    -->                    </script></head>
          <body>  <!--#include file=../menu/menu.asp-->          <form name="form" method="post">
          <input type=hidden name=idempresa                                        value=<%=idempresa%>>
          <input type=hidden name=numempleado                                        value=<%=numempleado%>>
                    <input type=hidden name=proviene                                        value=<%=proviene%>>
          <%                    call main_function()                                        'Funcion que presenta al
usuario los campos de busqueda                    'Localizacion = esta misma pagina                    if
proviene = "B" then                                        call busqueda(tipo_operacion,idempresa,numempleado)
                    end if          %>          </form>  </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../login.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
          'Fte_ - -          'Fecha de ultima modificacion        : 5 de julio 2007          'Funcionalidad
general                    : Pagina mediante la cual los usuarios se firman a la aplicacion%><!-- #include
file="connect/conne_open.asp"--><!--          Este archivo contiene la conexion a la BD--><!--          #include
file="includes/muestr_texto.asp"--><!--          Este archivo incluye la funcion que presenta una caja de texto--
><!--          #include file="includes/catal_empre.asp"--><!--          Este archivo incluye la funcion que el catalogo
de empresas--><!--          #include file="includes/catal_tipo_usuar.asp"--><!--          Este archivo incluye la funcion
del catalogo de tipos de usuario--><!--          #include file="includes/muestr_boton.asp"--><!--          Este archivo
incluye la funcion que los botones--><%
          '================================================================================
===============          'Estas variables son comunes a todas las paginas          dim
numempleado                                        'Numero de empleado que esta firmado
en la aplicacion          dim idempresa                                        'Numero de
empresa en la que se esta trabajando          numempleado = request("numempleado")
          idempresa = request("idempresa")
          '================================================================================
===============
          '================================================================================
===============          'Esta variable almacena la proveniencia de la peticion de las paginas
          dim proviene          'Variable que indica de donde viene la peticion de esta pagina
          proviene =          request("proviene")
          '================================================================================
===============
          '================================================================================
===============          'Esta variable se utiliza para imprimir en pantalla el mensaje de envio de
correos          dim mensaje          mensaje = replace(request("mensaje"),"_"," ")
          '================================================================================
===============          function autentica(idempresa,numempleado,idtipo_usuario,pwd)
                    'Funcion que autentica a un usuario                    'Recibe:
          'idempresa = empresa a la que el usuario se quiere firmar                    'numempleado =
numero de empleado con el que se quiere firmar                    'idtipo_usuario = tipo de usuario con el
que se quiere firmar                    'pwd = clave de acceso del usuario                    dim sql
                    dim res                    dim valid                    sql = "select
[dbo].fn_autentica_login("&idempresa&","&numempleado&","&idtipo_usuario&","'&pwd&"')"
          set res = connect.execute(sql)                    if not res.eof then
          valid = res(0)                    else                    valid = 0                    end
if                    if valid = 1 then                    session("idempresa") = idempresa
                    session("numempleado") = numempleado                    autentica =
true                    else                    autentica = false                    end if
          end function          function main_function(mensaje)                    'Esta funcion
presenta al usuario los campos de entrada de datos                    'para firmarse
```

```
        'Recibe                 'mensaje = mensaje de correo electronico
        response.Write "<table border=0 width=""50.5%"" align=center>"
        response.Write " <tr>"                  response.Write "        <td colspan=2
class=tabla_nivel_2>"           response.Write "                Datos de acceso"
        response.Write "        </td>"                  response.Write " </tr>"
        response.Write "</table>"               response.Write "<table border=0 width=""50%""
class=contenedor>"              response.Write " <tr>"
                                call muestra_texto("Numero de
empleado","numempleado","","texto_nivel_2",20,10)               response.Write " </tr>"
        response.Write " <tr>"
        call catalogo_empresa("Empresa","idempresa","","texto_nivel_2")
        response.Write " </tr>"                 response.Write " <tr>"
                                call catalogo_tipo_usuario("Tipo de
usuario","idtipo_usuario","","texto_nivel_2")           response.Write " </tr>"
        response.Write " <tr>"                  response.Write "        <td class=texto_nivel_2>"
                response.Write "                Clave de acceso"                response.Write "
        </td>"                  response.Write "        <td>"                   response.Write "
                <input type=password class=caja_texto name=pwd size=20 maxlength=10>"
        response.Write "        </td>"                  response.Write " </tr>"                 if
trim(mensaje) <> "" then                response.Write "<tr>"
        response.Write " <td colspan=2 class=aviso align=center>"&mensaje&"</td>"
        response.Write "</tr>"                  end if                  response.Write "</table>"
                response.Write "<table border=0 width=""50.5%"" align=center>"
        response.Write " <tr>"                  response.Write "        <td colspan=2 align=center>"
                                                                        call
muestra_boton("Olvide mi clave","olvide_clave();")              response.Write "
         "                                                                 call
muestra_boton("Ingresar","ingresar();")         response.Write "                 "
                                                                        call
muestra_boton("Limpiar","limpiar();")
                'Funcion que muestra los botones de operacion
                                'Localizacion = "includes/muestr_boton.asp"
        response.Write "        </td>"                  response.Write " </tr>"
        response.Write "</table>"       end function%><html>    <head>          <title>Login</title>
                <link rel="stylesheet" href="stylesheet/style.css">             <script
src="includes/no_right_mouse.js"></script>      <script language=javascript>            <!--
                function limpiar(){                     //Funcion que limpia la forma de busqueda
                        document.form.numempleado.value = '';
        document.form.idempresa.value = '';
        document.form.idtipo_usuario.value = '';                                document.form.pwd.value = '';
                                document.form.proviene.value = '';
        document.form.action = 'login.asp?login=login';
        document.form.submit();                 }                       function ingresar(){
                        //Funcion que valida los datos y eventualmente
        //direcciona al usuario a la aplicacion
        if(document.form.numempleado.value==''){                                alert('Favor de
ingresar su numero de empleado');                               document.form.numempleado.focus();
                                return false;                                   }
                if(isNaN(document.form.numempleado.value)){
        alert('El numero de empleado debe ser numerico');
        document.form.numempleado.focus();                                      return false;
                        }                               if(document.form.idempresa.value==0){
                                alert('Favor de seleccionar una empresa');
        document.form.idempresa.focus();                                        return false;
                }                               if(document.form.idtipo_usuario.value==0){
                        alert('Favor de seleccionar un tipo de usuario');
        document.form.idtipo_usuario.focus();                                   return false;
                        }                               if(document.form.pwd.value==''){
                        alert('Favor de ingresar una clave de acceso');
        document.form.pwd.focus();                                              return false;
                }                               document.form.proviene.value = 'IN';
                document.form.action = 'login.asp?login=login';
```

```
        document.form.submit();                 }                       function olvide_clave(){
                        //Funcion que manda a llamar una funcion para que                       //al
usuario se le envie un correo electronico en caso                       //de que se le haya olvidado
su clave de acceso                      if(document.form.numempleado.value==''){
                        alert('Favor de ingresar su numero de empleado');
        document.form.numempleado.focus();                                      return false;
                        }                               if(isNaN(document.form.numempleado.value)){
                                alert('El numero de empleado debe ser numerico');
                        document.form.numempleado.focus();
        return false;                           }
        if(document.form.idempresa.value==0){                                   alert('Favor de
seleccionar una empresa');                                      document.form.idempresa.focus();
                                return false;                           }
        if(document.form.idtipo_usuario.value==0){                              alert('Favor de
seleccionar un tipo de usuario');                               document.form.idtipo_usuario.focus();
                                return false;                           }
                numempleado = document.form.numempleado.value;
        idempresa = document.form.idempresa.value;                              idtipo_usuario =
document.form.idtipo_usuario.value;                             document.form.action =
'cambio_password/cambi_olvid_passw.asp?idempresa='+idempresa+'&numempleado='+numempleado+'&idtip
o_usuario='+idtipo_usuario+'&login=login'                       document.form.submit();
        }                       function actualizar_explorador(){
        document.form.action = 'navigator/navig_updat.asp';
        document.form.submit();                 }                       function browser_detection(){
                        //Funcion que detecta la version del navegador
        //Se requiere IE 6.0 +                          var nNam = navigator.appName;
                var nVer = navigator.appVersion;                        var nAgt =
navigator.userAgent;                            var fullVersion = parseFloat(nVer);
        var majorVersion = parseInt(nVer);                              // In Internet Explorer, the true version
is after "MSIE"                         if(nNam!='Microsoft Internet Explorer'){
                actualizar_explorador();                                }
        else{                                   if ((verOffset=nAgt.indexOf("MSIE"))!=-1) {
                                fullVersion = parseFloat(nAgt.substring(verOffset+5,nAgt.length));
                                majorVersion = parseInt(''+fullVersion);
                        if(fullVersion<6.00){
        actualizar_explorador();                                        }
                }                               }                               // In Opera, the true
version is after "Opera"                        //if ((verOffset=nAgt.indexOf("Opera"))!=-1) {
                // fullVersion = parseFloat(nAgt.substring(verOffset+6,nAgt.length));
        // majorVersion = parseInt(''+fullVersion);                     //}                       }
                -->             </script></head> <body Onload=javascript:browser_detection();>
        <form name="form" method="post">                <input type=hidden name=proviene
        value=<%=proviene%>>            <%                      if proviene = "IN" then
                if
autentica(request("idempresa"),request("numempleado"),request("idtipo_usuario"),request("pwd")) then
                        else                                            mensaje = "Datos no Validos"
                        end if                          end if                          call
main_function(mensaje)                          'Funcion que da al usuario la opcion de anexar un
archivo a la solicitud                          'Localizacion = esta misma pagina       %>
        </form>  </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../flujo_autoriza/flujo_panta_autor.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 7 de julio 2007       'Funcionalidad
general                 : Pagina mediante la cual se autorizan          '
                                las solicitudes%><!--   #include
file="../connect/conne_open.asp"--><!--   Este archivo contiene la conexion a la BD--><!--   #include
file="../includes/forma_fecha.asp"--><!--   Este archivo incluye la funcion que da formato a las fechas--><%
```

```
'================================================================================
'Estas variables son comunes a todas las paginas dim numempleado 'Numero de empleado que esta firmado en la aplicacion dim idempresa 'Numero de empresa en la que se esta trabajando numempleado = request("numempleado")
idempresa = request("idempresa")
'================================================================================
'================================================================================
'Estas variables se usan nada mas en las paginas de solicitudes
dim folio 'Numero de folio de la solicitud
folio = request("folio")
'================================================================================
'================================================================================
'Variable que define el tipo de operacion dim tipo_operacion
tipo_operacion = request("tipo_operacion")
'================================================================================
'================================================================================
'Variable que define si se presentara la pantalla de autorizaciones con opcion global o no '1 = global, los autorizadores pueden autorizar sin entrar al detalle
'0 = los autorizadores tienen que entrar al detalle de la solicitud para autorizar dim autorizador_global
'================================================================================
function ha_delegado_autoridad(idempresa,numempleado)
'Funcion que obtiene si el usuario ha delegado su autoridad 'Recibe:
'idempresa = empresa en la que se trabaja 'numempleado = empleado que pretende autorizar las solicitudes dim sql dim res sql = "exec [dbo].sp_obtiene_si_empleado_ha_transferido_autoridad "&idempresa&","&numempleado&""
set res = connect.execute(sql) if not res.eof then
ha_delegado_autoridad = res(0) else
ha_delegado_autoridad = 0 end if end function
function obtiene_autorizacion_global(idempresa,numempleado) 'Funcion que obtiene si ese usuario esta facultado para autorizar 'de manera global o no
'Recibe: 'idempresa = empresa en la que se trabaja
'numempleado = empleado que pretende autorizar las solicitudes dim sql
dim res sql = "select [dbo].fn_obtiene_autorizacion_perfil("&idempresa&","&numempleado&",'a')" set res = connect.execute(sql) if not res.eof then
obtiene_autorizacion_global = res(0) else
obtiene_autorizacion_global = 0 end if end function
autorizador_global = obtiene_autorizacion_global(idempresa,numempleado) function empleado_juega_rol(idempresa,numempleado,idstatus) 'Funcion que obtiene que tipo de autorizador es el usuario 'es decir, si es autorizador de ppto, es jefe, contabilidad, etc. 'Recibe: 'idempresa = empresa en la que se esta trabajando
'numempleado = empleado que pretende autorizar 'idstatus = status que se va a evaluar para ver si el usuario cumple ese rol dim sql dim res
sql = "exec [dbo].sp_flujo_autoriza_empleado_juega_rol "&idempresa&","&numempleado&",'"&idstatus&"'" set res = connect.execute(sql)
if not res.eof then empleado_juega_rol = res(0)
else empleado_juega_rol = 0 end if end function
function obtiene_folios(idempresa,numempleado,status,tipo_operacion)
'Funcione que obtiene los folios pendientes por autorizar 'Recibe:
'idempresa = empresa en la que se esta trabajando 'numempleado = empleado que esta haciendo las autorizaciones 'status = status de la solicitud a autorizar
'tipo_operacion = tipo de solicitud a autorizar (anticipo,pago,compra,etc)
dim sql dim res dim arRes dim t_arRes
dim count 'Response.Write sql & "<br>"
set res = connect.execute(sql) if not res.eof then
arRes = res.getrows t_arRes = ubound(arRes,2)
else t_arRes = -1 end if if
```

```
t_arRes >= 0 then                                   response.Write "<table border=""0"" width=""100%"">"
                                    response.Write " <tr class=tabla_nivel_2>"
        response.Write "            <td colspan=10>"&tipo_operacion&"</td>"
        response.Write "    </tr>"                          response.Write " <tr class=tabla_nivel_3
align=center>"                      response.Write "            <td>Folio</td>"
        response.Write "            <td>Beneficiario</td>"                  response.Write "
        <td>Monto</td>"                     response.Write "            <td>Fecha de solicitud</td>"
                                    if autorizador_global = 1 then
        response.Write " <td>Autorizar</td>"                            response.Write "
        <td>Rechazar</td>"                              response.Write "
        <td>Comentarios</td>"                       end if
        response.Write " <td>Ver detalle</td>"                  if autorizador_global = 1 then
                                    response.Write " <td colspan=2>¿C&oacute;mo se ver&iacute;a
si....</td>"                        end if                  response.Write " </tr>"
                        for count = 0 to t_arRes                        response.Write "<tr
align=center>"                      response.Write " <td
class=texto_nivel_2>"&arRes(0,count)&"</td>"                            response.Write " <td
class=texto_nivel_2>"&arRes(1,count)&"</td>"                            response.Write " <td
class=texto_nivel_2>"&formatcurrency(arRes(2,count),2)&"</td>"
        response.Write " <td class=texto_nivel_2>"&formato_fecha(arRes(3,count))&"</td>"
                        if autorizador_global = 1 then
        response.Write "<td><input type=checkbox
value=aMJ"&tipo_operacion&"MJ"&arRes(0,count)&"MJ"&status&"
name=registraMJaMJ"&tipo_operacion&"MJ"&arRes(0,count)&"
onclick=javascript:evaluar_aut_o_rec('a','"&tipo_operacion&"',"&arRes(0,count)&");></td>"
                                    response.Write "<td><input type=checkbox
value=rMJ"&tipo_operacion&"MJ"&arRes(0,count)&"MJ"&status&"
name=registraMJrMJ"&tipo_operacion&"MJ"&arRes(0,count)&"
onclick=javascript:evaluar_aut_o_rec('r','"&tipo_operacion&"',"&arRes(0,count)&");></td>"
                                    response.Write "<td><input type=textbox class=caja_texto
name=comentMJ"&tipo_operacion&"MJ"&arRes(0,count)&"></td>"                                  end
if                                  response.Write " <td class=texto_nivel_2>"
                response.Write "            <a
href=""javascript:ver_detalle('"&tipo_operacion&"',"&arRes(0,count)&",'"&status&"')"">Ver detalle</a>"
                                    response.Write " </td>"                                 if
autorizador_global = 1 then                                 response.Write "<td
class=texto_nivel_2><a
href=javascript:open_window('flujo_prono_movim.asp?tipo_operacion="&tipo_operacion&"&numempleado="&
numempleado&"&idempresa="&idempresa&"&folio="&arRes(0,count)&"&status="&status&"&opcion=a','title','sc
rollbars=yes,status=no,height=400,width=700,resizable=yes');>...autorizo?</a></td>"
                        response.Write "<td class=texto_nivel_2><a
href=javascript:open_window('flujo_prono_movim.asp?tipo_operacion="&tipo_operacion&"&numempleado="&
numempleado&"&idempresa="&idempresa&"&folio="&arRes(0,count)&"&status="&status&"&opcion=r','title','sc
rollbars=yes,status=no,height=400,width=700,resizable=yes');>...rechazo?</a></td>"
                end if                                  response.Write "</tr>"
        next                        response.Write "</table>"               else
        end if          end function            function
obtiene_tipos_operacion(idempresa,numempleado,status,nombre_status)                 'Funcione
que obtiene los tipos de operacion o solicitud disponibles para la empresa en cuestion
        'Recibe:                    'idempresa = empresa en la que se esta trabajando
        'numempleado = empleado que esta haciendo las autorizaciones                'status =
status de la solicitud a autorizar                  'nombre_status = nombre del status para presentarlo
                dim sql                 dim res                 dim arRes                   dim
t_arRes                 dim count                   sql = "exec
[dbo].sp_obtiene_tipos_de_operacion "&idempresa&", 'u' "                set res = connect.execute(sql)
                        if not res.eof then                     arRes = res.getrows
                t_arRes = ubound(arRes,2)                       else
        t_arRes = -1                    end if                  if t_arRes >= 0 then
                if empleado_juega_rol(idempresa,numempleado,status) = 1 then
                response.Write "<table border=0 width=""100%"">"
        response.Write " <tr>"                                  response.Write "            <td
class=tabla_nivel_1>"&nombre_status&"</td>"                             response.Write "
```

```
        </tr>"                                          for count = 0 to t_arRes
                response.Write "<tr>"                                                   response.Write "
        <td>"
        call obtiene_folios(idempresa,numempleado,status,arRes(1,count))
                                                                'Funcion que obtiene los folios
pendientes por autorizar para este tipo de operacion
                                                'Localizacion = esta misma pagina
                response.write    "         </td>"
        response.Write "</tr>"                                  next
        response.Write "</table>"                       end if                  end if          end
function        function obtiene_flujo(idempresa,numempleado)                   'Funcione que
obtiene el flujo de autorizacion de la empresa en cuestion                      'Recibe:
        'idempresa = empresa en la que se esta trabajando                       'numempleado = empleado
que esta haciendo las autorizaciones                    'status = status de la solicitud a autorizar
                'nombre_status = nombre del status para presentarlo             dim sql
        dim res                 dim arRes               dim t_arRes             dim
count                   sql = "exec [dbo].sp_flujo_autoriza_obtiene_flujo_autorizacion "&idempresa&""
                set res = connect.execute(sql)                  if not res.eof then
        arRes = res.getrows                             t_arRes = ubound(arRes,2)
        else                            t_arRes = -1            end if                  if
t_arRes >= 0 then                       response.Write "<table border=""0"" align=center
width=""90%"">"                         for count = 0 to t_arRes
        response.Write "<tr>"                                   response.Write "  <td>"
                                                                call
obtiene_tipos_operacion(idempresa,numempleado,arRes(0,count),arRes(1,count))
                                                                'Funcion que obtiene los tipos de
operacion para esta empresa
                'Localizacion = esta misma pagina                               response.write    "
        </td>"                                  response.Write "</tr>"                  next
                                if autorizador_global = 1 then
        response.Write "<tr>"                                   response.Write "  <td
align=center><input type=button class=boton value=""registrar movimientos""
onclick=javascript:registrar();></td>"                                  response.Write "</tr>"
                                end if                  response.Write "</table>"               end
if              end function%><html>    <head>          <title>Autorizaciones</title>
        <script src="../includes/open_windo.js"></script>               <script
src="../includes/no_right_mouse.js"></script>           <link rel="stylesheet"
href="../stylesheet/style.css">         <script language=javascript>            <!--
        function ver_detalle(tipo_operacion,folio,status){                      //Esta funcion llama a
la pagina a traves de la cual se ve el detalle de la solicitud                  //Recibe:
                        //tipo_operacion = tipo de operacion (anticipo,pago,compra,etc)
                //folio = folio de la solicitud                 //status = status actual de la solicitud
                        document.form.folio.value = folio;
        document.form.tipo_operacion.value = tipo_operacion;
        document.form.status.value = status;                            document.form.action =
'../solicitud/solic_ver_detal.asp?ver_menu=1';                          document.form.submit();
                }                       function registrar(form){                               //Esta
funcion llama a la pagina que propiamente registra la autorizacion o el rechazo                 //de
las solicitudes                         //Se evalua cuantas solicitudes pueden ser procesadas a la vez
                        //Maximo 20 se pueden procesar
        cajas_checadas = 0;                             count = document.form.elements.length;
                        permitidas = 20;                        var etiqueta;
        for (i=0;i<count;i++){                                  if(document.form.elements[i].checked
== true){                               cajas_checadas = cajas_checadas + 1;
                        }                               }
        if(cajas_checadas>permitidas){                                  deshabilitar =
eval(cajas_checadas-permitidas);                                        if(deshabilitar > 1){
                                etiqueta = 'solicitudes'                                        }
                                else{                                                   etiqueta = 'solicitud'
                                }                                                       alert('No se pueden
tramitar mas de ' +permitidas+ ' solicitudes a la vez\nFavor de deshabilitar ' +deshabilitar+ ' ' + etiqueta);
                                return false;                                   }
```

```
        document.form.action = 'flujo_regis_autor.asp'
        document.form.submit();                 }                       function
evaluar_aut_o_rec(aut_o_rec,tipo_operacion,folio){                      //Esta funcion selecciona o
deselecciona las diferentes opciones de                 //autorizacion o rechazo depende de
cual se seleccione                      //Es decir, "checa" los checkboxes a los que se les de
clic                    var cadena_a_evaluar_aut =
'document.form.registraMJaMJ'+tipo_operacion+'MJ'+folio+";                      var
cadena_a_evaluar_rec = 'document.form.registraMJrMJ'+tipo_operacion+'MJ'+folio+";
        if(aut_o_rec=='a'){
        if(eval(cadena_a_evaluar_aut).checked==true){
        eval(cadena_a_evaluar_aut).checked = true;
        eval(cadena_a_evaluar_rec).checked = false;                                     }
                        else{
        eval(cadena_a_evaluar_aut).checked = false;
        eval(cadena_a_evaluar_rec).checked = false;                                     }
                }                               else{
        if(eval(cadena_a_evaluar_rec).checked==true){
        eval(cadena_a_evaluar_aut).checked = false;
        eval(cadena_a_evaluar_rec).checked = true;                                      }
                        else{
        eval(cadena_a_evaluar_aut).checked = false;
        eval(cadena_a_evaluar_rec).checked = false;                                     }
                }               }                       function transferencia(){
                document.form.action = 'flujo_trans_autor.asp';
        document.form.submit();                 }               -->             </script></head>
        <body>  <!--#include file=../menu/menu.asp-->          <form name="form" method="post">
        <input type=hidden name=idempresa               value=<%=idempresa%>>          <input
type=hidden name=numempleado            value=<%=numempleado%>>             <input type=hidden
name=folio                      value=<%=folio%>>               <input type=hidden
name=tipo_operacion     value=<%=tipo_operacion%>>              <input type=hidden name=status
                value="">               <input type=hidden name=proviene        value=AU>
        <!--Este campo especifica de donde proviene esta llamada al detalle     AU = Proviene de la
pagina de AUtorizaciones-->             <input type=hidden name=tipo_autorizacion value=GL>
        <!--Este campo especifica de donde si la autorizacion es global(GL) o individual(IN)-->   <%      if
ha_delegado_autoridad(idempresa,numempleado) then               response.Write "<table align=center
width=""90%"">"         response.Write "  <tr>"         response.Write "        <td class=aviso
align=center>"          response.Write "                No puede autorizar ya que ha transferido su
autoridad, para controlar la transferencia haga clic <a href=javascript:transferencia();>AQUI</a>"
        response.Write "        </td>"          response.Write "  </tr>"          response.Write
"</table>"      else            call obtiene_flujo(idempresa,numempleado)       'Funcion que obtiene
el flujo de autorizacion de esta empresa        'Localizacion = esta misma pagina end if    %>
        </form>  </body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../flujo_autoriza/flujo_regis_autor.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 7 de julio 2007       'Funcionalidad
general                 : Pagina que registra las autorizaciones y los rechazos%>      <!--    #include
file="../connect/conne_open.asp"--><!--     Este archivo contiene la conexion a la BD--><%
        '=============================================================================
==============      'Estas variables son comunes a todas las paginas    dim numempleado
                                        'Numero de empleado que esta firmado en la aplicacion
        dim idempresa                                   'Numero de empresa en la que
se esta trabajando                      numempleado = request("numempleado")    idempresa =
request("idempresa")
        '=============================================================================
==============      dim folio                       'Folio de la solicitud      dim status
                                'Status que se autoriza         dim opcion
                'Opcion de autorizacion o rechazo (a = aut, r = rechazo)     dim tipo_operacion
```

```
                    'Tipo de operacion o solicitud      dim comentario
        'Comentario acerca la autorizacion o el rechazo     dim sql
        'Variable donde se almacena la sentencia sql        dim tipo_autorizacion               'Variable que
almacena el tipo de autorizacion (GL= Global, IN = Individual)dim objitem
        'Objeto dentro de la forma dim arreglo_split_registra  '        dim arreglo_split_valor             '
                tipo_autorizacion = request("tipo_autorizacion")            tipo_operacion =
ucase(trim(request("tipo_operacion")))      numempleado = request("numempleado")    idempresa =
request("idempresa")      folio = request("folio")      status = request("status")  comentario =
ucase(left(replace(request("comentario"),"'",""),250)) opcion = request("opcion")                   function
registra_movimiento(folio,numempleado,idempresa,status,comentario,opcion)           'Funcion que registra
el movimiento de autorizacion o rechazo             sql = "exec sp_flujo_autoriza_registra_autorizacion
"&folio&","&numempleado&","&idempresa&",'"&status&"','"&comentario&"','"&opcion&"',1"
        connect.execute(sql)                                end function                    'Si
la autorizacion es GLobal entonces, se hace lo siguiente:     if tipo_autorizacion = "GL" then          'Se
busca en todos los objetos de la forma              for each objItem in request.form
        arreglo_split_registra = split(objitem,"MJ")                'Los objetos que tengan dentro de su
nombre la palabra "REGISTRA"                if trim(ucase(arreglo_split_registra(0))) = "REGISTRA"
then                        arreglo_split_valor = split(request(objitem),"MJ")
        opcion = arreglo_split_valor(0)                     tipo_operacion = arreglo_split_valor(1)
                        folio = arreglo_split_valor(2)                          status =
arreglo_split_valor(3)                      comentario =
left(replace(request("coment MJ"&tipo_operacion&"MJ"&folio),"'"," "),250)                       call
registra_movimiento(folio,numempleado,idempresa,status,comentario,opcion)
        'Funcion que registra la autorizacion o el rechazo                  'Localizacion = esta
misma pagina                                        'Esta funcion se manda llamar
tantas veces como checkboxes se hayan seleccionado
        end if          next                        else        'Si la autorizacion es de tipo INdividual,
es decir, viene del detalle de la solicitud         'se manda llamar la funcion que registra la autorizacion o
el rechazo UNA SOLA VEZ             call
registra_movimiento(folio,numempleado,idempresa,status,comentario,opcion)       'Funcion que registra
la autorizacion o el rechazo            'Localizacion = esta misma pagina               end if
                response.Redirect
"flujo_panta_autor.asp?idempresa="&idempresa&"&numempleado="&numempleado&""       'Se
redirecciona a la pagina de autorizaciones          %>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../menu/menu_data.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<!--    #include file=../includes/menu_objet_exist.asp-->
<!--    Este archivo contiene la funcion que evalua si un usuario tiene acceso a cierto objeto del menu-->

<%
'TIPOS DE OPERACION
Const Anticipos                         = 1
Const Comprobacion_Gastos       = 2
Const Compras                           = 3
Const Pagos                                     = 4
Const Reservaciones                     = 5
Const Contratos                         = 6

'OPCIONES
Const Reporte_Autorizaciones    = 7
Const Cambio_Clave                      = 8

'AUTORIZACIONES
Const Autorizaciones            = 9
Const Transferencia                     = 10


'REPORTES
```

```
Const Reporte_PPTO                                  = 32
Const Reporte_Conciliacion                  = 33
%>

<script>
fixMozillaZIndex=true; //Fixes Z-Index problem with Mozilla browsers but causes odd scrolling problem, toggle to see if it helps
_menuCloseDelay=500;
_menuOpenDelay=150;
_subOffsetTop=2;
_subOffsetLeft=-2;

with(menuStyle=new mm_style()){
bordercolor="#296488";
borderstyle="solid";
borderwidth=1;
fontfamily="Verdana, Tahoma, Arial";
fontsize="75%";
fontstyle="normal";
headerbgcolor="#ffffff";
headercolor="#000000";
offbgcolor="#C1CDCD";
offcolor="#104E8B";
onbgcolor="#104E8B";
oncolor="#ffffff";
outfilter="randomdissolve(duration=0.0)";
overfilter="Fade(duration=0.2);Alpha(opacity=90);Shadow(color=#777777', Direction=135, Strength=5)";
padding=5;
pagebgcolor="#82B6D7";
pagecolor="black";
separatorcolor="#2D729D";
separatorsize=1;
subimage="../menu/arrow.gif";
subimagepadding=2;
}
with(milonic=new menuname("Solicitudes")){
overflow="scroll";
style=menuStyle;
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Anticipos,"nu") = 1 and ExisteFolder(server.MapPath("\TRAVEX\anticipo\")) then%>
aI("text=Anticipo;showmenu=Comprobacion Anticipo;url=../solicitud/solic_grale_captu.asp?tipo_operacion=anticipo&idempresa=<%=session("idempresa")%>&numempleado=<%=session("numempleado")%>;status=Solicitud de Anticipo")
<%end if%>
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Comprobacion_Gastos,"nu") = 1 then%>
aI("text=Comprobacion de Gastos;url=../solicitud/solic_grale_captu.asp?tipo_operacion=comprobacion_gasto&idempresa=<%=session("idempresa")%>&numempleado=<%=session("numempleado")%>;status=Solicitud de Comprobacion de Gastos")
<%end if%>
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Compras,"nu") = 1 and ExisteFolder(server.MapPath("\TRAVEX\compra\")) then%>
aI("text=Compras;url=../solicitud/solic_grale_captu.asp?tipo_operacion=compra&idempresa=<%=session("idempresa")%>&numempleado=<%=session("numempleado")%>;status=Solicitud de Compra")
<%end if%>
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Pagos,"nu") = 1 and ExisteFolder(server.MapPath("\TRAVEX\pago\")) then%>
aI("text=Pagos a Proveedores;url=../solicitud/solic_grale_captu.asp?tipo_operacion=pago&idempresa=<%=session("idempresa")%>&numempleado=<%=session("numempleado")%>;status=Solicitud de Pago a Proveedores")
<%end if%>
```

```
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Reservaciones,"nu") = 1 and
ExisteFolder(server.MapPath("\TRAVEX\reservacion\")) then%>
aI("text=Reservacion;showmenu=Comprobacion
Reservacion;url=../solicitud/solic_grale_captu.asp?tipo_operacion=reservacion&idempresa=<%=session("idem
presa")%>&numempleado=<%=session("numempleado")%>;status=Solicitud de Reservacion")
<%end if%>
}

with(milonic=new menuname("Comprobacion Anticipo")){
style=menuStyle;
<%if ExisteFolder(server.MapPath("\TRAVEX\comprobacion_anticipo\")) then%>
aI("text=Comprobacion de
Anticipo;url=../solicitud/solic_grale_captu.asp?tipo_operacion=comprobacion_anticipo&idempresa=<%=sessio
n("idempresa")%>&numempleado=<%=session("numempleado")%>;status=Solicitud de Comprobacion de
Anticipo")
<%end if%>
}

with(milonic=new menuname("Comprobacion Reservacion")){
style=menuStyle;
<%if ExisteFolder(server.MapPath("\TRAVEX\comprobacion_reservacion\")) then%>
aI("text=Comprobacion de
Reservacion;url=../solicitud/solic_grale_captu.asp?tipo_operacion=comprobacion_reservacion&idempresa=<%
=session("idempresa")%>&numempleado=<%=session("numempleado")%>;status=Solicitud de
Comprobacion de Reservacion")
<%end if%>
}

with(milonic=new menuname("Opciones")){
style=menuStyle;
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Cambio_Clave,"nu") = 1
then%>
aI("text=Cambio de Clave de
Acceso;url=../cambio_password/cambi_passw.asp?idempresa=<%=session("idempresa")%>&numempleado=
<%=session("numempleado")%>;status=Cambio de Clave de Acceso")
<%end if%>
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Reporte_Autorizaciones,"nu") =
1 then%>
aI("text=Reporte de
Autorizaciones;url=../reporte_autoriza/repor_autor_usuar.asp?idempresa=<%=session("idempresa")%>&nume
mpleado=<%=session("numempleado")%>;status=Reporte de Autorizaciones")
<%end if%>
}

with(milonic=new menuname("Autorizaciones")){
style=menuStyle;
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Autorizaciones,"nu") = 1
then%>
aI("text=Autorizaciones;url=../flujo_autoriza/flujo_panta_autor.asp?idempresa=<%=session("idempresa")%>&n
umempleado=<%=session("numempleado")%>;status=Pantalla de Autorizaciones")
<%end if%>
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Transferencia,"nu") = 1 then%>
aI("text=Transferencia de
Autoridad;url=../flujo_autoriza/flujo_trans_autor.asp?idempresa=<%=session("idempresa")%>&numempleado=
<%=session("numempleado")%>;status=Transferencia de Autoridad")
<%end if%>
}

with(milonic=new menuname("Reportes")){
style=menuStyle;
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Reporte_Conciliacion,"nu") = 1
then%>
```

```
aI("text=Reporte de Conciliacion de
Aplicaciones;url=../solicitud/solic_grale_captu.asp?tipo_operacion=comprobacion_gasto&idempresa=<%=sess
ion("idempresa")%>&numempleado=<%=session("numempleado")%>;status=Reporte de Conciliacion de
Aplicaciones")
<%end if%>
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Reporte_PPTO,"nu") = 1
then%>
aI("text=Reporte de
Presupuesto;url=../solicitud/solic_grale_captu.asp?tipo_operacion=comprobacion_gasto&idempresa=<%=sess
ion("idempresa")%>&numempleado=<%=session("numempleado")%>;status=Reporte de Presupuesto")
<%end if%>
}



with(milonic=new menuname("Usuarios")){
style=menuStyle;
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Usuarios_externos,"nu") = 1
then%>
aI("text=Usuarios
Externos;url=../catalogo/catal_usuar_exter.asp?idempresa=<%=session("idempresa")%>;status=Catalogo de
Usuarios Externos")
<%end if%>
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Usuarios_internos,"nu") = 1
then%>
aI("text=Usuarios
Internos;url=../catalogo/catal_usuar_inter.asp?idempresa=<%=session("idempresa")%>;status=Catalogo de
Usuarios Internos")
<%end if%>
}

with(milonic=new menuname("Kilometros")){
style=menuStyle;
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),Costo_Kilometro,"nu") = 1
then%>
aI("text=Costo por
Kilometro;url=../catalogo/catal_costo_kilom.asp?idempresa=<%=session("idempresa")%>;status=Costo por
Kilometro")
<%end if%>
<%if obtiene_menu_especifico(session("idempresa"),session("numempleado"),KilometrajeS,"nu") = 1 then%>
aI("text=Origenes y
Destinos;url=../catalogo/catal_kilom.asp?idempresa=<%=session("idempresa")%>;status=Catalogo de
Origenes y Destinos")
<%end if%>
}

drawMenus();
</script>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../menu/menu.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      function obtiene_menu_gral(idempresa,numempleado)                   dim sql             dim res
        dim arRes                   dim t_arRes                 dim count                   sql = "exec
[dbo].sp_menu_opciones_principales "&idempresa&", "&numempleado&""              set res =
connect.execute(sql)                if not res.eof then                     arRes = res.getrows
        t_arRes = ubound(arRes,2)                   else                    t_arRes = -1                end
if                  if t_arRes >= 0 then                    for count = 0 to t_arRes
        response.Write "aI(""text="&trim(arRes(1,count))&";showmenu="&trim(arRes(1,count))&";"");"
                next                end if     end function        function
obtiene_nombre(idempresa,numempleado)           dim sql             dim res             sql = "select
```

```
[dbo].fn_obtiene_nombre("&idempresa&","&numempleado&")"          set res = connect.execute(sql)
        if not res.eof then                 obtiene_nombre = trim(res(0))               else
        obtiene_nombre = "No definido"          end if  end function    function
obtiene_centro_costo(idempresa,numempleado)             dim sql         dim res         sql = "select
[dbo].fn_obtiene_centro_costo("&idempresa&","&numempleado&")"           set res = connect.execute(sql)
                if not res.eof then             obtiene_centro_costo = trim(res(0))         else
                obtiene_centro_costo = "No definido"            end if  end function    function
obtiene_puesto(idempresa,numempleado)           dim sql         dim res         sql = "select
[dbo].fn_obtiene_puesto("&idempresa&","&numempleado&")"          set res = connect.execute(sql)
        if not res.eof then                 obtiene_puesto = trim(res(0))               else
        obtiene_puesto = "No definido"          end if  end function    function
obtiene_logo(idempresa)         dim sql         dim res         sql = "select
[dbo].fn_obtiene_logo("&idempresa&")"           set res = connect.execute(sql)          if not res.eof
then                    obtiene_logo = res(0)           else                    obtiene_logo = ""
        end if  end function%><script type="text/javascript" src="../menu/milonic_src.js"></script><script
        type="text/javascript">if(ns4)_d.write("<scr"+"ipt type=text/javascript
src=../menu/mmenuns4.js><\/scr"+"ipt>");  else _d.write("<scr"+"ipt type=text/javascript
src=../menu/mmenudom.js><\/scr"+"ipt>");</script><!--#include file=../menu/menu_data.asp--><table
width=89% border=0 style="border:1px solid #aaaaaa">    <tr >          <td align=left width=1%>
                <img src=<%=application("dir_logica_emp")%><%=obtiene_logo(session("idempresa"))%>>
                </td>           <td align=left>                 <table border=0 width=100%
cellSpacing=0 cellPadding=7>                    <tr class=texto_nivel_2>
                <td>Nombre: <%=obtiene_nombre(session("idempresa"),session("numempleado"))%></td>
                                        <td>Centro de costos:
<%=obtiene_centro_costo(session("idempresa"),session("numempleado"))%></td>
        </tr>                           <tr class=texto_nivel_2>
        <td>Puesto: <%=obtiene_puesto(session("idempresa"),session("numempleado"))%></td>
                        </tr>                   <tr>                                    <td
align=center colspan=2>                 <script>
                with(milonic=new menuname("Main_Menu")){
        style=menuStyle;                        top=155;
                left=200;                       alwaysvisible=1;
                        orientation="horizontal";                                       overfilter="";
                                                position="relative";
                aI("text=MILONIC;url=http://www.google.com;");
        aI("text=INICIO;url=../inicio/inicio_inici.asp");                               <%
                                                call
obtiene_menu_gral(session("idempresa"),session("numempleado"))
        %>                                      aI("text=Terminar
Sesion;url=../logout/logout.asp");                                      }
                        drawMenus();                                    </script>
                        </td>                   </tr>           </table>
        </td>   </tr></table>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_costo_kilom.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -               'Fecha de ultima modificacion       : 26 de julio 2007          'Funcionalidad
general                 : Pagina mediante la cual se dan de alta, modifican y dan de baja               '
                                                        conceptos de gasto%>            <!--
        #include file="../connect/conne_open.asp"--><!--     Este archivo contiene la conexion a la BD--><!--
        #include file="../includes/muestr_texto.asp"--><!--  Este archivo incluye la funcion que presenta una
caja de texto-->        <!--    #include file="../includes/catal_moned.asp"--><!--  Este archivo incluye
la funcion que presenta el catalogo de monedas--><!--       #include file="../includes/muestr_boton.asp"--
><!--  Este archivo incluye la funcion que presenta los botones-->                     <%

        '=================================================================================
==============          'Estas variables son comunes a todas las paginas            dim
numempleado                                             'Numero de empleado que esta firmado
```

```
en la aplicacion		dim idempresa 'Numero de empresa en la que se esta trabajando numempleado = request("numempleado")		idempresa = request("idempresa")
	'================================================================================================
	'================================================================================================
	'Esta variable almacena la proveniencia de la peticion de las paginas
	dim proviene		'Variable que indica de donde viene la peticion de esta pagina
				'IA = Insertar una nueva solicitud que viene de la pagina anterior
		proviene =	request("proviene")

	'================================================================================================
	'================================================================================================
	'Variables exclusivas de esta pagina
	dim idmoneda		idmoneda = request("idmoneda") dim costo
		costo = request("costo")
	'================================================================================================
function guarda(idempresa,idmoneda,costo,accion)
		dim sql sql = "exec [dbo].sp_costo_kilometro_inserta_modifica "&idempresa&","&idmoneda&","&costo&",'"&accion&"'"
		connect.execute(sql) idmoneda = 0 costo = 0
		end function function modifica(idempresa,idmoneda)
		dim sql dim res sql = "exec [dbo].sp_costo_kilometro_catalogo "&idempresa&","&idmoneda&"" set res = connect.execute(sql) if not res.eof then idmoneda = res(0)
		costo = round(res(1),2) else
	idmoneda = 0 costo = 0 end if
			end function function main_function(idempresa,proviene)
		'Funcion que presenta al usuario la posibilidad de agregar un concepto de gasto
		'Son los campos necesarios para su captura
	'Recibe		'idempresa = empresa en la que se esta trabajando
	'proviene = variable que indica a la funcion qe accion la mando llamar, nada mas se usa para determinar el boton ' que se presenta (MD = Modificar concepto, GC = Guardar Concepto) response.Write "<table border=0 width=""90.5%"" align=center>" response.Write " <tr>"
	response.Write "	<td class=tabla_nivel_2 colspan=2>Costo por kilometro</td>"
	response.Write " </tr>" response.Write "<table>"
		response.Write "<table border=0 width=""90%"" align=center class=contenedor>" response.Write " <tr>" response.Write "
	<td>" call muestra_texto("Costo por kilometro $","costo",costo,"texto_nivel_2",5,7)
		'Funcion que muestra el catalogo de cuentas contable
		'Localizacion
"../includes/catal_cuent_conta.asp"
		response.Write "	</td>" response.Write "
	</tr>" response.Write " <tr>" response.Write "
	<td>" call catalogo_moneda("Moneda","idmoneda",idmoneda,"texto_nivel_2","")
		'Funcion que muestra el campo de texto para introducir el nombre del concepto 'Localizacion
= "../includes/muestr_texto.asp" response.Write "	</td>"
	response.Write " </tr>" response.Write "</table>"
		response.Write "<table border=0 width=""90.5%"" align=center>"
	response.Write " <tr>" response.Write "	<td colspan=3 align=center>" call muestra_boton("Guardar Para esta moneda","guardar('i');") response.Write "
	 " call muestra_boton("Guardar para todas las monedas","guardar('m');") response.Write "
		 "
		call muestra_boton("Imprimir","window.print();")
```

```
                                                                                response.Write "
                 "
                        call muestra_boton("Limpiar","limpiar();")                      response.Write "
        </td>"                  response.Write " </tr>"                 response.Write "<table>"
                        end function                                    function
ver_costos_existentes(idempresa)        dim sql                 dim res                 dim
arRes                   dim t_arRes             dim count
                sql = "exec [dbo].sp_costo_kilometro_catalogo "&idempresa&",0"                  set
res = connect.execute(sql)              if not res.eof then                     arRes = res.getrows
                        t_arRes = ubound(arRes,2)               else
                t_arRes = -1                    end if                                  if
t_arRes >= 0 then                       response.Write "<table width=""90%"">"
                response.Write "  <tr class=tabla_nivel_3>"
        response.Write "        <td>Costo</td>"                 response.Write "
        <td>Moneda</td>"                        response.Write "        <td>Modificar</td>"
                        response.Write " </tr>"                         for count = 0
to t_arRes                              response.Write "<tr class=texto_nivel_2>"
                        response.Write "  <td>"&formatcurrency(arRes(2,count),2)&"</td>"
                        response.Write "  <td>"&arRes(1,count)&"</td>"
        response.Write "  <td><a href=javascript:modificar("&arRes(0,count)&");>Modificar</a></td>"
                        response.Write "</tr>"                          next
                response.Write "</table>"                       end if
        end function            %><html>        <head>          <title>Catalogo de Costos por
Kilometro</title>       <link rel="stylesheet" href="../stylesheet/style.css">          <script
language=javascript>            <!--                    function limpiar(){
        document.form.proviene.value = '';              document.form.idmoneda.value = '';
                        document.form.costo.value = 0;
        document.form.action = 'catal_costo_kilom.asp'
        document.form.submit();                                         }
                        function selecciona_moneda(){                   }
                        function guardar(accion){
        if(document.form.idmoneda.value==0){                            alert('Favor de
seleccionar un tipo de moneda');                        document.form.idmoneda.focus();
                        return false;                           }
                                        if(isNaN(document.form.costo.value)){
                alert('Valor no valido para el costo');
        document.form.costo.value = 0;                          document.form.costo.focus();
                                                                        return false;
                        }                                       document.form.accion.value =
accion;
        document.form.proviene.value = 'GM';                    document.form.action =
'catal_costo_kilom.asp'                 document.form.submit();                 }
                                function modificar(idmoneda){
        document.form.proviene.value = 'MM';
        document.form.idmoneda.value = idmoneda;                        document.form.action
= 'catal_costo_kilom.asp'               document.form.submit();
                        }                                               -->
        </script></head> <body>  <!--#include file=../menu/menu.asp-->        <form name="form"
method="post" >                 <input type=hidden name=idempresa
        value=<%=idempresa%>>           <input type=hidden name=numempleado
        value=<%=numempleado%>>                 <input type=hidden name=proviene
        value=<%=request("proviene")%>>         <input type=hidden name=accion
        value="">                       <%                      select case ucase(proviene)
                case "GM"       : call guarda(idempresa,idmoneda,costo,request("accion"))
                                        case "MM"       : call modifica(idempresa,idmoneda)
                                                case "EM"       : call
elimina(idempresa,idmoneda)             end select                      call
main_function(idempresa,proviene)       'Funcion principal a traves de la cual se pide o se
muestra al usuario los datos del concepto de gasto              'Localizacion = esta misma pagina
                                call ver_costos_existentes(idempresa)
        'Funcion que presenta los conceptos de gasto existentes                 'Localizacion = esta
misma pagina            %>                      </form>         </body></html>
```

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_tipo_cambi.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
'Fte_ - - 'Fecha de ultima modificacion : 29 de julio 2007 'Funcionalidad general : Pagina que muestra el catalogo de tipos de cambio ' Este catalogo es de SOLO LECTURA, ya que los datos ' se extraen de otro sistema%> <!-- #include file="../connect/conne_open.asp"--><!-- Este archivo contiene la conexion a la BD--><!-- #include file="../includes/forma_fecha.asp"--><!-- Este archivo incluye la funcion que da formato a las fechas--><!-- #include file="../includes/muestr_fecha.asp"--><!-- Este archivo incluye la funcion que presenta las cajas de texto para que se seleccione una fecha--> <!--
#include file="../includes/muestr_boton.asp"--><!-- Este archivo incluye la funcion que presenta los botones--> <%
'==========
========== 'Estas variables son comunes a todas las paginas dim numempleado 'Numero de empleado que esta firmado en la aplicacion dim idempresa 'Numero de empresa en la que se esta trabajando numempleado = request("numempleado") idempresa = request("idempresa")
'==========
========== dim fecha fecha = request("fecha") if fecha = "" then fecha = "0" end if function fecha_tipo_cambio() response.Write "<table border=0 width=""90%"" align=center>" response.Write " <tr>"
response.Write " <td class=tabla_nivel_2 colspan=2>Tipos de cambio</td>"
response.Write " </tr>" response.Write "</table>"
response.Write "<table width=""90%"" align=center class=contenedor>"
response.Write " <tr>" response.Write " <td>" call muestra_fecha("Fecha","fecha","","texto_nivel_2") response.Write " </td>"
response.Write " <td>"
call muestra_boton("Buscar","fecha_tipo_cambio();")
response.Write " </td>"
response.Write " </tr>" response.Write "</table>" end function
function ver_registros_existentes(fecha) dim sql
dim res dim arRes dim t_arRes
dim count sql = "exec [dbo].sp_tipo_cambio_catalogo '"&fecha&"'" set res = connect.execute(sql) if not res.eof then
arRes = res.getrows t_arRes = ubound(arRes,2)
else t_arRes = -1 end if
response.Write "<table align=center width=""90%"" border=0>" if t_arRes >= 0 then
response.Write " <tr class=tabla_nivel_3>"
response.Write " <td>Moneda Origen</td>"
response.Write " <td>Moneda Destino</td>"
response.Write " <td>Fecha</td>" response.Write " <td>Tipo de cambio</td>" response.Write " </tr>"
for count = 0 to t_arRes
response.Write "<tr class=texto_nivel_2>" response.Write " <td>"&arRes(0,count)&"</td>" response.Write " <td>"&arRes(1,count)&"</td>" response.Write " <td>"&formato_fecha(arRes(2,count))&"</td>"
response.Write " <td>"&formatcurrency(arRes(3,count),2)&"</td>"
response.Write "</tr>" next else
response.Write " <tr class=aviso>" response.Write " <td colspan=4 align=center>No se encontro tipo de cambio con la fecha proporcionada</td>"

```
                response.Write " </tr>"                    end if
        response.Write " <tr class=texto_nivel_2>"                    response.Write "        <td
colspan=4><hr></td>"                    response.Write " </tr>"
        response.Write " <tr>"                    response.Write "        <td colspan=4 align=center>"
                                                                                call
muestra_boton("Imprimir","window.print();")                    response.Write "        </td>"
        response.Write " </tr>"                                                response.Write "</table>"
        end function%><html>    <head>          <title>Catalogo de Tipos de Cambio</title>
        <script src="../includes/open_windo.js"></script>              <script
src="../includes/calen_picke.js"></script>                    <link rel="stylesheet"
href="../stylesheet/style.css">                <script language=javascript>                <!--
        function fecha_tipo_cambio(){                                  document.form.action =
'catal_tipo_cambi.asp'                    document.form.submit();                }
                -->                </script></head> <body>  <!--#include file=../menu/menu.asp-->
        <form name="form" method="post">                               <input type=hidden name=idempresa
                value=<%=idempresa%>>                    <input type=hidden name=numempleado
        value=<%=numempleado%>>                    <%                  call
fecha_tipo_cambio()                                    call ver_registros_existentes(fecha)
                'Funcion que presenta los conceptos de gasto existentes                'Localizacion
= esta misma pagina             %>                    </form>         </body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_banco.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion       : 28 de julio 2007          'Funcionalidad
general                 : Pagina que muestra el catalogo de bancos          '
                                        Este catalogo es de SOLO LECTURA, ya que los datos
        '                                                                       se extraen de otro
sistema%>               <!--    #include file="../connect/conne_open.asp"--><!--    Este archivo contiene
la conexion a la BD-->          <!--    #include file="../includes/muestr_boton.asp"--><!--    Este archivo
incluye la funcion que presenta los botones-->                                  <%
        '=============================================================================
==============          'Estas variables son comunes a todas las paginas            dim
numempleado                                                    'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                                  'Numero de
empresa en la que se esta trabajando                                    numempleado =
request("numempleado")          idempresa = request("idempresa")
        '=============================================================================
==============                                                 function ver_registros_existentes()
                dim sql                 dim res                 dim arRes                       dim
t_arRes                 dim count                                       sql = "exec
[dbo].sp_banco_catalogo 'a'"                    set res = connect.execute(sql)                  if
not res.eof then                        arRes = res.getrows                             t_arRes =
ubound(arRes,2)                 else                    t_arRes = -1                    end
if                                      response.Write "<table border=0 width=""90%""
align=center>"                  response.Write " <tr>"                  response.Write "        <td
class=tabla_nivel_2 colspan=2>Bancos</td>"                     response.Write " </tr>"
        response.Write "<table>"                                        if t_arRes >= 0 then
                                        response.Write "<table align=center width=""90%""
border=0>"                      response.Write " <tr class=tabla_nivel_3>"
        response.Write "        <td>Nombre</td>"                        response.Write "
        <td>Tipo</td>"                  response.Write "        <td>Cuenta Nacional</td>"
                                        response.Write "        <td>Cuenta Extranjera</td>"
                response.Write "        <td>Estado</td>"                        response.Write "
        </tr>"                                                  for count = 0 to t_arRes
                                        response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"&arRes(0,count)&"</td>"
        response.Write " <td>"&arRes(1,count)&"</td>"
        response.Write " <td>"&arRes(2,count)&"</td>"
```

```
            response.Write "  <td>"&arRes(3,count)&"</td>"
            response.Write "  <td>"&arRes(4,count)&"</td>"
            response.Write "</tr>"                                  next
            response.Write "  <tr class=texto_nivel_2>"                                 response.Write "              <td
colspan=5><hr></td>"                                  response.Write "  </tr>"
            response.Write "  <tr>"                                 response.Write "              <td colspan=5
align=center>"                                                                                                    call
muestra_boton("Imprimir","window.print();")
            response.Write "              </td>"                                response.Write "  </tr>"

            response.Write "</table>"                           else                                      response.Write
"<table border=0 width=""90%"" align=center>"                                 response.Write "  <tr>"
                        response.Write "              <td class=aviso colspan=2>No hay bancos
definidos</td>"                                 response.Write "  </tr>"
            response.Write "<table>"                                                                      end if
                        end function%><html>      <head>              <title>Catalogo de Bancos</title>
            <script src="../includes/open_windo.js"></script>               <link rel="stylesheet"
href="../stylesheet/style.css">                   <script language=javascript>                        <!--
                                                  -->                 </script></head> <body>  <!--#include
file=../menu/menu.asp-->  <form name="form" method="post">                                    <input type=hidden
name=idempresa              value=<%=idempresa%>>                   <input type=hidden
name=numempleado                    value=<%=numempleado%>>                   <%
            call ver_registros_existentes()                 'Funcion que presenta los conceptos de gasto
existentes                          'Localizacion = esta misma pagina               %>
            </form>               </body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_prove_detal.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
            'Fte_ - -             'Fecha de ultima modificacion       : 29 de julio 2007              'Funcionalidad
general                           : Pagina que muestra el detalle de las empresas                 '
                                                      Este catalogo es de SOLO LECTURA, ya que los datos
                  '                                                                                 se extraen de otro
sistema%>                   <!--        #include file="../connect/conne_open.asp"--><!--       Este archivo contiene
la conexion a la BD-->                  <!--        #include file="../includes/catal_conce_gasto.asp"--><!--
            Este archivo incluye la funcion que muestra el catalogo de conceptos de gasto-->                  <!--
            #include file="../includes/muestr_boton.asp"--><!--    Este archivo incluye la funcion que presenta los
botones-->                                        <%
            '=============================================================================
===============                   'Estas variables son comunes a todas las paginas                dim
numempleado                                                   'Numero de empleado que esta firmado
en la aplicacion              dim idempresa                                                       'Numero de
empresa en la que se esta trabajando                                  numempleado =
request("numempleado")              idempresa = request("idempresa")
            '=============================================================================
===============                                               dim idproveedor             idproveedor
= request("idproveedor")
            '=============================================================================
===============                   'Esta variable almacena la proveniencia de la peticion de las paginas
            dim proviene              'Variable que indica de donde viene la peticion de esta pagina
                                                'IA = Insertar una nueva solicitud que viene de la pagina anterior
                        proviene =          request("proviene")

            '=============================================================================
===============                                               dim mensaje
            function guarda(idempresa,idproveedor,idconcepto,accion)                    dim sql
            dim res                                           sql = "exec
[dbo].sp_proveedor_concepto_inserta_modifica "&idempresa&","&idproveedor&","&idconcepto&",'"&accion&"'"
                                    set res = connect.execute(sql)                      if not res.eof then
```

```
                mensaje = res(0)                else                    mensaje = ""
                end if                                                  end function
                function conceptos_asociados(idempresa,idproveedor,mensaje)                     dim
sql             dim res                 dim arRes                       dim t_arRes
                dim count                                               sql = "exec
[dbo].sp_proveedor_concepto_catalogo "&idempresa&","&idproveedor&""                             set
res = connect.execute(sql)                              if not res.eof then
        arRes = res.getrows                             t_arRes = ubound(arRes,2)
        else                    t_arRes = -1            end if
                if t_arRes >= 0 then                    response.Write "<tr>"
                response.Write " <td class=tabla_nivel_2>Conceptos existentes</td>"
                response.Write " <td class=tabla_nivel_2>Eliminar</td>"
        response.Write "</tr>"
                for count = 0 to t_arRes                response.Write "<tr>"
                        response.Write " <td class=texto_nivel_2_dato>"&arRes(1,count)&"</td>"
                                response.Write " <td>"
        response.Write "        <a href=javascript:elimina("&arRes(0,count)&");>"
                response.Write "                <img src=../images/cruz.gif border=0>"
                response.Write "        </a>"
        response.Write " </td>"                                 response.Write "</tr>"
                                                        next            end if
                end function                                    function
ver_registros_existentes(idempresa,idproveedor)                 dim sql                 dim res
                dim arRes               dim t_arRes             dim count
                                dim identificador               dim nombre
        dim rfc                 dim calle               dim numero                      dim
interior                dim colonia             dim delegacion          dim estado
                dim pais                dim cp                  dim telefono
        dim fax                 dim mail                dim serv_producto                       dim
banco                   dim no_cuenta           dim status
                        sql = "exec [dbo].sp_proveedor_detalle_catalogo
"&idempresa&","&idproveedor&" "         set res = connect.execute(sql)                  if
not res.eof then                        identificador   = res(0)                        nombre
                = res(1)                        rfc             = res(2)
         calle                  = res(3)                        numero          = res(4)
                        interior        = res(5)                        colonia         = res(6)
                        delegacion      = res(7)                        estado
        = res(8)                        pais            = res(9)                        cp
                        = res(10)                       telefono        = res(11)
                        fax             = res(12)
mail                    = res(13)                       serv_producto   = res(14)
                        banco           = res(15)                       no_cuenta
                = res(16)                               status          = res(17)
                else                    identificador   = "No definido"
         nombre                 = "No definido"                 rfc                     =
"No definido"                           calle           = "No definido"
         numero                 = "No definido"                 interior        = "No
definido"                       colonia         = "No definido"                 delegacion
                = "No definido"                         estado          = "No definido"
                        pais            = "No definido"                 cp
                        = "No definido"                 telefono        = "No definido"
                        fax             = "No definido"                 mail
                = "No definido"                         serv_producto   = "No definido"
                 banco                  = "No definido"                 no_cuenta
        = "No definido"                         status          = "No definido"
                        end if                                  response.Write
"<table border=0 width=""90.5%"" align=center>"         response.Write " <tr>"
        response.Write "        <td class=tabla_nivel_2 colspan=2>Detalle de proveedor</td>"
                response.Write " </tr>"                 response.Write "</table>"
                        response.Write "<table align=center width=""90%"" border=0>"
                response.Write " <tr>"                  response.Write "        <td
class=tabla_nivel_4>Identificador</td>"                 response.Write "        <td
```

```
class=texto_nivel_2>"&identificador&"</td>"                    response.Write " </tr>"
                                    response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>Nombre</td>"                    response.Write "          <td
class=texto_nivel_2>"&nombre&"</td>"                    response.Write " </tr>"
                                    response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>R.F.C</td>"                    response.Write "          <td
class=texto_nivel_2>"&rfc&"</td>"                    response.Write " </tr>"
                                                response.Write " <tr>"
        response.Write "          <td class=tabla_nivel_4>Calle</td>"
        response.Write "          <td class=texto_nivel_2>"&calle&"</td>"
        response.Write " </tr>"                    response.Write " <tr>"
        response.Write "          <td class=tabla_nivel_4>Numero</td>"
        response.Write "          <td class=texto_nivel_2>"&numero&"</td>"
        response.Write " </tr>"
        response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>Interior</td>"                    response.Write "          <td
class=texto_nivel_2>"&interior&"</td>"                    response.Write " </tr>"
        response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>Colonia</td>"                    response.Write "          <td
class=texto_nivel_2>"&colonia&"</td>"                    response.Write " </tr>"
                                                response.Write " <tr>"
        response.Write "          <td class=tabla_nivel_4>Delegacion o Municipio</td>"
        response.Write "          <td class=texto_nivel_2>"&delegacion&"</td>"
        response.Write " </tr>"
                response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>Estado</td>"                    response.Write "          <td
class=texto_nivel_2>"&estado&"</td>"                    response.Write " </tr>"
                                    response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>Pais</td>"                    response.Write "          <td
class=texto_nivel_2>"&pais&"</td>"                    response.Write " </tr>"
                                    response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>C.P.</td>"                    response.Write "          <td
class=texto_nivel_2>"&cp&"</td>"                    response.Write " </tr>"
                                                response.Write " <tr>"
        response.Write "          <td class=tabla_nivel_4>Telefono</td>"
        response.Write "          <td class=texto_nivel_2>"&telefono&"</td>"
        response.Write " </tr>"                                                    response.Write "
        <tr>"                    response.Write "          <td class=tabla_nivel_4>Fax</td>"
        response.Write "          <td class=texto_nivel_2>"&fax&"</td>"
        response.Write " </tr>"                                                    response.Write "
        <tr>"                    response.Write "          <td class=tabla_nivel_4>Correo
electronico</td>"                    response.Write "          <td class=texto_nivel_2>"&mail&"</td>"
                response.Write " </tr>"
                                                response.Write " <tr>"
        response.Write "          <td class=tabla_nivel_4>Servicio o producto</td>"
        response.Write "          <td class=texto_nivel_2>"&serv_producto&"</td>"
        response.Write " </tr>"                                                    response.Write "
        <tr>"                    response.Write "          <td class=tabla_nivel_4>Banco</td>"
                response.Write "          <td class=texto_nivel_2>"&banco&"</td>"
        response.Write " </tr>"                                    response.Write " <tr>"
        response.Write "          <td class=tabla_nivel_4>Numero de Cuenta</td>"
        response.Write "          <td class=texto_nivel_2>"&no_cuenta&"</td>"
        response.Write " </tr>"                                    response.Write " <tr>"
        response.Write "          <td class=tabla_nivel_4>Estado</td>"
        response.Write "          <td class=texto_nivel_2>"&status&"</td>"
        response.Write " </tr>"
        response.Write " <tr>"                    response.Write "          <td colspan=2><hr></td>"
                response.Write " </tr>"
        'response.Write " <tr>"                    'response.Write "          <td class=tabla_nivel_2
colspan=2>Asociar Conceptos</td>"                    'response.Write " </tr>"
        'response.Write " <tr>"                    '                                                    call
catalogo_concepto_gasto(idempresa,"Concepto de
```

```
gasto","idconcepto","","onchange=javascript:guarda_concepto(this.value)","","texto_nivel_2","a","")
                                                                                    'Funcion que muestra
el catalogo de conceptos de gasto
        'Localizacion = "../includes/catal_conce_gasto.asp"                 'response.Write " </tr>"
                                                    'if trim(mensaje) <> "" then                    '
        response.Write "<tr>"                       '       response.Write "  <td class=aviso colspan=2
align=center>"&mensaje&"</td>"                      '       response.Write "</tr>"
                                    'end if
                                    'call conceptos_asociados(idempresa,idproveedor,mensaje)
                                                                                    'response.Write " <tr>"
                'response.Write "        <td colspan=2><hr></td>"                    'response.Write "
        </tr>"
                                    response.Write "  <tr>"                         response.Write "        <td
colspan=2 align=center>"                                                                                    call
muestra_boton("Cerrar","window.close();")
        response.Write "        </td>"                      response.Write "  </tr>"
                end function%><html>    <head>          <title>Datos de detalle de los
proveedores</title>                 <link rel="stylesheet" href="../stylesheet/style.css">          <script
language=javascript>                <!--                    function guarda_concepto(idconcepto){
                        if(idconcepto!=0){                                      document.form.proviene.value
= 'GC';                                 document.form.action = 'catal_prove_detal.asp'
                        document.form.submit();                                 }
                                }                                                       function
elimina(idconcepto){                                    document.form.idconcepto.value = idconcepto;
                        document.form.proviene.value = 'EC';
        document.form.action = 'catal_prove_detal.asp'
        document.form.submit();                 }
                -->             </script></head> <body>  <form name="form" method="post">
                <input type=hidden name=idempresa               value=<%=idempresa%>>
        <input type=hidden name=numempleado             value=<%=numempleado%>>
        <input type=hidden name=idproveedor             value=<%=idproveedor%>>
        <input type=hidden name=proviene        value=<%=proviene%>>                    <%
                if proviene = "GC" then                                 call
guarda(idempresa,idproveedor,request("idconcepto"),"i")                 end if                  if
proviene = "EC" then                                    call
guarda(idempresa,idproveedor,request("idconcepto"),"d")                 end if
                                        call ver_registros_existentes(idempresa,idproveedor)
        'Funcion que presenta los conceptos de gasto existentes                 'Localizacion = esta
misma pagina            %>                      </form>         </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_moned.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 29 de julio 2007      'Funcionalidad
general                 : Pagina que muestra el catalogo de ivas                '
                                Este catalogo es de SOLO LECTURA, ya que los datos
        '                                                               se extraen de otro
sistema%>               <!--    #include file="../connect/conne_open.asp"--><!--    Este archivo contiene
la conexion a la BD-->          <!--    #include file="../includes/muestr_boton.asp"--><!--    Este archivo
incluye la funcion que presenta los botones-->                          <%
        '=============================================================================
==============          'Estas variables son comunes a todas las paginas        dim
numempleado                                             'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                           'Numero de
empresa en la que se esta trabajando                    numempleado =
request("numempleado")  idempresa = request("idempresa")
        '=============================================================================
==============                                          function ver_registros_existentes()
                dim sql                 dim res                 dim arRes                       dim
t_arRes                 dim count                               sql = "exec
```

```
[dbo].sp_moneda_catalogo 'a'"                    set res = connect.execute(sql)                    if
not res.eof then                    arRes = res.getrows                    t_arRes =
ubound(arRes,2)                    else                    t_arRes = -1                    end
if                    response.Write "<table border=0 width=""90%""
align=center>"                    response.Write " <tr>"                    response.Write "
<td
class=tabla_nivel_2 colspan=2>Monedas</td>"                    response.Write " </tr>"
        response.Write "</table>"                    if
t_arRes >= 0 then                    response.Write "<table align=center width=""90%""
border=0>"                    response.Write " <tr class=tabla_nivel_3>"
        response.Write "        <td>Identificador</td>"                    response.Write "
        <td>Descripcion</td>"                    response.Write "        <td>Estado</td>"
                response.Write " </tr>"
        for count = 0 to t_arRes                    response.Write "<tr
class=texto_nivel_2>"                    response.Write " <td>"&arRes(1,count)&"</td>"
                    response.Write " <td>"&arRes(2,count)&"</td>"
                response.Write " <td>"&arRes(3,count)&"</td>"
        response.Write "</tr>"                    next
        response.Write " <tr class=texto_nivel_2>"                    response.Write "        <td
colspan=4><hr></td>"                    response.Write " </tr>"
        response.Write " <tr>"                    response.Write "        <td colspan=4
align=center>"                    call
muestra_boton("Imprimir","window.print();")
                    response.Write "
        </td>"                    response.Write " </tr>"
                    response.Write "</table>"
                else                    response.Write "<table border=0 width=""90%""
align=center>"                    response.Write " <tr>"
        response.Write "        <td class=aviso colspan=2>No hay monedas definidas</td>"
                response.Write " </tr>"                    response.Write "<table>"
                    end if                    end
function%><html>        <head>        <title>Catalogo de Monedas</title>        <script
src="../includes/open_windo.js"></script>        <link rel="stylesheet" href="../stylesheet/style.css">
        <script language=javascript>        <!--
                -->        </script></head> <body>  <!--#include file=../menu/menu.asp-->
        <form name="form" method="post">                <input type=hidden name=idempresa
                value=<%=idempresa%>>                <input type=hidden name=numempleado
        value=<%=numempleado%>>                <%                call
ver_registros_existentes()                'Funcion que presenta los conceptos de gasto existentes
                'Localizacion = esta misma pagina        %>                </form>
        </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_iva.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -        'Fecha de ultima modificacion        : 29 de julio 2007        'Funcionalidad
general                : Pagina que muestra el catalogo de ivas                '
                                Este catalogo es de SOLO LECTURA, ya que los datos
        '                                                se extraen de otro
sistema%>                <!--        #include file="../connect/conne_open.asp"--><!--        Este archivo contiene
la conexion a la BD-->                <!--        #include file="../includes/muestr_boton.asp"--><!--    Este archivo
incluye la funcion que presenta los botones-->                <%
        '=============================================================================
===============                'Estas variables son comunes a todas las paginas                dim
numempleado                'Numero de empleado que esta firmado
en la aplicacion                dim idempresa                'Numero de
empresa en la que se esta trabajando                numempleado =
request("numempleado")        idempresa = request("idempresa")
        '=============================================================================
===============                function ver_registros_existentes()
                dim sql                dim res                dim arRes                dim
```

```
t_arRes                 dim count                                       sql = "exec
[dbo].sp_iva_catalogo 'a'"              set res = connect.execute(sql)          if not res.eof
then                    arRes = res.getrows                             t_arRes =
ubound(arRes,2)         else                                    t_arRes = -1            end
if                                      response.Write "<table border=0 width=""90%""
align=center>"          response.Write " <tr>"                  response.Write "        <td
class=tabla_nivel_2 colspan=2>IVA</td>"                 response.Write " </tr>"
        response.Write "</table>"                                       if
t_arRes >= 0 then                       response.Write "<table align=center width=""90%""
border=0>"                      response.Write " <tr class=tabla_nivel_3>"
        response.Write "        <td>Identificador</td>"                 response.Write "
        <td>Porcentaje</td>"                    response.Write "
        <td>Descripcion</td>"                   response.Write "        <td>Estado</td>"
                        response.Write " </tr>"
        for count = 0 to t_arRes                        response.Write "<tr
class=texto_nivel_2>"                   response.Write " <td>"&arRes(0,count)&"</td>"
                        response.Write " <td>"&arRes(1,count)&"</td>"
                response.Write " <td>"&arRes(2,count)&"</td>"
        response.Write " <td>"&arRes(3,count)&"</td>"
        response.Write "</tr>"                  next
        response.Write " <tr class=texto_nivel_2>"              response.Write "        <td
colspan=4><hr></td>"                    response.Write " </tr>"
        response.Write " <tr>"                  response.Write "        <td colspan=4
align=center>"                                                          call
muestra_boton("Imprimir","window.print();")
        response.Write "        </td>"                  response.Write " </tr>"

        response.Write "</table>"               else                    response.Write
"<table border=0 width=""90%"" align=center>"                   response.Write " <tr>"
                response.Write "        <td class=aviso colspan=2>No hay IVAS definidos</td>"
                        response.Write " </tr>"                 response.Write
"<table>"                                               end if
        end function%><html>    <head>          <title>Catalogo de IVA</title>          <script
src="../includes/open_windo.js"></script>       <link rel="stylesheet" href="../stylesheet/style.css">
        <script language=javascript>            <!--
                -->             </script></head> <body>  <!--#include file=../menu/menu.asp-->
        <form name="form" method="post">                <input type=hidden name=idempresa
                value=<%=idempresa%>>                   <input type=hidden name=numempleado
        value=<%=numempleado%>>                         <%                      call
ver_registros_existentes()              'Funcion que presenta los conceptos de gasto existentes
                'Localizacion = esta misma pagina       %>                      </form>
        </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_empre.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 28 de julio 2007      'Funcionalidad
general                 : Pagina que muestra el catalogo de centros de costo            '
                                                        Este catalogo es de SOLO LECTURA, ya que
los datos               '                                                       se extraen
de otro sistema%>               <!--    #include file="../connect/conne_open.asp"--><!--    Este archivo
contiene la conexion a la BD-->         <!--    #include file="../includes/muestr_boton.asp"--><!--
        Este archivo incluye la funcion que presenta los botones-->             <%

        '=============================================================================
==============          'Estas variables son comunes a todas las paginas        dim
numempleado                                             'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                           'Numero de
empresa en la que se esta trabajando                    numempleado =
request("numempleado")          idempresa = request("idempresa")
```

```
'================================================================================
===============                                                    function ver_registros_existentes()
                dim sql                 dim res                 dim arRes                               dim
t_arRes                 dim count                                       sql = "exec
[dbo].sp_empresa_catalogo 'a'"                  set res = connect.execute(sql)                  if
not res.eof then                        arRes = res.getrows                                     t_arRes =
ubound(arRes,2)                 else                                    t_arRes = -1                    end
if                                      response.Write "<table border=0 width=""90%""
align=center>"                  response.Write " <tr>"                  response.Write "        <td
class=tabla_nivel_2 colspan=2>Catalogo de Empresas</td>"                response.Write " </tr>"
                response.Write "<table>"
        if t_arRes >= 0 then                                    response.Write "<table align=center
width=""90%"" border=0>"                        response.Write " <tr class=tabla_nivel_3>"
                response.Write "        <td>Nombre</td>"
        response.Write "        <td>Telefono</td>"                                      response.Write "
        <td>Correo elctronico</td>"                             response.Write "
        <td>Identificador</td>"                         response.Write "        <td>Estado</td>"
                        response.Write " </tr>"
        for count = 0 to t_arRes                                response.Write "<tr
class=texto_nivel_2>"                                   response.Write " <td><a
href=javascript:ver_detalle("&arRes(0,count)&");>"&arRes(1,count)&"</a></td>"
        response.Write " <td>"&arRes(2,count)&"</td>"
        response.Write " <td>"&arRes(3,count)&"</td>"
        response.Write " <td>"&arRes(4,count)&"</td>"
        response.Write " <td>"&arRes(5,count)&"</td>"
        response.Write "</tr>"                                  next
        response.Write " <tr class=texto_nivel_2>"                      response.Write "        <td
colspan=5><hr></td>"                            response.Write " </tr>"
        response.Write " <tr>"                          response.Write "        <td colspan=5
align=center>"                                                                          call
muestra_boton("Imprimir","window.print();")
        response.Write "        </td>"                          response.Write " </tr>"

        response.Write "</table>"               else                            response.Write
"<table border=0 width=""90%"" align=center>"                   response.Write " <tr>"
                response.Write "        <td class=aviso colspan=2>No hay empresas
definidas</td>"                 response.Write " </tr>"
        response.Write "<table>"                                                end if
                        end function%><html>    <head>          <title>Catalogo de Empresas</title>
                <script src="../includes/open_windo.js"></script>               <link rel="stylesheet"
href="../stylesheet/style.css">                 <script language=javascript>                    <!--
        function ver_detalle(idempresa){
        open_window('catal_empre_detal.asp?idempresa='+idempresa+"','scrollbars=yes,status=yes,resizab
le=yes,width=800,height=500');                  }                                               -->
        </script></head> <body>  <!--#include file=../menu/menu.asp-->      <form name="form"
method="post">                  <input type=hidden name=idempresa
        value=<%=idempresa%>>                   <input type=hidden name=numempleado
        value=<%=numempleado%>>                         <%                      call
ver_registros_existentes()              'Funcion que presenta los conceptos de gasto existentes
                'Localizacion = esta misma pagina       %>                      </form>
        </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_flujo_autor.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 26 de julio 2007      'Funcionalidad
general                 : Pagina mediante la cual se dan de alta, modifican y dan de baja               '
                                                                        los tipos de concepto%>         <!--
        #include file="../connect/conne_open.asp"--><!--   Este archivo contiene la conexion a la BD-->
                <!--    #include file="../includes/muestr_texto.asp"--><!--    Este archivo incluye la funcion
```

```
que presenta una caja de texto--><!--          #include file="../includes/muestr_boton.asp"--><!--    Este archivo
incluye la funcion que presenta los botones-->                                                  <%
          '==========================================================================
===============                    'Estas variables son comunes a todas las paginas               dim
numempleado                                                                'Numero de empleado que esta firmado
en la aplicacion               dim idempresa                                                              'Numero de
empresa en la que se esta trabajando                                                numempleado =
request("numempleado")               idempresa = request("idempresa")
          '==========================================================================
===============
          '==========================================================================
===============                    'Esta variable almacena la proveniencia de la peticion de las paginas
          dim proviene               'Variable que indica de donde viene la peticion de esta pagina
                                             'IA = Insertar una nueva solicitud que viene de la pagina anterior
                    proviene =          request("proviene")

          '==========================================================================
===============
          '==========================================================================
===============                    function obtiene_tipos_operacion(idempresa)
                    'Funcione que obtiene los tipos de operacion o solicitud disponibles para la empresa en
cuestion                    'Recibe:                    'idempresa = empresa en la que se esta trabajando
                    'numempleado = empleado que esta haciendo las autorizaciones
          'status = status de la solicitud a autorizar                         'nombre_status = nombre del status
para presentarlo                                                                dim sql
                    dim res                                        dim arRes                                        dim
t_arRes                              dim count                         dim tipo_operacion
                                        sql = "exec [dbo].sp_obtiene_tipos_de_operacion "&idempresa&", 'u' "
                    set res = connect.execute(sql)                    if not res.eof then
          arRes = res.getrows                                        t_arRes = ubound(arRes,2)
          else                                        t_arRes = -1                         end if
                              if t_arRes >= 0 then                                        response.Write "<table
align=center width=""90%"" border=0 align=center>"
          response.Write "<tr>"                                        response.Write " <td
class=tabla_nivel_1 align=center>"                                        response.Write
          "Tipo de flujo"                                        response.write     "          </td>"
                                        response.Write "</tr>"
                              response.Write "<tr>"                                        response.Write " <td
align=center>"                              response.Write                         "<iframe
name=tipo_flujo height=200 scrollbars=no width=600 src=catal_tipo_flujo.asp?idempresa="&idempresa&"
frameborder=0></iframe>"                                                            response.write     "
          </td>"                              response.Write "</tr>"
                                        for count = 0 to t_arRes
          response.Write "<tr>"                                        response.Write " <td
class=tabla_nivel_1 align=center>"                                        response.Write
          arRes(2,count)                                        response.write     "          </td>"
                                        response.Write "</tr>"
                              response.Write "<tr>"                                        response.Write " <td
align=center>"                              response.Write                         "<iframe
name=frameMJ"&arRes(1,count)&" height=200 scrollbars=no width=600
src=catal_orden_flujo.asp?idempresa="&idempresa&"&idtipo_operacion="&arRes(0,count)&"
frameborder=0></iframe>"                                                            response.write     "
          </td>"                              response.Write "</tr>"
                              next                                        response.Write "</table>"
                              else                              end if                         end function
                    function main_function(idempresa,proviene)                    'Funcion que presenta al
usuario la posibilidad de agregar un concepto de gasto                              'Son los campos necesarios
para su captura                                        'Recibe                    'idempresa =
empresa en la que se esta trabajando                         'proviene = variable que indica a la funcion qe
accion la mando llamar, nada mas se usa para determinar el boton                         '
          que se presenta (MD = Modificar concepto, GC = Guardar Concepto)
                                        response.Write "<table border=0 width=""90.5%"" align=center>"
```

```
                response.Write " <tr>"                          response.Write "            <td
class=tabla_nivel_2 colspan=2>Flujo de autorizacion</td>"                   response.Write " </tr>"
                response.Write "</table>"
                                response.Write "<table border=0 width=""90%"" align=center
class=contenedor>"                          response.Write " <tr>"                          response.Write "
        <td>"                                                                                   call
obtiene_tipos_operacion(idempresa)
                'Funcion que muestra el campo de texto para introducir el nombre del concepto
                                                                        'Localizacion =
"../includes/muestr_texto.asp"                          response.Write "            </td>"
        response.Write " </tr>"                         response.Write "</table>"
                        end function                                    %><html>        <head>
        <title>Catalogo de Flujo de Autorizacion</title>                <link rel="stylesheet"
href="../stylesheet/style.css">                 <script language=javascript>                <!--
                        -->                     </script></head> <body>  <!--#include file=../menu/menu.asp-->
        <form name="form" method="post">                                <input type=hidden name=idempresa
                                value=<%=idempresa%>>                   <input type=hidden
name=numempleado                                value=<%=numempleado%>>
        <input type=hidden name=proviene                                value=<%=request("proviene")%>>
        <%                      call main_function(idempresa,proviene)                  'Funcion
principal a traves de la cual se pide o se muestra al usuario los datos del concepto de gasto
        'Localizacion = esta misma pagina       %>                      </form>
        </body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_centr_costo.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 28 de julio 2007      'Funcionalidad
general                 : Pagina que muestra el catalogo de centros de costo                    '
                                                        Este catalogo es de SOLO LECTURA, ya que
los datos               '                                                               se extraen
de otro sistema%><!--   #include file="../connect/conne_open.asp"--><!--   Este archivo contiene la
conexion a la BD--><!--   #include file="../includes/muestr_boton.asp"--><!--   Este archivo incluye la funcion
que presenta los botones--><%
        '=============================================================================
===============         'Estas variables son comunes a todas las paginas        dim
numempleado                                             'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                           'Numero de
empresa en la que se esta trabajando            numempleado = request("numempleado")
        idempresa = request("idempresa")
        '=============================================================================
===============         function ver_registros_existentes()             dim sql
        dim res                 dim arRes                       dim t_arRes                     dim
count                   sql = "exec [dbo].sp_centro_costo_catalogo "&idempresa&", 'a'"
        set res = connect.execute(sql)                  if not res.eof then
        arRes = res.getrows                             t_arRes = ubound(arRes,2)
        else                                    t_arRes = -1                    end if
        response.Write "<table border=0 width=""90%"" align=center>"
        response.Write " <tr>"                          response.Write "            <td class=tabla_nivel_2
colspan=2>Centros de costo</td>"                        response.Write " </tr>"
        response.Write "<table>"                        if t_arRes >= 0 then
        response.Write "<table align=center width=""90%"" border=0>"
        response.Write " <tr class=tabla_nivel_3>"                      response.Write "
        <td>Clave</td>"                                 response.Write "            <td>Nombre</td>"
                        response.Write "            <td>Estado</td>"
        response.Write " </tr>"                                 for count = 0 to t_arRes
                response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"&arRes(1,count)&"</td>"
        response.Write " <td>"&arRes(2,count)&"</td>"
```

```
        response.Write "  <td>"&arRes(3,count)&"</td>"
        response.Write "</tr>"                                  next
        response.Write "  <tr class=texto_nivel_2>"                     response.Write "        <td
colspan=4><hr></td>"                            response.Write "  </tr>"
        response.Write "  <tr>"                                 response.Write "        <td colspan=4
align=center>"                                                                          call
muestra_boton("Imprimir","window.print();")                     response.Write "        </td>"
                        response.Write "  </tr>"                        response.Write "</table>"
                else                            response.Write "<table border=0 width=""90%""
align=center>"                          response.Write "  <tr>"
        response.Write "        <td class=aviso colspan=2>No hay centros de costo definidos</td>"
                        response.Write "  </tr>"                        response.Write "<table>"
                end if          end function%><html>    <head>          <title>Catalogo de Centros de
Costo</title>           <script src="../includes/open_windo.js"></script>               <link rel="stylesheet"
href="../stylesheet/style.css">         <script language=javascript>            <!--            -->
                </script></head> <body>  <!--#include file=../menu/menu.asp-->      <form name="form"
method="post">          <input type=hidden name=idempresa               value=<%=idempresa%>>
        <input type=hidden name=numempleado             value=<%=numempleado%>>         <%
                        call ver_registros_existentes()                 'Funcion que presenta los
conceptos de gasto existentes                   'Localizacion = esta misma pagina       %>
        </form>  </body></html>
-0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0-

../catalogo/catal_monto_limit.asp
-0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0--0-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 26 de julio 2007      'Funcionalidad
general                 : Pagina mediante la cual se dan de alta, modifican y dan de baja               '
                                                                        conceptos de gasto%>            <!--
        #include file="../connect/conne_open.asp"--><!--   Este archivo contiene la conexion a la BD--><!--
        #include file="../includes/muestr_texto.asp"--><!--   Este archivo incluye la funcion que presenta una
caja de texto-->        <!--    #include file="../includes/catal_opera_usuar.asp"--><!--        Este archivo
incluye la funcion que muestra al usuario el tipo de operacion a seleccionar--><!--        #include
file="../includes/catal_moned.asp"--><!--   Este archivo incluye la funcion que presenta el catalogo de
monedas--><!--   #include file="../includes/catal_nivel.asp"--><!--      Este archivo incluye la funcion que
presenta el catalogo de niveles--><!--      #include file="../includes/catal_conce_gasto.asp"--><!--
        Este archivo incluye la funcion que presenta el catalogo de conceptos de gasto--><!--   #include
file="../includes/catal_terri.asp"--><!--       Este archivo incluye la funcion que presenta el catalogo de
territorios--><!--   #include file="../includes/muestr_boton.asp"--><!--   Este archivo incluye la funcion que
presenta los botones-->                         <%
        '=============================================================================
==============          'Estas variables son comunes a todas las paginas        dim
numempleado                                             'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                           'Numero de
empresa en la que se esta trabajando                                    numempleado =
request("numempleado")          idempresa = request("idempresa")
        '=============================================================================
==============
        '=============================================================================
==============          'Esta variable almacena la proveniencia de la peticion de las paginas
        dim proviene            'Variable que indica de donde viene la peticion de esta pagina
                                        'IA = Insertar una nueva solicitud que viene de la pagina anterior
                proviene =      request("proviene")

        '=============================================================================
==============
        '=============================================================================
==============          'Variables exclusivas de esta pagina
        dim idmoneda            idmoneda = request("idmoneda")                  dim idnivel
                idnivel = request("idnivel")                    dim idconcepto          idconcepto =
request("idconcepto")                   dim tipo_operacion              tipo_operacion =
```

```
request("tipo_operacion")                    dim idterritorio              idterritorio =
request("idterritorio")                      dim monto                     monto = request("monto")
                        dim tipo_guardar     tipo_guardar = request("tipo_guardar")

        '=============================================================================
===============                              function
guarda(idnivel,idconcepto,tipo_operacion,idempresa,idmoneda,idterritorio,monto,opcion)          dim
sql                                          sql = "exec [dbo].sp_monto_limite_inserta_modifica
0,"&idnivel&","&idconcepto&",'"&trim(tipo_operacion)&"',"&idempresa&","&idmoneda&","&idterritorio&","&monto
&",'"&opcion&"'"                             connect.execute(sql)          tipo_operacion = 0
                idmoneda = 0                 idconcepto = 0                idterritorio = 0
                monto = 0                end function                      function
modifica(idempresa,idmonto_limite)                   dim sql               dim res
                        sql = "exec [dbo].sp_monto_limite_catalogo "&idempresa&",
","&idmonto_limite&""                        set res = connect.execute(sql)          if not res.eof
then                                         idmonto_limite = res(0)       idnivel = res(1)
                        idconcepto = res(2)          tipo_operacion = trim(res(3))
                        idmoneda = res(4)            idterritorio = res(5)
                        monto = res(6)                        else
                        idmonto_limite = 0           idnivel = 0
                idconcepto = 0               idtipo_operacion = 0
        idmoneda = 0                         idterritorio = 0                        monto = 0
                        end if                               end function
        function elimina(idempresa,idmonto_limite)           dim sql               sql = "exec
[dbo].sp_monto_limite_inserta_modifica "&idmonto_limite&",0,0,'0',"&idempresa&",0,0,0,'d'"
        connect.execute(sql)             end function                  function
main_function(idempresa,proviene)            'Funcion que presenta al usuario la posibilidad de
agregar un concepto de gasto                 'Son los campos necesarios para su captura
                        'Recibe              'idempresa = empresa en la que se esta
trabajando                                   'proviene = variable que indica a la funcion qe accion la mando llamar,
nada mas se usa para determinar el boton     '                             que se presenta (MD
= Modificar concepto, GC = Guardar Concepto)
        response.Write "<table border=0 width=""90.5%"" align=center>"
        response.Write "  <tr>"              response.Write "    <td class=tabla_nivel_2
colspan=2>Montos Limite</td>"                response.Write "  </tr>"
        response.Write "<table>"
        response.Write "<table border=0 width=""90%"" align=center class=contenedor>"
        response.Write "  <tr>"              response.Write "    <td>"
                                                     call
catalogo_operaciones_usuario(idempresa,0,"Tipo de
operacion","tipo_operacion",tipo_operacion,"texto_nivel_2")          response.Write "
        </td>"                   response.Write "  </tr>"                        if proviene =
"ST" or proviene = "MM" then                 response.Write "  <tr>"
        response.Write "    <td>"
                call catalogo_concepto_gasto(idempresa,"Concepto de
gasto","idconcepto",idconcepto,"",tipo_operacion,"texto_nivel_2","u",0)
                                             'Funcion que muestra el catalogo de areas de
responsabilidad                                                      'Localizacion
"../includes/catal_area_respo.asp"
                        response.Write "     </td>"                  response.Write "
        </tr>"                           end if
        response.Write "  <tr>"              response.Write "    <td>"
                                                     call
catalogo_territorio("Territorio","idterritorio",idterritorio,"texto_nivel_2")
                                             'Funcion que muestra el catalogo de tipos de concepto
                                                     'Localizacion
"../includes/catal_tipo_conce.asp"
                        response.Write "     </td>"                  response.Write "
        </tr>"                                       response.Write "  <tr>"
        response.Write "    <td>"
                call catalogo_nivel(idempresa,"Nivel","idnivel",idnivel,"texto_nivel_2")
                                             'Funcion que muestra el catalogo de tipos de
```

```
concepto                                                                    'Localizacion
"../includes/catal_tipo_conce.asp"
                                response.Write "          </td>"                    response.Write "
        </tr>"                                            response.Write " <tr>"
        response.Write "          <td>"
                call muestra_texto("Monto limite $","monto",monto,"texto_nivel_2",17,15)
                                                                    'Funcion que muestra el catalogo de
cuentas contable                                                            'Localizacion
"../includes/catal_cuent_conta.asp"
                                response.Write "          </td>"                    response.Write "
        </tr>"                  response.Write " <tr>"                  response.Write "
        <td>"                                                                       call
catalogo_moneda("Moneda","idmoneda",idmoneda,"texto_nivel_2","")
                                        'Funcion que muestra el campo de texto para introducir el nombre
del concepto                                                                'Localizacion
= "../includes/muestr_texto.asp"            response.Write "          </td>"
        response.Write " </tr>"                                     response.Write "</table>"
                                        response.Write "<table border=0 width=""90.5%"" align=center>"
                        response.Write " <tr>"                  response.Write "          <td
colspan=3 align=center>"                                                                call
muestra_boton("Guardar Para esta moneda","guardar('u');")               response.Write "
         "                                                                 call
muestra_boton("Guardar para todas las monedas","guardar('m');")                 response.Write "
                 "
                        call muestra_boton("Limpiar","limpiar();")              response.Write "
        </td>"                  response.Write " </tr>"                 response.Write "<table>"
                                end function                            function
obtiene_tipos_operacion(idempresa)                  'Funcione que obtiene los tipos de operacion o
solicitud disponibles para la empresa en cuestion           'Recibe:            'idempresa =
empresa en la que se esta trabajando                'numempleado = empleado que esta haciendo
las autorizaciones                      'status = status de la solicitud a autorizar
        'nombre_status = nombre del status para presentarlo
                dim sql                                 dim res                                 dim
arRes                                   dim t_arRes                             dim count
        dim tipo_operacion                                              sql = "exec
[dbo].sp_obtiene_tipos_de_operacion ''&idempresa&'', 'u' "              set res = connect.execute(sql)
                        if not res.eof then                     arRes = res.getrows
                t_arRes = ubound(arRes,2)                       else
        t_arRes = -1                    end if                                  if t_arRes >=
0 then                                  response.Write "<table align=center width=""90%"" border=0>"
                                        for count = 0 to t_arRes
        response.Write "<tr>"                                   response.Write " <td>"
                                                                        call
ver_montos_existentes(idempresa,trim(arRes(1,count)),trim(arRes(2,count)))
                                                        'Funcion que obtiene los folios pendientes por
autorizar para este tipo de operacion
                        'Localizacion = esta misma pagina
                response.write  "         </td>"                                response.Write
"</tr>"                                                 next
        response.Write "</table>"                                       else                    end
if                                      end function                            function
ver_montos_existentes(idempresa,tipo_operacion,nombre_operacion)                dim sql
        dim res                 dim arRes                       dim t_arRes             dim
count                                           sql = "exec [dbo].sp_monto_limite_catalogo
''&idempresa&"',"'&trim(tipo_operacion)&"',0"                   set res = connect.execute(sql)
        if not res.eof then                             arRes = res.getrows
        t_arRes = ubound(arRes,2)                       else                    t_arRes = -1
                        end if                                  if t_arRes >= 0 then
                                                        response.Write " <tr class=tabla_nivel_1>"
                        response.Write "        <td colspan=7>"&nombre_operacion&"</td>"
                        response.Write " </tr>"
        response.Write " <tr class=tabla_nivel_3>"                      response.Write "
```

```
        <td>Nivel</td>"                                response.Write "             <td>Concepto de gasto</td>"
                                response.Write "             <td>Territorio</td>"
        response.Write "             <td>Monto Limite</td>"                                response.Write "
        <td>Moneda</td>"                                response.Write "             <td>Modificar</td>"
                                response.Write "             <td>Eliminar</td>"
        response.Write " </tr>"                                for count = 0 to t_arRes
                        response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"&arRes(1,count)&"</td>"
        response.Write " <td>"&arRes(2,count)&"</td>"
        response.Write " <td>"&arRes(3,count)&"</td>"
        response.Write " <td>"&formatcurrency(arRes(4,count),2)&"</td>"
        response.Write " <td>"&arRes(5,count)&"</td>"
        response.Write " <td><a href=javascript:modificar("&arRes(0,count)&");>Modificar</a></td>"
                        response.Write " <td><a
href=javascript:eliminar("&arRes(0,count)&");>Eliminar</a></td>"
        response.Write "</tr>"                                next                        else
        end if                                end function                %><html>        <head>
        <title>Catalogo de Montos Limite</title>                <link rel="stylesheet"
href="../stylesheet/style.css">                <script language=javascript>                <!--
        function selecciona_tipo_operacion(tipo_operacion){                //Funcion que se
ejecuta cuando se selecciona un concepto de gasto para modificar algun dato
                        //Recibe                        //idconcepto = identificador del
concepto de gasto a modificar
        document.form.tipo_operacion.value = tipo_operacion;
        document.form.proviene.value = 'ST';        //Se asigna el valor proviene a MD (Modificar Concepto)
                        document.form.action = 'catal_monto_limit.asp'                }
                                function limpiar(){
        document.form.proviene.value = '';                document.form.tipo_operacion.value =
'';                        document.form.idmoneda.value = '';
        document.form.idterritorio.value = '';                document.form.monto.value =
0;                        document.form.action = 'catal_monto_limit.asp'                }
                                function selecciona_moneda(){                }
                                function guardar(opcion){
        if(document.form.tipo_operacion.value==0){                alert('Favor de
seleccionar un tipo de operacion');                document.form.tipo_operacion.focus();
                        return false;                }
                                if(document.form.idmoneda.value==0){
                alert('Favor de seleccionar un tipo de moneda');
        document.form.idmoneda.focus();                return false;
                }
        if(document.form.idconcepto){
        if(document.form.idconcepto.value==0){                alert('Favor
de seleccionar un concepto de gasto');                return false;
                        }                }
                        else{                alert('Favor de seleccionar un
tipo de operacion que tenga conceptos de gastos disponibles');
        return false;                }
                if(document.form.idterritorio.value==0){
        alert('Favor de seleccionar un territorio');
        document.form.idterritorio.focus();                return false;
                }
        if(isNaN(document.form.monto.value)){                alert('Valor no valido
para el monto');                document.form.monto.value = 0;
                document.form.monto.focus();
                        return false;                }

        document.form.proviene.value = 'GM';
        document.form.tipo_guardar.value = opcion;                document.form.action =
'catal_monto_limit.asp'                document.form.submit();                }
                        function modificar(idmonto_limite){
        document.form.idmonto_limite.value = idmonto_limite;
        document.form.proviene.value = 'MM';                document.form.action =
```

```
'catal_monto_limit.asp'                                    document.form.submit();
                                    }                                                    function
eliminar(idmonto_limite){                                  document.form.idmonto_limite.value = idmonto_limite;
                                        document.form.proviene.value = 'EM';
        document.form.action = 'catal_monto_limit.asp'
        document.form.submit();                                                          }
                                                   -->             </script></head> <body>  <!--#include
file=../menu/menu.asp-->  <form name="form" method="post" >                          <input type=hidden
name=idempresa          value=<%=idempresa%>>              <input type=hidden
name=numempleado                value=<%=numempleado%>>            <input type=hidden
name=proviene           value=<%=request("proviene")%>>    <input type=hidden name=tipo_guardar
        value=<%=tipo_guardar%>>           <input type=hidden name=idmonto_limite    value="">
                <%                              select case ucase(proviene)
        case "GM"       : call
guarda(idnivel,idconcepto,tipo_operacion,idempresa,idmoneda,idterritorio,monto,tipo_guardar)
                                                   case "MM"       : call
modifica(idempresa,request("idmonto_limite"))
        case "EM"       : call elimina(idempresa,request("idmonto_limite"))                  end select
                        'Funcion principal a traves de la cual se pide o se muestra al usuario los datos del
concepto de gasto                       'Localizacion = esta misma pagina
                call obtiene_tipos_operacion(idempresa)                 'Funcion que presenta los
conceptos de gasto existentes                   'Localizacion = esta misma pagina           %>
                </form>         </body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../catalogo/catal_empre_detal.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion       : 29 de julio 2007          'Funcionalidad
general                 : Pagina que muestra el detalle de las empresas             '
                                                Este catalogo es de SOLO LECTURA, ya que los datos
                '                                                                       se extraen de otro
sistema%><!--   #include file="../connect/conne_open.asp"--><!--    Este archivo contiene la conexion a la
BD--><!--       #include file="../includes/muestr_boton.asp"--><!--  Este archivo incluye la funcion que
presenta los botones--><%
        '=============================================================================
===============         'Estas variables son comunes a todas las paginas        dim
numempleado                                             'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                           'Numero de
empresa en la que se esta trabajando            numempleado = request("numempleado")
        idempresa = request("idempresa")
        '=============================================================================
===============
        '=============================================================================
===============         'Estas variables se usa para saber de donde proviene la peticion de esta
pagina, se usa en       'la mayoria de las paginas      dim proviene            proviene =
request("proviene")
        '=============================================================================
===============
        '=============================================================================
===============         'Esta variable se utiliza para imprimir en pantalla el mensaje de error o de
cumplimiento            'de funciones de la insercion o del borrado de los anexos del detalle
        'Este mensaje viene de la pagina "../upload/upload_function.asp"         dim mensaje
        mensaje = request("mensaje")
        '=============================================================================
===============         dim cargado             cargado = request("cargado")
        function ver_registros_existentes(idempresa)                    dim sql                 dim
res                     dim arRes               dim t_arRes                     dim count
                        dim identificador       dim nombre                      dim rfc
                dim direccion                   dim colonia                     dim
```

```
delegacionmunicip                    dim estado                    dim pais                    dim
cp                    dim telefono                    dim fax                    dim email
                dim logo                    dim status                    sql = "exec
[dbo].sp_empresa_detalle_catalogo "&idempresa&""                    set res = connect.execute(sql)
                if not res.eof then                    identificador                    = res(0)
                 nombre                    = res(1)                    rfc
                = res(2)                    direccion                    = res(3)
                 colonia                    = res(4)                    delegacionmunicip          =
res(5)                    estado                    = res(6)
pais                    = res(7)                    cp
          = res(8)                    telefono                    = res(9)                    fax
                    = res(10)                    email
          = res(11)                    logo                    = res(12)
                    status                    = res(13)                    else
                    identificador                = "No definido"                    nombre
                    = "No definido"                    rfc                    =
"No definido"                    direccion                    = "No definido"
                 colonia                    = "No definido"                    delegacionmunicip
          = "No definido"                    estado                    = "No definido"
                    pais                    = "No definido"                    cp
                    = "No definido"                    telefono
          = "No definido"                    fax                    = "No definido"
                    email                    = "No definido"
logo                    = "No definido"                    status
          = "No definido"                    end if                    response.Write "<table border=0
width=""90.5%"" align=center>"                    response.Write " <tr>"
          response.Write "          <td class=tabla_nivel_2 colspan=2>Detalle de empresa</td>"
          response.Write " </tr>"                    response.Write "</table>"
          response.Write "<table align=center width=""90%"" border=0>"
          response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>Identificador</td>"                    response.Write "          <td
class=texto_nivel_2>"&identificador&"</td>"                    if trim(logo) <> "" then
          response.Write " <td class=texto_nivel_2 rowspan=9><img
src="""&lcase(application("dir_logica_emp")&logo)&"""></td>"                    end if
          response.Write " </tr>"                    response.Write " <tr>"
          response.Write "          <td class=tabla_nivel_4>Nombre</td>"
          response.Write "          <td class=texto_nivel_2>"&nombre&"</td>"
          response.Write " </tr>"                    response.Write " <tr>"
          response.Write "          <td class=tabla_nivel_4>R.F.C</td>"
          response.Write "          <td class=texto_nivel_2>"&rfc&"</td>"
          response.Write " </tr>"                    response.Write " <tr>"
          response.Write "          <td class=tabla_nivel_4>Direccion</td>"
          response.Write "          <td class=texto_nivel_2>"&direccion&"</td>"
          response.Write " </tr>"                    response.Write " <tr>"
          response.Write "          <td class=tabla_nivel_4>Colonia</td>"
          response.Write "          <td class=texto_nivel_2>"&colonia&"</td>"
          response.Write " </tr>"                    response.Write " <tr>"
          response.Write "          <td class=tabla_nivel_4>Delegacion o Municipio</td>"
          response.Write "          <td class=texto_nivel_2>"&delegacionmunicip&"</td>"
          response.Write " </tr>"                    response.Write " <tr>"
          response.Write "          <td class=tabla_nivel_4>Estado</td>"
          response.Write "          <td class=texto_nivel_2>"&estado&"</td>"
          response.Write " </tr>"                    response.Write " <tr>"
          response.Write "          <td class=tabla_nivel_4>Pais</td>"                    response.Write "
          <td class=texto_nivel_2>"&pais&"</td>"                    response.Write " </tr>"
          response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>C.P.</td>"                    response.Write "          <td
class=texto_nivel_2>"&cp&"</td>"                    response.Write " </tr>"
          response.Write " <tr>"                    response.Write "          <td
class=tabla_nivel_4>Telefono</td>"                    response.Write "          <td
class=texto_nivel_2>"&telefono&"</td>"                    if trim(logo) <> "" then
          response.Write " <td class=texto_nivel_2 rowspan=1>"&logo&"</td>"                    end if
```

```
                response.Write " </tr>"                 response.Write " <tr>"
        response.Write "        <td class=tabla_nivel_4>Fax</td>"               response.Write "
        <td class=texto_nivel_2>"&fax&"</td>"           response.Write " </tr>"
        response.Write " <tr>"                  response.Write "        <td
class=tabla_nivel_4>Correo electronico</td>"            response.Write "        <td
class=texto_nivel_2>"&email&"</td>"             response.Write " </tr>"
        response.Write " <tr>"                  response.Write "        <td
class=tabla_nivel_4>Logotipo</td>"              response.Write "        <td colspan=2>"
                response.Write "                <img src=../images/clip.gif alt=""Logotipo"">"
                response.Write "                <input type=file name=file class=caja_texto>"
                response.Write "                <input type=button style=border-color:gray;border-
width:1px;height:17px;font-size:10px; value=GUARDAR onclick=javascript:guardar_archivo(1,");>"
                response.Write "        </td>"          response.Write " </tr>"
        response.Write " <tr>"                  response.Write "        <td
class=tabla_nivel_4>Estado</td>"                response.Write "        <td
class=texto_nivel_2>"&status&"</td>"            response.Write " </tr>"
        response.Write " <tr>"                  response.Write "        <td colspan=3><hr></td>"
                response.Write " </tr>"                 response.Write " <tr>"
        response.Write "        <td colspan=6 class=aviso align=center>"
        response.Write                  request("mensaje")
        response.Write "        </td>"                  response.Write " </tr>"
        response.Write " <tr>"                  response.Write "        <td colspan=3 align=center>"
                                                                        call
muestra_boton("Cerrar","window.close();")               response.Write "        </td>"
        response.Write " </tr>"         end function%><html>    <head>          <title>Datos de
detalle de la empresas</title>          <script src="../includes/open_windo.js"></script>
        <link rel="stylesheet" href="../stylesheet/style.css">          <script language=javascript>
        <!--                    function guardar_archivo(valor,nombre_archivo){
        //Esta funcion ejecuta la forma para que se anexe un archivo al detalle de la solicitud
                //valor = opcion de guardar o eliminar un archivo (1,0; respectivamente)
                //nombre_archivo = nombre del archivo, en caso de eliminarlo
        //idanexo = identificador del anexo                     document.form.nombre_archivo.value
= nombre_archivo;                       document.form.save.value = valor;
        document.form.submit();                 }               -->             </script></head>
        <body>  <form name="form" method="post" action="../upload/upload_function_imagen_empresa.asp"
enctype="multipart/form-data">          <input type=hidden name=idempresa
        value=<%=idempresa%>>          <input type=hidden name=numempleado
        value=<%=numempleado%>>                 <input type=hidden name=proviene
        value=<%=proviene%>>           <input type=hidden name=saveto                   value=disk>
                <input type=hidden name=save                    value=0>                <input
type=hidden name=nombre_archivo         value="">               <input type=hidden name=idanexo
                value="">               <%                      call
ver_registros_existentes(idempresa)                     'Funcion que presenta los conceptos de gasto
existentes                      'Localizacion = esta misma pagina               if cargado = 1 then
                                response.Write "<script language=javascript>"
        response.Write " function recarga(){"                                   response.Write "
        opener.document.location.reload();"                                     response.Write " }"
                        response.Write " recarga();"                            response.Write
"</script>"                     end if          %>      </form>  </body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../historial/repor_histo.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 5 de julio 2007       'Funcionalidad
general                 : Pagina que presenta el historial de la solicitud%><!--        #include
file="../connect/conne_open.asp"--><!--     Este archivo contiene la conexion a la BD--><!--       #include
file="../includes/catal_centr_costo.asp"--><!--         Este archivo incluye la funcion que muestra el catalogo
de centros de costo--><!-- #include file="../includes/muestr_boton.asp"--><!--    Este archivo incluye la funcion
que los botones--><!--      #include file="../includes/forma_fecha.asp"--><!--     Este archivo incluye la funcion
```

```
que da formato a las fechas--><%
	'=============================================================================
==============	'Estas variables son comunes a todas las paginas	dim
numempleado	'Numero de empleado que esta firmado
en la aplicacion	dim idempresa	'Numero de
empresa en la que se esta trabajando	numempleado = request("numempleado")
	idempresa = request("idempresa")
	'=============================================================================
==============
	'=============================================================================
==============	'Estas variables se usan nada mas en las paginas de solicitudes
	dim folio	'Numero de folio de la solicitud
		folio = request("folio")
	'=============================================================================
==============	function datos_grales(folio,clase)	dim sql
	dim res	'Esta funcion presenta los datos generales de la solicitud
	sql = "exec [dbo].sp_muestra_datos_grale "&folio&" "	set res = connect.execute(sql)
		if not res.eof then	response.Write "<tr
class=tabla_nivel_2>"	response.Write " <td colspan=5>Historial</td>"
		response.Write "</tr>"	response.Write "<tr
class="&clase&">"	response.Write " <td>Tipo de solicitud</td>"
		response.Write " <td colspan=4>"&res(7)&"</td>"
	response.Write "</tr>"	response.Write "<tr class="&clase&">"
		response.Write " <td>Folio</td>"	response.Write " <td
colspan=4>"&res(0)&"</td>"	response.Write "</tr>"
	response.Write "<tr class="&clase&">"	response.Write "
	<td>Solicitante</td>"	response.Write " <td
colspan=4>"&res(1)&"</td>"	response.Write "</tr>"
	response.Write "<tr class="&clase&">"	response.Write "
	<td>Status</td>"	response.Write " <td colspan=4>"&res(5)&"</td>"
		response.Write "</tr>"	end if	end function
	function main_function(folio)	dim sql	dim res
	dim arRes	dim t_arRes	dim count
	dim backcolor	'Esta funcion presenta el detalle historico de la solicitud
		sql = "exec [dbo].sp_obtiene_historial "&folio&" "	set res =
connect.execute(sql)	if not res.eof then	arRes = res.getrows
		t_arRes = ubound(arRes,2)	else
		t_arRes = -1	end if	response.Write "<tr>"
		response.Write " <td colspan=5 align=center><hr></td>"
	response.Write "</tr>"	if t_arRes >= 0 then
	response.Write "<tr class=tabla_nivel_3>"	response.Write "
	<td>Persona</td>"	response.Write " <td>Status</td>"
		response.Write " <td>Fecha</td>"
	response.Write " <td>Hora</td>"	response.Write "
	<td>Observacion</td>"	response.Write "</tr>"
		for count = 0 to t_arRes	if
trim(lcase(mid(arRes(2,count),1,1))) = "r" then	backcolor =
"#FFFFCC"	else
	backcolor = ""	end if
	response.Write "<tr class=texto_nivel_2 bgcolor="&backcolor&">"
	response.Write " <td>"&arRes(1,count)&"</td>"
	response.Write " <td>"&arRes(3,count)&"</td>"
	response.Write " <td>"&formato_fecha(arRes(4,count))&"</td>"
	response.Write " <td>"&arRes(5,count)&"</td>"
	response.Write " <td>"&arRes(6,count)&"</td>"
	response.Write "</tr>"	response.Write "<tr>"
		response.Write " <td colspan=5 align=center><hr></td>"
		response.Write "</tr>"	next	else
		response.Write "<tr class=aviso>"	response.Write " <td
colspan=5 align=center>"	response.Write "	Todavia no hay historial
registrado para esta solicitud"	response.Write " </td>"
	response.Write "</tr>"	response.Write "<tr>"
```

```
response.Write " <td colspan=5 align=center><hr></td>"
response.Write "</tr>"                    end if                    response.Write " <tr>"
response.Write "          <td colspan=5 align=center>"
                                          call muestra_boton("Cerrar","window.close();")
                                                                    'Funcion que muestra los
botones de operacion
'Localizacion = "../includes/muestr_boton.asp"                     response.Write "
</td>"                    response.Write " </tr>"          end function%><html>     <head>
<title>Historial de Autorizaciones</title>          <script
src="../includes/no_right_mouse.js"></script>          <link rel="stylesheet"
href="../stylesheet/style.css">          <script language=javascript>          <!--          -->
          </script></head> <body>  <form name="form" method="post">          <input
type=hidden name=idempresa          value=<%=idempresa%>>          <input type=hidden
name=numempleado          value=<%=numempleado%>>          <table align=center
width=90%>          <%                    call datos_grales(folio,"texto_nivel_2")
call main_function(folio)                    'Funcion que da al usuario la opcion de anexar un
archivo a la solicitud                    'Localizacion = esta misma pagina          %>
</table> </form>  </body></html>


-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../connect/conne_close.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%        'Fte_ - -  'Fecha de ultima modificacion        : 5 de julio 2007  'Funcionalidad general
          : Cierra la conexion y asigna al objeto el valor de nulo          connect.close      set connect =
nothing%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../connect/conne_open.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%        'Fte_ - -  'Fecha de ultima modificacion        : 5 de julio 2007  'Funcionalidad general
          : Se crea la variable de conexion para las consultas sql          'Si se quiere ejecutar una pagina
diferente a la de login o logout (por voluntad propia)  'se envia al usuario a la pagina de logout, misma que
destruye la sesion del usuario        'y lo obliga a firmarse de nuevo      if request("logout") <> "logout" and
request("login") <> "login" then                    if session("idempresa") = "" or session("numempleado") = "" or
session("idempresa") = 0 or session("numempleado") = 0  then                                  response.Redirect
"../logout/logout.asp"                    end if     end if     dim connect 'Variable de conexion  set connect =
server.createobject("ADODB.Connection")   connect.Open application("odbc")%><!--      #include
file="../includes/funci_exist_folde.asp"--><!--Este archivo contiene la funcion que dice si cierto folder existe o
no en la estrucura de la aplicacion-->
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../errores/error_manag.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%     Option Explicit 'Esta sentencia obliga a declarar variables       On Error
Resume Next      Response.Clear  Dim objError       Set objError = Server.GetLastError()
          'Fte_ - -          'Fecha de ultima modificacion        : 5 de julio 2007          'Funcionalidad
general                    : Pagina mediante la cual se controla                    '
                                        la presentacion de los errores en la aplicacion
          '=============================================================================
===============                    'Estas variables son comunes a todas las paginas                dim
numempleado                                                    'Numero de empleado que esta firmado
en la aplicacion          dim idempresa                                                    'Numero de
empresa en la que se esta trabajando                    numempleado = request("numempleado")
          idempresa = request("idempresa")
          '=============================================================================
===============%><!--#include file="../connect/conne_open.asp"--><!--      Este archivo contiene la
conexion a la BD--><!--     #include file="../includes/muestr_boton.asp"--><!--     Este archivo incluye la funcion
```

```
que los botones--><%	function obtiene_nombre(idempresa,numempleado)	'Funcion que obtiene el nombre de un empleado	'Recibe:	'idempresa = empresa en la que se esta trabajando	'numempleado = de quien se quiere obtener el nombre	dim sql
	dim res	sql = "select [dbo].fn_obtiene_nombre("&idempresa&","&numempleado&")"
	set res = connect.execute(sql)	if not res.eof then	obtiene_nombre = trim(res(0))	else	obtiene_nombre = "No definido"	end if	end function	function obtiene_email(idempresa,numempleado)	'Funcion que obtiene el correo de un empleado	'Recibe:	'idempresa = empresa en la que se esta trabajando
	'numempleado = de quien se quiere obtener el correo	dim sql	dim res
	sql = "select [dbo].fn_obtiene_email("&idempresa&","&numempleado&")"	set res = connect.execute(sql)	if not res.eof then	obtiene_email = trim(res(0))
	else	obtiene_email = "No definido"	end if	end function
	function envia_mail_administrador(idempresa,numempleado,numero,archivo,linea,descripcion)
	'Funcion que envia un correo al administrador del sistema, avisandole del error ocurrido
	'Recibe:	'idempresa = empresa en la que se esta trabajando	'numempleado = empleado a quien le ocurrio el error	'numero = numero descriptivo del error
	'archivo = archivo en donde ocurrio el error	'linea = linea en donde ocurrio el error
	'descripcion = descripcion del error	dim sql	dim body	body =
		"Solicitante : " & obtiene_nombre(idempresa,numempleado) & "<br>"	body = body &	"Numero : " & numero & "<br>"	body = body &	"Archivo : " & archivo & "<br>"
	body = body &	"Linea : " & linea & "<br>"	body = body &	"Descripcion : " & replace(descripcion,"'","") & "<br>"	connect.execute(sql)	end function%><html>	<head>
	<title>Manejo de Errores</title>	<link rel="stylesheet" href="../stylesheet/style.css">
	<script src="../includes/no_right_mouse.js"></script>	<script language=javascript>
	<!--	function pagina_inicio(){	//Funcion que lleva al usuario a la pagina de bienvenida	document.form.action = '../inicio/inicio_inici.asp';
		document.form.submit();	}
	function tiene_opener(){	//Se revisa si la ventana que contiene el error fue abierta por alguna	//otra ventana	//Si fue abierta por otra se pinta un boton de cerrar	if(window.opener){
		boton.innerHTML = '<input class=boton type=button value=cerrar onclick=window.close();>';	}	//Si no fue abierta por otra se direcciona a la pagina de inicio	else{
	boton.innerHTML = '<input class=boton type=button value="Ir a pagina de Inicio" onclick=pagina_inicio();>';	}	}	-->
	</script></head> <body onload=javascript:tiene_opener();>	<script src="../includes/no_right_mouse.js"></script>	<form name="form" method="post">
	<input type=hidden name=idempresa	value=<%=idempresa%>>
	<input type=hidden name=numempleado	value=<%=numempleado%>>
	<%	'Funcion que envia el mail al administrador	'Localizacion = esta misma pagina	%>	<table width=60.5% border=0>	<tr>
	<td class=tabla_nivel_2 align=center>Error</td>	</tr>	</table>
	<table width=60% border=0 class=contenedor>	<tr>
	<td class=texto_nivel_2 align=center>Se ha detectado un error en la aplicaci&oacute;n</td>
		</tr>	<tr>	<td class=texto_nivel_2 align=center>Se le ha enviado un correo al admistrador del sistema</td>	</tr>
		<tr>	<td class=texto_nivel_2 align=center>Favor de intentar nuevamente en unos momento m&aacute;s</td>	</tr>	</table>
	<table width=50.5% border=0>	<tr>	<td align=center>	<div id=boton>	<!-- Aqui se pinta un boton dependiendo de si la ventana abierta es nueva o no-->
	</div>	</td>	</tr>	</table> </form>
	</body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../kilometro/catal_kilom.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
	'Fte_ - -	'Fecha de ultima modificacion	: 6 de julio 2007	'Funcionalidad general	: Pagina mediante la cual se seleccionan	'
```

```
                                        el origen y destino cuando se selecciona el concepto
        '                                                                   de kilometraje%><!--
        #include file="../connect/conne_open.asp"--><!--       Este archivo contiene la conexion a la BD--><%

        '=========================================================================
==============        'Estas variables son comunes a todas las paginas        dim
numempleado                                             'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                           'Numero de
empresa en la que se esta trabajando            numempleado = request("numempleado")
        idempresa = request("idempresa")
        '=========================================================================
==============
        '=========================================================================
==============        'Estas variables se usan nada mas en las paginas del detalle de las
solicitudes             dim idkm
        'Valor del idetificador del viaje (relacion origen-destino)             idkm = request("idkm")
        '=========================================================================
==============
        '=========================================================================
==============        'Estas variables se usan nada mas en las paginas del detalle de las
solicitudes             dim no_viajes                                           'Numero de
viajes para esa relacion origen-destino         no_viajes = request("no_viajes")
        '=========================================================================
==============
        '=========================================================================
==============        'Estas variables se usan nada mas en las paginas del detalle de las
solicitudes             dim idmoneda                                            'Identificador
de la moneda de la solicitud            idmoneda = request("idmoneda")          'sirve para
obtener el costo por kilometro
        '=========================================================================
==============        function kilometraje(idempresa,idkm,no_viajes,idmoneda)
        'Esta funcion presenta los origenes y destinos para la empresa          'en cuestion
                'idempresa = empresa de la que se quiere obtener el catalogo de origenes y
destinos                'idkm = identificador del viaje                 'no_viajes = numero de viajes
para ese identificador                  'idmoneda = identificador de la moneda de la solicitud
                dim sql                 dim res                 dim arRes                       dim
t_arRes                 dim count                       dim gran_total                  dim
seleccionar                     dim valor                       sql = "exec
[dbo].sp_distancias_catalogo "&idempresa&", "&idmoneda&" "               if not res.eof then
                arRes = res.getrows                             t_arRes = ubound(arRes,2)
                else                            t_arRes = -1                    end if
        response.Write "<table border=0 width=""90%"" align=center>"
        response.Write " <tr>"                  response.Write "        <td class=tabla_nivel_2
colspan=5>Distancias</td>"                      response.Write " </tr>"                 if t_arRes >=
0 then                          gran_total = 0                          response.Write " <tr
class=tabla_nivel_3>"                           response.Write "        <td>No.</td>"
                response.Write "        <td>Origen</td>"                        response.Write "
        <td>Destino</td>"                               response.Write "        <td>Distancia
(KM)</td>"                      response.Write "        <td>No. Viajes</td>"
                response.Write " </tr>"                 for count = 0 to t_arRes
                        if trim(idkm) = trim(arRes(0,count)) then
        seleccionar = "checked"                                         valor = no_viajes
                        else                                            seleccionar = ""
                                valor = 1                               end if
                response.Write "<tr>"                                   response.Write " <td><input
"&seleccionar&" type=radio name=idkm id=idkm"&count&"
onclick=javascript:registra_kilometro("&arRes(0,count)&","&arRes(3,count)&",document.form.no_viajes"&count
&".value,"&arRes(4,count)&","&count&");></td>"                                  response.Write " <td
class=texto_nivel_2>"&arRes(1,count)&"</td>"                                    response.Write " <td
class=texto_nivel_2>"&arRes(2,count)&"</td>"                                    response.Write " <td
class=texto_nivel_2>"&arRes(3,count)&"</td>"                                    response.Write "
        <td><input class=caja_texto_numeros size=3 type=text name=no_viajes"&count&"
```

```
onblur=javascript:registra_kilometro("&arRes(0,count)&","&arRes(3,count)&",document.form.no_viajes"&count
&".value,"&arRes(4,count)&","&count&");  value="&valor&"></td>"
        response.Write "</tr>"                                  next                    else
                response.Write "<tr class=aviso>"                               response.Write "  <td
colspan=5 align=center>No hay distancias registradas, favor de notificar al administrador</td>"
                response.Write "</tr>"                          end if                  response.Write "<tr
class=texto_nivel_2>"                   response.Write "  <td colspan=5><hr></td>"
        response.Write "</tr>"                          response.Write "<tr>"
        response.Write "  <td colspan=5 align=center><input class=boton type=button value=""Cerrar""
onclick=javascript:window.close();></td>"                       response.Write "</tr>"
        response.Write "</table>"       end function%><html>    <head>
        <title>Cat&aacute;logo de distancias</title>            <script
src="../includes/open_window.js"></script>          <script src="../includes/no_right_mouse.js"></script>
        <link rel="stylesheet" href="../stylesheet/style.css">          <script language=javascript>
        <!--                    function registra_kilometro(idkm,distancia,no_viajes,costo,idfila){
                        //Esta funcion ingresa los datos correspondientes a la pagina que la abrio
(../solicitud/solic_detal_captu.asp)                    //Recibe:
        //idkm = identificador del registro de kilometro                        //distancia = numero
de kilometros entre el origen y el destino                      //no_viajes = numero de viajes
                        //costo = costo por kilometro                           //idfila = identificador
de la fila seleccionada..(el contador)                  //                      esto sirve para
seleccionar el mismo registro en caso de que el                         //                  gasto ya
haya sido capturado y se pretenda modificarlo                           registro =
'document.form.idkm'+idfila+'.checked = true';                          eval(registro);
                if((no_viajes==0)||(no_viajes<0)||(isNaN(no_viajes))){                   //Se
evaluan los valores introducidos                        alert('Valor no valido para el numero de
viajes');                       no_viajes = 1;                                  registro =
'document.form.no_viajes'+idfila+'.value = ' + no_viajes + '';
        eval(registro);                         }                               //Si los datos
introducidos estan correctos,                   //se copian estos datos a la pagina que abrio
este catalogo                   //(../solicitud/solic_detal_captu.asp)
        kilometraje_reco = eval(distancia)*eval(no_viajes);                     kilometraje_costo =
eval(costo);                    total_viajes = eval(distancia)*eval(costo)*eval(no_viajes);
                        window.opener.document.form.kilometraje_reco.value = kilometraje_reco;
                        window.opener.document.form.kilometraje_costo.value = kilometraje_costo;
                        window.opener.document.form.idkm.value = idkm;
        window.opener.document.form.no_viajes.value = no_viajes;
        window.opener.document.form.subtotal.value = total_viajes;                      //Se ejecuta
en la pagina principal la funcion que calcula los totales
        window.opener.suma_total();                     }               -->             </script>
        </head> <body>  <form name="form" method="post">                <input type=hidden
name=idempresavalue=<%=idempresa%>>         <input type=hidden name=numempleado
        value=<%=numempleado%>>     <%                  call
kilometraje(idempresa,idkm,no_viajes,idmoneda)              'Funcion que presenta el catalogo de kilometros
(origenes y distancias)             'Localizacion = esta misma pagina %>        </form>  </body></html>

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../navigator/navig_updat.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -           'Fecha de ultima modificacion       : 5 de julio 2007           'Funcionalidad
general                     : Pagina que muestra los requisitos minimos         '
                                    para operar el sistema%><!--    #include
file="../includes/muestr_boton.asp"--><!--  Este archivo incluye la funcion que los botones--><%
        function actualizar_explorador()                        response.Write "<table border=0
width=""50.5%"" align=center>"                  response.Write "  <tr>"
        response.Write "        <td colspan=2 class=tabla_nivel_2>"
        response.Write "                Requisitos Minimos"                     response.Write "
        </td>"                  response.Write "  </tr>"                        response.Write "</table>"
                response.Write "<table border=0 width=""50%"" class=contenedor>"
```

```
        response.Write " <tr>"                          response.Write "          <td class=texto_nivel_2>"
                response.Write "                        Para un mejor funcionamiento, la aplicacion debe ser
operada bajo los siguientes requisitos:<br>"                    response.Write "                 <ol>"
                response.Write "                                <li>Explorador: Internet Explorer</li>"
                response.Write "                                <li>Version: 6.0 o mayor</li>"
        response.Write "                                <li>Javascript: Habilitado</li>"
        response.Write "                </ol>"                  response.Write "          </td>"
                response.Write " </tr>"                 response.Write "</table>"
        response.Write "<table border=0 width=""50.5%"" align=center>"
        response.Write " <tr>"                          response.Write "          <td colspan=2 align=center>"
                                                                call
muestra_boton("Obtener Internet Explorer 6.0","actualizar_explorador('IE');")
        response.Write "        </td>"                  response.Write " </tr>"
        response.Write "</table>"       end function%><html>    <head>          <title>Actualizacion
del Explorador</title>          <link rel="stylesheet" href="../stylesheet/style.css">          <script
src="../includes/no_right_mouse.js"></script>           <script language=javascript>            <!--
                function actualizar_explorador(){               document.form.action
= 'http://www.microsoft.com/downloads/details.aspx?FamilyID=1e1550cb-5e5d-48f5-b02b-
20b602228de6&displaylang=es';                   document.form.submit();                 }
                -->             </script></head> <body>  <form name="form" method="post">
        <%                      call actualizar_explorador()                    'Funcion que
presenta los requisitos minimos para operar el sistema                  'Localizacion = esta misma
pagina          %>      </form>  </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../cambio_password/cambi_passw.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -       'Fecha de ultima modificacion   : 26 de julio 2007      'Funcionalidad
general                 : Pagina mediante la cual los usuarios cambian su clave de acceso%><!--
        #include file="../connect/conne_open.asp"--><!--   Este archivo contiene la conexion a la BD--><!--
        #include file="../includes/muestr_texto.asp"--><!--   Este archivo incluye la funcion que presenta una
caja de texto--><!--    #include file="../includes/muestr_boton.asp"--><!--   Este archivo incluye la funcion
que presenta los botones--><%
        '=============================================================================
==============          'Estas variables son comunes a todas las paginas        dim
numempleado                                     'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                   'Numero de
empresa en la que se esta trabajando            numempleado = request("numempleado")
        idempresa = request("idempresa")
        '=============================================================================
==============
        '=============================================================================
==============          'Esta variable almacena la proveniencia de la peticion de las paginas
        dim proviene            'Variable que indica de donde viene la peticion de esta pagina
        proviene =      request("proviene")             'Variable en donde se guarda la respuesta que
el sistema le da al usuario     'respecto de su peticion de cambiar su clave de acceso          dim
respuesta               'La clave actual del usuario            dim clave_actual        clave_actual
= request("clave_actual")       'La clave nueva del usuario             dim clave_nueva
        clave_nueva = request("clave_nueva")
        '=============================================================================
==============          function
guardar_clave(idempresa,numempleado,clave_actual,clave_nueva)           'Funcion que
actualiza la clave del usuario                  'Recibe:                'idempresa = empresa a la
que pertenece el usuario                'numempleado = empleado que quiere cambiar su clave de
acceso                  'clave_actual = la clave de acceso actual que tiene el usuario
        'clave_nueva = la clave de acceso nueva                 dim sql                 dim res
                sql = "exec [dbo].sp_clave_acceso_inserta_modifica
"&idempresa&","&numempleado&","&clave_actual&"","&clave_nueva&""""                      set res =
connect.execute(sql)                    if not res.eof then                     respuesta = res(0)
                else                                    respuesta = ""                  end if
```

```
        end function                    function main_function(idempresa,proviene)                      'Funcion que
presenta al usuario la posibilidad de cambiar su clave de acceso                                        'Recibe
        'idempresa = empresa en la que se esta trabajando                       'proviene = variable que se
utiliza para determinar la accion a seguir                                      '                       GC = Guardar Clave
                        response.Write "<table border=0 width=""90.5%"" align=center>"
        response.Write " <tr>"                  response.Write "                <td class=tabla_nivel_2
colspan=2>Cambio de clave de Acceso</td>"               response.Write " </tr>"
        response.Write "</table>"               response.Write "<table border=0 width=""90%""
align=center class=contenedor>"                 response.Write " <tr>"
        response.Write "            <td class=texto_nivel_2 width=250>Clave de Acceso Actual</td>"
                response.Write "            <td>"                       response.Write "
        <input type=password name=clave_actual class=caja_texto size=15 maxlength=10>"
        response.Write "            </td>"                      response.Write " </tr>"
        response.Write " <tr>"                  response.Write "            <td class=texto_nivel_2>Clave
de Acceso</td>"                 response.Write "            <td>"                       response.Write "
                <input type=password name=clave_nueva class=caja_texto size=15 maxlength=10>"
                response.Write "            </td>"                      response.Write " </tr>"
        response.Write " <tr>"                  response.Write "            <td
class=texto_nivel_2>Confirmar Clave de Acceso</td>"                     response.Write "
        <td>"                   response.Write "                        <input type=password
name=clave_conf class=caja_texto size=15 maxlength=10>"                 response.Write "
        </td>"                  response.Write " </tr>"                 'Si hubo algun mensaje por
parte del sistema, se le despliega al usuario                   if trim(respuesta) <> "" then
                response.Write "<tr>"                   response.Write " <td class=aviso
colspan=2 align=center>"&respuesta&"</td>"                      response.Write "</tr>"
                end if                  response.Write "</table>"                       response.Write
"<table border=0 width=""90.5%"" align=center>"                 response.Write " <tr>"
        response.Write "            <td colspan=3 align=center>"
                        call muestra_boton("Guardar","guarda();")
                                                        'Funcion que muestra los botones a utilizarse
                                                                'Localizacion
"../includes/muestr_boton.asp"                  response.Write "            </td>"
        response.Write " </tr>"                 response.Write "<table>"            end
function%><html>        <head>          <title>Actualizaci&oacute;n de Clave de Acceso</title>
        <script src="../includes/open_windo.js"></script>               <script
src="../includes/no_right_mouse.js"></script>                   <link rel="stylesheet"
href="../stylesheet/style.css">                 <script language=javascript>                    <!--
        function guarda(){                                      //Funcion que valida los datos de entrada y
envia a actualizar                              //la clave de acceso                                    var
carac_invalidos = "',;";                                if
(document.form.clave_nueva.value!=document.form.clave_conf.value){
        alert('Las claves de acceso no coinciden\nFavor de introducir los datos de nuevo');
                        document.form.clave_nueva.value = '';
        document.form.clave_conf.value = '';
        document.form.clave_nueva.focus();                                                      return false;
                        }                                       valida = 1;
        for(mj=0;mj<=document.form.clave_nueva.value.length - 1;mj++){
        if(carac_invalidos.indexOf(document.form.clave_nueva.value.charAt(mj))!=-1){
                        alert('Caracter No Valido: ' + document.form.clave_nueva.value.charAt(mj) +
'\nFavor de modificar la clave de acceso');
        document.form.clave_nueva.value = '';
        document.form.clave_conf.value = '';
        document.form.clave_nueva.focus();                                                      valida = 0;
                                                        break;                                  }
                        }                               if(valida==0)
        return false;                                   document.form.proviene.value = 'GC';
                document.form.action = 'cambi_passw.asp';
        document.form.submit();                         }               -->             </script></head>
        <body>  <!--#include file=../menu/menu.asp-->       <form name="form" method="post">
        <input type=hidden name=idempresa                                       value=<%=idempresa%>>
        <input type=hidden name=numempleado                                     value=<%=numempleado%>>
                <input type=hidden name=proviene
```

value=<%=request("proviene")%>> <% if proviene = "GC" then call guardar_clave(idempresa,numempleado,clave_actual,clave_nueva) 'Funcion que actualiza la clave de acceso 'Localizacion = esta misma pagina end if call main_function(idempresa,proviene) 'Funcion principal a traves de la cual se da la posibilidad al usuario de cambiar su clave de acceso 'Localizacion = esta misma pagina %> </form> </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../cambio_password/cambi_olvid_passw.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><!-- #include file="../connect/conne_open.asp"--><!-- Este archivo contiene la conexion a la BD--><%'Esta pagina le envia un correo elctronico a la persona que haya olvidado su clave de'acceso'Empresa a la que el usuario dice que pertencedim idempresaidempresa = request("idempresa")'El numero de empleado que el usuario dice tenerdim numempleadonumempleado = request("numempleado")'El tipo de usuario que el usuario dice serdim idtipo_usuarioidtipo_usuario = request("idtipo_usuario")'Sentencia sqldim sqldim resdim mensaje'Con estos datos se envia un correo electronico al usuario que ha olvidado su clave de accesosql = "exec [dbo].sp_recupera_password "&idempresa&", "&numempleado&", "&idtipo_usuario&""set res = connect.execute(sql)if not res.eof then mensaje = res(0)else mensaje = "No_se_encontro_registro_alguno_con_esos_datos_no_se_envio_ningun_correo"end if'Se redirecciona al usuario a la pagina de loginresponse.Redirect "../login.asp?mensaje="&mensaje&"&login=login"%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../deposito/catal_depos.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><% 'Fte_ - - 'Fecha de ultima modificacion : 6 de julio 2007 'Funcionalidad general : Pagina mediante la cual se asocia ' un deposito a una comprobacion de anticipo%><!-- #include file="../connect/conne_open.asp"--><!-- Este archivo contiene la conexion a la BD--><!-- #include file="../includes/forma_fecha.asp"--><!-- Este archivo incluye la funcion que da formato a las fechas--><%

'== 'Estas variables son comunes a todas las paginas dim numempleado 'Numero de empleado que esta firmado en la aplicacion dim idempresa 'Numero de empresa en la que se esta trabajando numempleado = request("numempleado") idempresa = request("idempresa")
'==
'== 'Estas variables se usan nada mas en las paginas del detalle de las solicitudes dim folio_deposito 'Identificador unico del deposito folio_deposito = request("folio_deposito")
'==
'== 'Estas variables se usan nada mas en las paginas del detalle de las solicitudes dim idmoneda 'Identificador de la moneda de la solicitud idmoneda = request("idmoneda")
'== function deposito(idempresa,numempleado,folio_deposito,idmoneda) 'Esta funcion presenta los depositos hechos por el empleado en cuestion 'que se pueden asociar a una comprobacion de anticipo 'idempresa = empresa a la que pertenece la comprobacion 'numempleado = empleado que hace la comprobacion 'folio_deposito = identificador del deposito 'idmoneda = identificador de la moneda de la solicitud dim sql dim res dim arRes dim t_arRes dim count dim gran_total dim seleccionar dim valor

```
	sql = "exec [dbo].sp_obtiene_depositos_por_empleado
"idempresa&","&numempleado&","&idmoneda&", "&folio_deposito&"                    set res =
connect.execute(sql)                    if not res.eof then                    arRes = res.getrows
                    t_arRes = ubound(arRes,2)                    else
          t_arRes = -1                    end if          response.Write "<table
border=0 width=""90%"" align=center>"          response.Write " <tr class=texto_nivel_2>"
          response.Write "          <td colspan=8><hr></td>"          response.Write "
	</tr>"          if t_arRes >= 0 then          gran_total = 0
                response.Write "<tr>"          response.Write " <td
class=tabla_nivel_2 colspan=4>Depositos</td>"          response.Write " <td
class=tabla_nivel_2 colspan=2>Datos de visualizacion</td>"          response.Write " <td
class=tabla_nivel_2 colspan=2>Datos originales</td>"          response.Write
"</tr>"          response.Write "<tr class=tabla_nivel_3>"
	response.Write " <td></td>"          response.Write " <td>Referencia</td>"
          response.Write " <td>Fecha de deposito</td>"
	response.Write " <td>Banco</td>"          response.Write " <td>Monto</td>"
                response.Write " <td>Moneda</td>"
	response.Write " <td>Monto</td>"          response.Write " <td>Moneda</td>"
          response.Write "</tr>"          for count = 0 to
t_arRes          if trim(folio_deposito) = trim(arRes(0,count)) then
          seleccionar = "checked"          else
          seleccionar = ""          end if
          response.Write "<tr class=texto_nivel_2>"
	response.Write " <td><input "&seleccionar& type=radio name=folio_deposito
id=folio_deposito"&count&"
onclick=javascript:registra_deposito("&arRes(0,count)&","&arRes(4,count)&","&count&");></td>"
                response.Write " <td>"&arRes(1,count)&"</td>"
	response.Write " <td>"&formato_fecha(arRes(2,count))&"</td>"
	response.Write " <td>"&arRes(3,count)&"</td>"
	response.Write " <td>$"&arRes(4,count)&"</td>"
	response.Write " <td>"&arRes(5,count)&"</td>"
	response.Write " <td>$"&arRes(6,count)&"</td>"
	response.Write " <td>"&arRes(7,count)&"</td>"
	response.Write "</tr>"          next          else
          response.Write "<tr class=aviso>"          response.Write " <td
colspan=8 align=center>No hay depositos pendientes por asociar a una comprobacion</td>"
          response.Write "</tr>"          end if
	response.Write "<tr class=texto_nivel_2>"          response.Write " <td
colspan=8><hr></td>"          response.Write "</tr>"
	response.Write "<tr>"          response.Write " <td colspan=8
align=center><input class=boton type=button value=""Cerrar"" onclick=javascript:window.close();></td>"
                response.Write "</tr>"          response.Write "</table>"
	end function%><html>	<head>	<title>Dep&oacute;sitos Registrados</title>
	<script src="../includes/open_window.js"></script>	<script
src="../includes/no_right_mouse.js"></script>	<link rel="stylesheet"
href="../stylesheet/style.css">	<script language=javascript>	<!--
	function registra_deposito(folio_deposito,monto,idfila){          //Esta
funcion ingresa los datos correspondientes a la pagina que la abrio (../solicitud/solic_detal_captu.asp)
                //Recibe:          //folio_deposito = identificador del
registro del deposito          //monto = monto del deposito
	//idfila = identificador de la fila seleccionada..(el contador)          //
	  esto sirve para seleccionar el mismo registro en caso de que el          //
	      gasto ya haya sido capturado y se pretenda modificarlo
	registro = 'document.form.folio_deposito+idfila+'.checked = true';
	eval(registro);          //Si los datos introducidos estan correctos,
	    //se copian estos datos a la pagina que abrio este catalogo
	//(../solicitud/solic_detal_captu.asp)
	window.opener.document.form.folio_deposito.value = folio_deposito;
	window.opener.document.form.subtotal.value = monto;          //Se ejecuta
en la pagina principal la funcion que calcula los totales
	window.opener.suma_total();          }          -->          </script>
	</head> <body>  <form name="form" method="post">          <input type=hidden
```

```
name=idempresavalue=<%=idempresa%>>  <input type=hidden name=numempleado
value=<%=numempleado%>>  <%  call
deposito(idempresa,numempleado,folio_deposito,idmoneda)  'Funcion que presenta el catalogo de
depositos de un empleado  'Localizacion = esta misma pagina %>  </form> </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../solicitud/solic_grale_captu.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
'Fte_ - -  'Fecha de ultima modificacion  : 6 de julio 2007  'Funcionalidad
general  : Pagina mediante la cual se capturan los datos generales de  '
las solicitudes
'================================================================================
==========  'Estas variables son comunes a todas las paginas  dim
numempleado  'Numero de empleado que esta firmado
en la aplicacion  dim idempresa  'Numero de
empresa en la que se esta trabajando  numempleado = request("numempleado")
idempresa = request("idempresa")
'================================================================================
==========
'================================================================================
==========  'Estas variables se usan nada mas en las paginas de solicitudes
dim folio  'Numero de folio de la solicitud
folio = request("folio")
'================================================================================
==========
'================================================================================
==========  'Estas variables se usan nada mas en las paginas de solicitudes
dim folio_principal  'Numero de folio de la
solicitud principal a la  folio_principal = request("folio_principal")'que la solicitud va asociada
'================================================================================
==========
'================================================================================
==========  'Esta variable almacena el tipo de operacion al que se relaciona la
operacion actual  dim tipo_operacion_principal  tipo_operacion_principal =
request("tipo_operacion_principal")
'================================================================================
==========
'================================================================================
==========  'Estas variables son las que almacenan datos especificos de la solicitud en
proceso  dim idmoneda  'Variable donde se almacena la moneda de la solicitud
dim idmoneda_pago  'Variable donde se almacena la moneda de la solicitud  dim
idterritorio  'Variable donde se almacena el territorio de la solicitud  dim descripcion
'Variable donde se almacena la desripcion de la solicitud
'================================================================================
==========
'================================================================================
==========  'Esta variable almacena la proveniencia de la peticion de las paginas
dim proviene  'Variable que indica de donde viene la peticion de esta pagina
'IA = Insertar un nuevo anticipo en la pagina que sigue
'MA = Mostrar los datos de un anticipo en esta pagina
'MA = Modificar los datos de un anticipo
existente en la pagin siguiente  proviene =  request("proviene")
'================================================================================
==========
'================================================================================
==========  'Variable que almacena el tipo de operacion (anticipo,pago,compra,etc)
dim tipo_operacion  tipo_operacion = request("tipo_operacion")
'================================================================================
==========
'================================================================================
```

```
==========                    'Datos especificos de los anticipos             dim fecha_ini       'Fecha de
inicio del anticipo             dim fecha_fin       'Fecha de fin del anticipo
        '================================================================================
==========
        '================================================================================
==========                    'Datos especificos de los pagos                 dim valor_proveedor
        'Proveedor seleccionado             dim valor_factura'Factura que avala el pago al proveedor
        '================================================================================
==========
        '================================================================================
==========                    'Datos especificos de las compras               dim valor_depto
        'Departamento solicitante de la compra
        '================================================================================
==========
        '================================================================================
==========                    'Variables que almacenan las solicitudes que el usuario puede hacer o
tiene en proceso,             'de acuerdo a las politicas corporativas                dim operaciones_politica
        'Solicitudes permitidas de acuerdo al perfil del uuario             dim operaciones_pendientes
        'Solicitudes pendientes por mandar a autorizar                      dim operaciones_rechazadas
        'Solicitudes rechazadas             dim operaciones_proceso                 'Solicitudes en
proceso de autorizacion             dim operaciones_permitidas                  'Cuantas solicitudes mas
puede hacer el usuario
        'permitidas = politica - (pendientes + rechazadas + proceso)
        '================================================================================
==========%><!--#include file="../connect/conne_open.asp"--><!--       Este archivo contiene la
conexion a la BD--><!--     #include file="../includes/muestr_folio.asp"--><!--       Este archivo incluye la funcion
que muestra el folio de la solicitud al usuario--><!--    #include file="../includes/catal_moned.asp"--><!--
        Este archivo incluye la funcion que presenta el catalogo de monedas--><!--       #include
file="../includes/catal_terri.asp"--><!--           Este archivo incluye la funcion que presenta el catalogo de
territorios--><!--    #include file="../includes/catal_prove.asp"--><!--       Este archivo incluye la funcion que
presenta el catalogo de proveedores--><!--  #include file="../includes/muestr_descr.asp"--><!--     Este archivo
incluye la funcion que presenta la caja de texto de la descripcion--><!--#include
file="../includes/muestr_texto.asp"--><!--      Este archivo incluye la funcion que presenta una caja de texto--
><!--      #include file="../includes/muestr_boton.asp"--><!--     Este archivo incluye la funcion que presenta los
botones--><!--     #include file="../includes/muestr_fecha.asp"--><!--     Este archivo incluye la funcion que
presenta la caja de texto para seleccionar alguna fecha--><!--#include file="../includes/forma_fecha.asp"--><!-
-           Este archivo incluye la funcion da formato a las fechas--><!--  #include
file="../upload/masive_delete.asp"--><!--       Este archivo incluye la funcion que elimina archivos anexos en
caso de que se elimine alguna soicitud--><!--           #include file="../includes/pide_datos_grale.asp"--><!--
        Este archivo incluye la funcion que pide al usuario todos los datos generales de las solicitudes--><!--
        #include file="../anticipo/antic_pide_datos_parti.asp"--><!--     #include
file="../anticipo/antic_carga_datos_parti.asp"--><!--   #include
file="../comprobacion_anticipo/comprobacion_anticipo_pide_datos_parti.asp"--><!--       #include
file="../comprobacion_anticipo/comprobacion_anticipo_carga_datos_parti.asp"--><!--     #include
file="../pago/pago_pide_datos_parti.asp"--><!--         #include file="../pago/pago_carga_datos_parti.asp"--><!--
        #include file="../reservacion/reservacion_pide_datos_parti.asp"--><!--  #include
file="../reservacion/reservacion_carga_datos_parti.asp"--><!--#include
file="../comprobacion_reservacion/comprobacion_reservacion_pide_datos_parti.asp"--><!--       #include
file="../comprobacion_reservacion/comprobacion_reservacion_carga_datos_parti.asp"--><!--      #include
file="../compra/compra_pide_datos_parti.asp"--><!--  #include file="../compra/compra_carga_datos_parti.asp"--
><!--      #include file=../includes/muestr_folio_relac.asp--><!--Este archivo incluye la funcion que muestra los
folios relacionados para ese tipo de solicitud--><%    function
revisa_proceso(idempresa,numempleado,tipo_operacion,clase)                  'Esta funcion obtiene cuantas
solicitudes tiene el usuario:                   'permitidas por realizar, pendientes, rechazadas, en proceso y
restantes                   'Recibe:            'idempresa = empresa en la que se trabaja
        'numempleado = numero de empleado que hace la solicitud             'tipo_operacion = tipo de
operacion que se solicita (anticipo, pago, compra, etc)                     dim sql             if tipo_operacion =
"comprobacion_anticipo" or tipo_operacion = "comprobacion_reservacion" then
        operaciones_permitidas = 1                  else                    sql = " exec
[dbo].sp_revisa_operaciones_en_proceso "&idempresa&","&numempleado&",'"&tipo_operacion&"'"
                set res = connect.execute(sql)                              if not res.eof then
        operaciones_politica = res(0)                                       operaciones_pendientes = res(1)
```

```
                operaciones_rechazadas = res(2)                operaciones_proceso
= res(3)                operaciones_permitidas = res(4)
        response.Write "<tr>"                response.Write "  <td class=tabla_nivel_2
colspan=2>Pol&iacute;ticas corporativas</td>"                response.Write "</tr>"
                response.Write "<tr>"                response.Write "  <td
colspan=2>"                response.Write "  <table border=0 class=contenedor
width=""100%"">"                response.Write "  <tr>"
        response.Write "                <td width=300 class="&clase&">L&iacute;mite de
operaciones en proceso</td>"                response.Write "                <td
class=texto_nivel_2_dato>"&operaciones_politica&"</td>"                response.Write "
                </tr>"                response.Write "                <tr>"
                response.Write "                <td class="&clase&">Solicitudes nuevas</td>"
                response.Write "                <td
class=texto_nivel_2_dato>"&operaciones_pendientes&"</td>"                response.Write "
                </tr>"                response.Write "                <tr>"
                response.Write "                <td class="&clase&">Solicitudes
rechazadas</td>"                response.Write "                <td
class=texto_nivel_2_dato>"&operaciones_rechazadas&"</td>"
        response.Write "                </tr>"                response.Write "
        <tr>"                response.Write "                <td
class="&clase&">Solicitudes en proceso</td>"                response.Write "
                <td class=texto_nivel_2_dato>"&operaciones_proceso&"</td>"
        response.Write "                </tr>"                response.Write "
        <tr>"                response.Write "                <td
class="&clase&">Solicitudes permitidas</td>"                response.Write "
                <td class=texto_nivel_2_dato>"&operaciones_permitidas&"</td>"
        response.Write "                </tr>"                response.Write "
        </table>"                response.Write "  </td>"
        response.Write "</tr>"                else                operaciones_politica
        = 0                operaciones_pendientes  = 0
        operaciones_rechazadas  = 0                operaciones_proceso                = 0
                operaciones_permitidas  = 0                end if                end
if        end function        function obtiene_folios(tipo_status,idempresa,numempleado,tipo_operacion)
        'Esta funcion obtiene los folios de las solicitudes:                'pendientes, rechazadas y en proceso
                'Recibe:                'tipo_status = tipo de solicitud que se busque                '
                n = pendiente (solicitud Nueva)                '                r =
rechazadas (Rechazadas)                '                e = proceso (Espera de autorizacion)
                'idempresa = empresa en la que se trabaja                'numempleado = numero de empleado
que hace la solicitud                'tipo_operacion = tipo de operacion que se solicita (anticipo, pago, compra,
etc)                dim sql                dim t_arRes                dim res                dim count                dim
inicial_status                dim liga                dim eliminar                dim titulo                dim
liga_folio_abre                dim liga_folio_cierra                dim liga_titulo_abre                dim
liga_titulo_cierra                inicial_status = mid(tipo_status,1,1)                sql = "exec [dbo].sp_obtiene_folios
"&idempresa&", "&numempleado&", '"&tipo_operacion&"', '"&inicial_status&"'                set res =
connect.execute(sql)                if not res.eof then                arRes = res.getrows
        t_arRes = ubound(arRes,2)                else                t_arRes = -1                end
if                if t_arRes >= 0 then                if ucase(inicial_status) <> "E" then
                liga = 1                eliminar = 1                titulo =
"Eliminar"                else                eliminar = 0
        liga = 0                titulo = "ver detalle"                end if
        response.Write "<table width=""100%"" border=0 align=center>"
        response.Write "  <tr>"                response.Write "  <td
class=tabla_nivel_2>Solicitudes "&tipo_status&"</td>"                response.Write "  </tr>"
                response.Write "</table>"                response.Write "<table class=contenedor
width=""99.5%"" align=center>"                response.Write "  <tr class=tabla_nivel_3>"
        response.Write "        <td>Folio</td>"                response.Write "        <td>Fecha
de solicitud</td>"                response.Write "        <td>Descripcion</td>"
        response.Write "        <td>Total</td>"                response.Write "
        <td>Estado</td>"                response.Write "        <td>"&titulo&"</td>"
        response.Write "  </tr>"                for count = 0 to t_arRes                if
liga = 1 then                liga_folio_abre = "<a
href=javascript:carga_datos("&arRes(0,count)&",'"&tipo_operacion&"');>"
```

```
        liga_folio_cierra = "</a>"                                  liga_titulo_abre = "<a
href=javascript:elimina_solicitud("                                 liga_titulo_cierra = ");><img
src=../images/cruz.gif border=0 alt=Eliminar></a>"                          else
                liga_titulo_abre = "<a href=javascript:ver_detalle("
        liga_titulo_cierra = ");>Ver detalle</a>"                           end if
        response.Write "<tr class=texto_nivel_2>"                   response.Write " <td>"
                        response.Write                  liga_folio_abre
        response.Write                          arRes(0,count)
        response.Write                  liga_folio_cierra                       response.Write "
        </td>"                          response.Write " <td>"&formato_fecha(arRes(1,count))&"</td>"
                                response.Write " <td>"&arRes(2,count)&"</td>"
        response.Write " <td>"&formatcurrency(arRes(3,count),2)&"</td>"
        response.Write " <td>"&arRes(4,count)&"</td>"                       response.Write "
        <td>"                           response.Write                  liga_titulo_abre
                response.Write                          arRes(0,count)
        response.Write                  liga_titulo_cierra                      response.Write "
        </td>"                          response.Write "</tr>"                  next
        response.Write "</table>"       else            end if  end function    function
carga_datos_grales(folio)       'Esta funcion carga los datos generales de una solicitud cuando se
selecciona              'para ser modificada o ver su detalle           'Recibe:        'folio =
numero de folio de la solicitud         dim sql         dim res         sql = "exec
[dbo].sp_obtiene_datos_grales "&folio&" "        set res = connect.execute(sql)         if not res.eof
then                    idmoneda = res(0)                       idmoneda_pago = res(1)
        idterritorio = res(2)                   descripcion = res(3)            else
        idmoneda = ""                   idmoneda_pago = ""                      idterritorio = ""
                descripcion = ""        end if  end function    function elimina_solicitud(folio)
        'Funcion a traves de la cual se eliminan los archivos anexos relacionados a una solicitud
        'Recibe:         'Folio = numero de folio de la solicitud                dim sql         'Funcion que
propiamente elimina los archivos anexos         'Localizacion = "../upload/masive_delete.asp"
        sql = "exec sp_solicitud_inserta_modifica "&folio&", 0,0,",0,0,0,0,'','d'"      connect.execute(sql)
                folio = "" end function 'Ocurre cuando se ModificA una solicitud        if trim(proviene) =
"MA" then               call carga_datos_grales(folio)          select case tipo_operacion
                case "anticipo"                                         :       call
carga_datos_anticipo(folio)
                                'Funcion que muestra los datos particulares de los anticipos
        case "pago"                                                     :       call
carga_datos_pago(folio)
                        'Funcion que muestra los datos particulares de los pagos
        case "reservacion"                                      :       call
carga_datos_reservacion(folio)
                                'Funcion que muestra los datos particulares de las reservaciones
                case "comprobacion_reservacion"         :       call
carga_datos_comprobacion_reservacion(folio_principal)
                                                        'Funcion que muestra los datos particulares de
las reservaciones               case "compra"                                           :       call
carga_datos_compra(folio)
                                'Funcion que muestra los datos particulares de las compras      end
select  end if  'Ocurre cuando se Elimina una Solicitud  if trim(proviene) = "ES" then          call
elimina_solicitud(folio)        'Localizacion = esta misma pagina end if%><html>  <head>
        <title>Captura de Solicitud</title>             <script language=javascript type="text/javascript"
src="../includes/open_windo.js"></script>               <script language=javascript type="text/javascript"
src="../includes/calen_picke.js"></script>              <script language=javascript type="text/javascript"
src="../includes/valida_fechas.js"></script>            <script language=javascript type="text/javascript"
src="../includes/selecciona_monedas.js"></script>               <link rel="stylesheet" media=screen
href="../stylesheet/style.css">         <link rel="stylesheet" media=print href="../stylesheet/print.css">
                <script language=javascript>            <!--                    function
insertar(tipo_operacion,valor,operaciones_permitidas,fecha_hoy){                       //Esta
funcion valida ciertos aspectos antes de mandar a insertar una                  //nueva
solicitud                       //Recibe:                       //tipo_operacion = tipo de
solicitud (anticipo,pago,compra,etc)                    //valor de proveniencia = (IA = InsertAr Solicitud,
MA = ModificA solicitud )               //Operaciones_permitidas = cuantas operaciones mas
puede hacer el usuario                  //fecha_hoy = fecha actual (fecha del servidor, no fecha
```

```
local)                                  //Que el usuario pueda hacer mas solicitudes
        if((tipo_operacion=='comprobacion_anticipo')||(tipo_operacion=='comprobacion_reservacion')){
                                        operaciones_permitidas = 1;                                     }
                                if (operaciones_permitidas<=0){                                         alert('Por el
momento, usted no puede hacer mas operaciones');                                        return false;
                                }                                       if(document.form.idmoneda.value==0){
                                        alert('Favor de seleccionar la moneda de la solicitud');
                                document.form.idmoneda.focus();                                         return false;
                                        }
        if(document.form.idmoneda_pago.value==0){                                               alert('Favor
de seleccionar la moneda del pago');
        document.form.idmoneda_pago.focus();                                            return false;
                                }                                       if(document.form.idterritorio.value==0){
                                        alert('Favor de seleccionar el territorio');
        document.form.idterritorio.focus();                                     return false;
                }                                       if(document.form.descripcion.value==''){
                                alert('Favor de introducir una breve descripcion');
        document.form.descripcion.focus();                                      return false;
                }                                       if(document.form.descripcion.value.length>250){
                                        alert('La descripcion no puede ser mayor a 250 caracteres');
                                document.form.descripcion.focus();                                      return false;
                                        }                               //Si se trata de una solicitud de
anticipo, entonces valida lo siguiente:                                 if(tipo_operacion == 'anticipo'){
                                        if(document.form.fecha_ini.value==''){
                alert('Favor de introducir la fecha de inicio');
        document.form.fecha_ini.focus();                                                return false;
                                        }
        if(document.form.fecha_fin.value==''){                                                  alert('Favor
de introducir la fecha de fin');
        document.form.fecha_fin.focus();                                                return false;
                                        }                               //Que la fecha de inicio del
anticipo no sea menor a la fecha de hoy                                 //Localizacion =
"../includes/valida_fechas.js"
        if(!valida_fechas(fecha_hoy,document.form.fecha_ini.value)){
        alert('Fecha de hoy mayor a fecha de inicio');
        return false;                                   }
        //Que la fecha de fin no sea menor a la fecha de inicio
        //Localizacion = "../includes/valida_fechas.js"
        if(!valida_fechas(document.form.fecha_ini.value,document.form.fecha_fin.value)){
                                        alert('Fecha de inicio mayor a fecha de fin');
                return false;                                   }                                       }
                                        if(tipo_operacion=='comprobacion_anticipo'){
                if(document.form.folio_principal.value==''){
        alert('Favor de asociar esta comprobacion con un anticipo');
        return false;                                   }                                       }
                                if(tipo_operacion == 'pago'){
        if(document.form.idproveedor.value==0){                                                 alert('Favor
de seleccionar un proveedor');
        document.form.idproveedor.focus();                                                      return false;
                                        }
        if(document.form.no_factura.value==''){                                                 alert('Favor
de introducir el numero de la factura');
        document.form.no_factura.focus();                                               return false;
                                        }
        if(document.form.no_factura.value.length>10){
        alert('El numero de la factura no puede ser mayor a 10 caracteres');
                document.form.no_factura.focus();                                               return false;
                                        }                               }
        if(tipo_operacion == 'compra'){
        if(document.form.idproveedor.value==0){                                                 alert('Favor
de seleccionar un proveedor');
        document.form.idproveedor.focus();                                                      return false;
```

```
                                        }
        if(document.form.depto.value==0){                                       alert('Favor de
especificar el departamento que hace la solicitud');
        document.form.depto.focus();                                            return false;
                                        }                               }
        if(tipo_operacion == 'reservacion'){
        if(document.form.idproveedor.value==0){                                         alert('Favor
de seleccionar un proveedor');
        document.form.idproveedor.focus();                                      return false;
                                        }                               }
        if(tipo_operacion=='comprobacion_reservacion'){
        if(document.form.folio_principal.value==''){                                    alert('Favor
de asociar esta comprobacion con una reservacion');                             return false;
                                        }                               }
        //Se manda a la pagina del detalle                      document.form.proviene.value = valor;
                                document.form.submit();                 }
        function limpiar_forma(tipo_operacion,idempresa,numempleado){
        //Funcion que limpia la forma de captura                        document.form.proviene.value
= '';                           document.form.folio.value = '';
        document.form.folio_principal.value = '';
        document.form.tipo_operacion.value = tipo_operacion;
        document.form.idempresa.value = idempresa;
        document.form.numempleado.value = numempleado;
        document.form.action = 'solic_grale_captu.asp';
        document.form.submit();                 }               function carga_datos(folio){
                                //Esta funcion carga los datos de la solicitud en caso se que se quiera
                        //modificar o entrar al detalle                         //Recibe:
                        //folio = numero de folio de la solicitud                       //se asigna a
proviene el valor MA (ModificA solicitud)                       document.form.folio.value = folio;
                        document.form.proviene.value = 'MA';
        document.form.action = 'solic_grale_captu.asp';
        document.form.submit();                 }               function
elimina_solicitud(folio){               //Funcion a traves de la cual se eliminan las solicitudes
                                //Recibe:                               //folio = numero de folio de la
solicitud                               //se asigna a proviene el valor ES (Elimina Solicitud)
        document.form.folio.value = folio;                      document.form.proviene.value = 'ES';
                                document.form.action = 'solic_grale_captu.asp';
        document.form.submit();                 }               function ver_detalle(folio){
                        //Funcion a traves de la cual se accede al detalle de la solicitud
                //La solicitud tiene que estar en proceso de autorizacion para que
        //se tenga acceso a esta funcionalidad                          //Recibe:
                //folio = numero de folio de la solicitud                       //se asigna a
proviene el valor US (USuario). Asi se restringen las operaciones                       //que el
usuario tiene dentro del detalle de la solicitud                        tipo_operacion =
document.form.tipo_operacion.value;                     proviene = 'US';
        open_window('solic_ver_detal.asp?folio='+folio+'&tipo_operacion='+tipo_operacion+'&proviene='+pro
viene+'&ver_menu=0','Detalle','status=yes,resizable=yes,scrollbars=yes');                       }
        -->             </script></head> <body>  <!--#include file=../menu/menu.asp-->       <form
name="form" method="post" enctype=application/x-www-form-urlencoded>           <table border=0
width=90% align=center>                 <tr>                    <td colspan=2>
                        <table border=0 width=100%>                             <%
                        if folio = "" then                                      call
revisa_proceso(idempresa,numempleado,tipo_operacion,"texto_nivel_2")
                'Funcion que revisa las solicitudes en proceso
        'Localizacion = esta misma pagina                       else
                        select case tipo_operacion
        case "comprobacion_anticipo"    : tipo_operacion_principal = "anticipo"
                        case "pago"                                             :
tipo_operacion_principal = "compra"                                             case
"comprobacion_reservacion"      : tipo_operacion_principal = "reservacion"
                end select                                              call
muestra_folio_relaciones(folio,"5")                     'Funcion que verifica con que
```

```
otros folio se relaciona el folio propuesto                                                    'Localizacion =
"../includes/muestr_folio_relac.asp"                                                           end if
        %>                                                  </table>                           </td>
        </tr>                       <tr>                    <td colspan=2>
                <table border=0 width=100%>                                                    <tr>
                                                            <td class=tabla_nivel_2 colspan=5>Informaci&oacute;n
General</td>                                                </tr>
        <tr>                                                <td valign=top>
                                            <table border=0 width=100% class=contenedor>
                                                            <tr>
                                            <td>
                        <%
                'Esta funcion pide al usuario todos los datos generales de todas las solicitudes
                                                                        call
pide_datos_grale(folio,idmoneda,idmoneda_pago,idterritorio,descripcion,tipo_operacion,"texto_nivel_2")
                                                                        'Localizacion =
"../includes/pide_datos_grale.asp"
        %>                                                                                     </td>
                                                            </tr>
                                    </table>                                                   </td>
                                            <%
        if tipo_operacion <> "comprobacion_gasto"  then
        %>                                                  <td valign=top>
                                            <table border=0 width=100% class=contenedor>
                                                            <tr>
                                            <td>
                        <%
                'Esta funcion pide al usuario los datos particulares de cada tipo de solicitud
                                                                        select case tipo_operacion

        case "anticipo"                                 :       call
antic_pide_datos_parti(fecha_ini,fecha_fin,"texto_nivel_2")

                        'Localizacion = "../anticipo/antic_pide_datos_parti.asp"
                                                                        case "comprobacion_anticipo"
        :       call comprobacion_anticipo_pide_datos_parti(idempresa,numempleado,"texto_nivel_2")

                                                                        'Localizacion =
"../comprobacion_anticipo/comprobacion_anticipo_pide_datos_parti.asp"
                                                                        case "pago"
                        :       call
pago_pide_datos_parti(idempresa,numempleado,valor_proveedor,valor_factura,"texto_nivel_2")

                                                            'Localizacion =
"../pago/pago_pide_datos_parti.asp"
                                    case "reservacion"                  :       call
reservacion_pide_datos_parti(idempresa,numempleado,valor_proveedor,"texto_nivel_2")

                                                            'Localizacion =
"../reservacion/reservacion_pide_datos_parti.asp"
                                            case "comprobacion_reservacion"  :      call
comprobacion_reservacion_pide_datos_parti(idempresa,numempleado,valor_proveedor,"texto_nivel_2")

                                                                        'Localizacion =
"../comprobacion_reservacion/comprobacion_reservacion_pide_datos_parti.asp"
                                                                        case "compra"
                        :       call
compra_pide_datos_parti(idempresa,valor_proveedor,valor_depto,"texto_nivel_2")

                                                    'Localizacion = "../compra/compra_pide_datos_parti.asp"
                                                                        end select
                                                            %>
```

```
                                                                    </td>
                                                </tr>
        </table>                                                    </td>
                                <%                                                              end if
                                                %>                                                              </tr>
                                        </table>                    </td>                                       </tr>
                        <tr>                                <td colspan=2 align=center>
        <%                                                                                      call
muestra_boton("Limpiar forma","limpiar_forma('"&tipo_operacion&"',"&idempresa&","&numempleado&");")
                                response.Write "   "                       if folio = "" then
                                                                        call muestra_boton("Capturar
detalle","insertar('"&tipo_operacion&"','IA',"&operaciones_permitidas&",'"&day(date)&"/"&month(date)&"/"&year
(date)&"');")                                   else
                        call muestra_boton("Capturar
detalle","insertar('"&tipo_operacion&"','MA',1,'"&day(date)&"/"&month(date)&"/"&year(date)&"');")
                                                                        'Funcion que presenta los botones
                                                                        'Localizacion = "../includes/muestr_boton.asp"
                                        end if                          %>
        </td>                   </tr>                   <tr>                                    <td
colspan=2>                                      <table align=center width=100%>
                                <%                                                              call
obtiene_folios("nuevas",idempresa,numempleado,tipo_operacion)
                        call obtiene_folios("rechazadas",idempresa,numempleado,tipo_operacion)
                                                call obtiene_folios("en espera de
autorizacion",idempresa,numempleado,tipo_operacion)
        'Funcion que obtiene los folios de las solicitudes
        'Localizacion = esta misma pagina                                                       %>
                                </table>                        </td>                   </tr>
        </table>        <input type=hidden name=idempresa
        value=<%=idempresa%>>           <input type=hidden name=numempleado
                        value=<%=numempleado%>>         <input type=hidden name=tipo_operacion
                                value=<%=tipo_operacion%>>              <input type=hidden name=proviene
                                        value=<%=request("proviene")%>>         <input type=hidden
name=folio                                                      value=<%=folio%>>               <input
type=hidden name=folio_principal                                value=<%=folio_principal%>>
        <input type=hidden name=tipo_operacion_principal  value=<%=tipo_operacion_principal%>>
        </form>  </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../solicitud/solic_ver_detal.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -               'Fecha de ultima modificacion       : 6 de julio 2007           'Funcionalidad
general                         : Pagina que muestra el detalle de las solicitudes
        '================================================================================
===============                 'Estas variables son comunes a todas las paginas            dim
numempleado                                                     'Numero de empleado que esta firmado
en la aplicacion            dim idempresa                                                   'Numero de
empresa en la que se esta trabajando            numempleado = request("numempleado")
        idempresa = request("idempresa")
        '================================================================================
===============
        '================================================================================
===============                 'Estas variables son comunes a todas las paginas de solicitudes
        dim idmoneda                                                    'Moneda de la solicitud
        dim idmoneda_pago                                               'Moneda en que se pretende
se realice el pago          dim tipo_operacion                                              'Tipo de
solicitud (anticipo,pago,compra,etc)        dim idterritorio
        'Territorio en que se hizo o se hara el gasto       idmoneda = request("idmoneda")
        idmoneda_pago = request("idmoneda_pago")                tipo_operacion =
request("tipo_operacion")       idterritorio = request("idterritorio")
```

```
'==========================================================================
==========
'==========================================================================
==========  'Estas variables se usan nada mas en las paginas de solicitudes
dim folio                                                   'Numero de folio de la solicitud
    folio = request("folio")
'==========================================================================
==========
'==========================================================================
==========  'Variables de desplegado de informacion dentro del detalle  dim
despliega_distribucion  'Variable que guarda si se quiere mostrar la distribucion de los
                                                    'detalles entre los centros de costo
despliega_distribucion = request("despliega_distribucion")  if despliega_distribucion = ""
then  despliega_distribucion = 0  end if  dim despliega_informacion
'Variable que guarda si se quiere mostrar informacion adicional de los
                                    'detalles (informacion adicional y anexos)
despliega_informacion = request("despliega_informacion")  if despliega_informacion = ""
then  despliega_informacion = 0  end if  'Ambas variables se inicializan
en cero
'==========================================================================
==========
'==========================================================================
==========  'Esta variable almacena la proveniencia de la peticion de las paginas
dim proviene  'Variable que indica de donde viene la peticion de esta pagina
                    'AU = El detalle se solicita a traves de la pagina de
AUtorizaciones                                  'US = El detalle lo solicita el USuario
que hace la solicitud  proviene =  request("proviene")
'==========================================================================
==========
'==========================================================================
==========  'Estas variables se usan nada mas en las paginas de solicitudes
dim folio_principal                                         'Numero de folio de la
solicitud principal a la  folio_principal = request("folio_principal")'que la solicitud va asociada
'==========================================================================
==========
'==========================================================================
==========  'Esta variable almacena el status actual de la solicitud, solo se utiliza si la
peticion  'de ver el detalle viene de la pagina de autorizaciones  dim status
status = request("status")
'==========================================================================
==========
'==========================================================================
==========  'Esta variable evalua si se muestra o no el menu  dim
ver_menu  ver_menu = request("ver_menu")
'==========================================================================
==========%><!--#include file="../connect/conne_open.asp"--><!--  Este archivo contiene la
conexion a la BD--><!--  #include file="../includes/muestr_folio.asp"--><!--  Este archivo incluye la funcion
que muestra el folio de la solicitud al usuario--><!--  #include file="../includes/muestr_datos_grale.asp"--><!--
Archivo que incluye la funcion que muestra los datos generales de las solicitudes--><!-- #include
file="../includes/muestr_descr.asp"--><!--  Este archivo incluye la funcion que presenta la caja de texto de la
descripcion--><!--#include file="../includes/muestr_boton.asp"--><!--  Este archivo incluye la funcion que
presenta los botones--><!--  #include file="../includes/muestr_tipo_conce.asp"--><!--  Este archivo
incluye la funcion que pide al usuario ciertos datos dependiendo del tipo de concepto del concepto--><!--
#include file="../includes/forma_fecha.asp"--><!--  Este archivo incluye la funcion que da formatoa
las fechas--><!--  #include file="../anticipo/antic_muestr_datos_parti.asp"--><!-- #include
file="../pago/pago_muestr_datos_parti.asp"--><!--  #include
file="../reservacion/reservacion_muestr_datos_parti.asp"--><!--  #include
file="../compra/compra_muestr_datos_parti.asp"--><!--  #include file=../includes/muestr_folio_relac.asp--
><!--  Este archivo incluye la funcion que muestra los folios relacionados para ese tipo de solicitud--><%
function opciones_proveniencia(status,folio,proviene)  'Esta funcion presenta varias opciones
dependiendo de la  'proveniencia de la solicitud para ver el detalle  'Recibe:
'status = status de la solicitud, solo aplica cuando proviene = AU  'folio = folio de la
```

```
solicitud		'proviene = de donde viene la peticion de ver el detalle		'
	AU = de la pagina de AUtorizaciones		'		US = del USuario que
hace la solicitud		dim sql		dim res		dim status_solicitud
	response.Write "<table border=0 width=""100%"">"		select case proviene
			'Si la peticion proviene de la pagina de AUtorizaciones
	case "AU":		response.Write " <tr>"
				response.Write "	<td class=tabla_nivel_2
colspan=1>"		response.Write "
	Flujo de autorizacion"		response.Write "
	</td>"		response.Write "	<td
class=tabla_nivel_2 colspan=1>"		response.Write "
		Como se ver&iacute;a el registro si..."
	response.Write "	</td>"
	response.Write " </tr>"		response.Write "
	<tr>"		response.Write "	<td>"
				response.Write "	<table border=0>"
					response.Write "
	<tr>"		response.Write "
		<td class=texto_nivel_2>"
	response.Write "		Comentarios"
			response.Write "		</td>"
				response.Write "		</tr>"
				response.Write "		<tr>"
				response.Write "			<td>"
				response.Write "
	<textarea name=comentario cols=40 rows=5 class=caja_texto></textarea>"
			response.Write "		</td>"
				response.Write "		</tr>"
				response.Write "	</table>"
				response.Write "	</td>"
			response.Write "	<td valign=top>"
		response.Write "	<table border=0>"
			response.Write "		<tr class=texto_nivel_2>"
				response.Write "			<td>"
				response.Write "
	<a
href=javascript:open_window('../flujo_autoriza/flujo_prono_movim.asp?status="&status&"&numempleado="&nu
mempleado&"&idempresa="&idempresa&"&folio="&folio&"&opcion=a','title','scrollbars=yes,status=no,height=4
00,width=700,resizable=yes');>...autorizo?</a>"
	response.Write "		</td>"
		response.Write "		</tr>"
		response.Write "		<tr class=texto_nivel_2>"
			response.Write "		<td>"
				response.Write "			<a
href=javascript:open_window('../flujo_autoriza/flujo_prono_movim.asp?status="&status&"&numempleado="&nu
mempleado&"&idempresa="&idempresa&"&folio="&folio&"&opcion=r','title','scrollbars=yes,status=no,height=40
0,width=700,resizable=yes');>...rechazo?</a>"
	response.Write "		</td>"
		response.Write "		</tr>"
		response.Write "	</table>"
		response.Write "	</td>"
	response.Write " </tr>"		response.Write "
	<tr>"		response.Write "	<td
colspan=2 align=center>"		response.Write "
	<hr>"		response.Write "	</td>"
				response.Write " </tr>"
			response.Write " <tr>"
	response.Write "	<td colspan=2 align=center>"
				call
muestra_boton("Autorizar","javascript:registra_aut_o_rec('A','"&status&"');")
			response.Write "		 "
				call
```

```
muestra_boton("Rechazar","javascript:registra_aut_o_rec('R','"&status&"');")
                                response.Write "                                 "
                                                                                call
muestra_boton("Regresar","regresar_a_pantalla_autoriza();")
        response.Write "                         "
                                                                                call muestra_boton("Imprimir","window.print();")

                'Funcion que muestra los botones
                                                                                'Localizacion = "../includes/muestr_boton.asp"
                                                                                response.Write "                </td>"
                                                        response.Write " </tr>"                                 'Si la
peticion viene del USuario que hace la solicitud                                case "US":
                                                        response.Write " <tr>"
                response.Write "                <td colspan=2 align=center>"
                        response.Write "                                <hr>"
                response.Write "                </td>"
        response.Write " </tr>"                                                                 response.Write "
        <tr>"                                                                   response.Write "                <td
colspan=2 align=center>"
                                        call muestra_boton("Cerrar","window.close();")
                                        response.Write "                         "
                                                                                call
muestra_boton("Imprimir","window.print();")
                                                        'Funcion que muestra los botones
                                                                                'Localizacion
= "../includes/muestr_boton.asp"                                                response.Write "
        </td>"                                                                  response.Write " </tr>"
        end select              response.Write "</table>" end function      function
muestra_anexos(detalle)         'Esta funcion muestra los anexos de la solicitud                'Recibe:
        'detalle = numero del detalle dentro de la solicitud                    dim sql         dim res         dim
arRes           dim t_arRes             dim count               dim plural              dim
tamano_total            sql = "exec [dbo].sp_muestra_anexos "&detalle&" "       set res =
connect.execute(sql)            if not res.eof then                     arRes = res.getrows
        t_arRes = ubound(arRes,2)               else                    t_arRes = -1            end
if              if t_arRes >= 0 then                    tamano_total = 0
        response.Write " <tr class=tabla_nivel_4>"                      response.Write "        <td>"
                response.Write "                        <img src=../images/clip.gif border=0>"
        response.Write "        </td>"                  response.Write "        <td>"
        response.Write "                Anexo"          response.Write "        </td>"
                response.Write "        <td>"           response.Write "
        Tama&ntilde;o"                  response.Write "        </td>"
        response.Write "        <td>"                   response.Write "                Tipo"
                response.Write "        </td>"          response.Write "        <td>"
                response.Write "                        Fecha de creaci&oacute;n"
        response.Write "        </td>"                  response.Write " </tr>"                 for
count = 0 to t_arRes                                    tamano_total = tamano_total + CSng(arRes(3,count))
                                response.Write " <tr class=texto_nivel_2>"
        response.Write "        <td>"                   response.Write
        count + 1                                       response.Write "        </td>"
        response.Write "        <td>"                   response.Write "                <a
href=javascript:mostrar_anexo('"&application("dir_logica")&arRes(2,count)&"');>"&arRes(2,count)&"</a>"
                        response.Write "        </td>"                          response.Write "
        <td>"                           response.Write                          arRes(3,count) & "
KB"                             response.Write "        </td>"
        response.Write "        <td>"                   response.Write
        arRes(4,count)                                  response.Write "        </td>"
        response.Write "        <td>"                   response.Write
        formato_fecha(arRes(5,count))                   response.Write "        </td>"
                        response.Write " </tr>"         next                    if t_arRes =
0 then                          plural = ""             else
        plural = "s"                    end if          response.Write " <tr>"
        response.Write "        <td colspan=6><hr></td>"                response.Write " </tr>"
```

```
                response.Write " <tr class=texto_nivel_2>"                response.Write "          <td
colspan=2>"&t_arRes + 1 &" archivo"& plural & " anexo" & plural &"</td>"
        response.Write "          <td colspan=1>"&tamano_total&" KB </td>"
        response.Write " </tr>"          else          end if     end function      function
muestra_informacion(folio_detalle)          'Esta funcion muestra informacion adicional del detalle capturado
                'Todo depende del tipo de concepto que se haya introducido en ese detalle.          'Por
ejemplo: si fue un concepto que pide al usuario el numero de dias,          'entonces esta funcion,
presenta los numeros de dias como informacion          'adicional no relevante          'Recibe:
        'detalle = numero del detalle dentro de la solicitud          dim sql          dim res          dim
ask_boleto_avion          dim origen_boleto_avion          dim destino_boleto_avion          dim
aerolinea          dim hora_salida          dim ask_kilometraje          dim origen
        dim destino          dim distancia          dim no_viajes          dim total_kilometros
                dim costo_por_kilometro          dim ask_precio_unitario_cantidad          dim
precio_unitario          dim unidades          dim ask_dias          dim dias          dim
ask_comensales          dim comensales          dim ask_deposito          dim referencia          dim
subtotal_diario          dim monto_limite          dim titulo          sql = "exec
[dbo].sp_obtiene_informacion_detallada "&folio_detalle&" "          set res = connect.execute(sql)
        if not res.eof then          ask_boleto_avion = res("ask_boleto_avion")
        origen_boleto_avion = res("origen_boleto_avion")          destino_boleto_avion =
res("destino_boleto_avion")          aerolinea = res("aerolinea")
        hora_salida = res("hora_salida")          ask_kilometraje = res("ask_kilometraje")
                origen = res("origen")          destino = res("destino")
        distancia = res("distancia")          no_viajes = res("no_viajes")
        total_kilometros = res("total_kilometros")          costo_por_kilometro =
res("costo_por_kilometro")          ask_precio_unitario_cantidad =
res("ask_precio_unitario_cantidad")          precio_unitario = res("precio_unitario")
                unidades = res("unidades")          ask_dias = res("ask_dias")
        dias = res("dias")          ask_comensales = res("ask_comensales")
        comensales = res("comensales")          ask_deposito = res("ask_deposito")
        referencia = res("referencia")          subtotal_diario = res("subtotal_diario")
                monto_limite = res("monto_limite")          else          ask_boleto_avion = 0
                        origen_boleto_avion = ""          destino_boleto_avion = ""
        aerolinea = ""          hora_salida = ""          ask_kilometraje = 0
                origen = 0          destino = 0          distancia = 0
                no_viajes = 0          total_kilometros = 0
        costo_por_kilometro = 0          ask_precio_unitario_cantidad = 0
        precio_unitario = 0          unidades = 1          ask_dias = 0
                dias = 1          ask_comensales = 0          comensales = 1
                ask_deposito = 0          referencia = ""          subtotal_diario = 0
                        monto_limite = -1          end if          response.Write "<table border=0
width=""100%"">"          response.Write " <tr>"          response.Write "          <td
class=tabla_nivel_4 colspan=9>Informaci&oacute;n detallada</td>"          response.Write " </tr>"
        if ask_deposito = 1 then          response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"          response.Write "          Referencia de deposito"
                response.Write " </td>"          response.Write " <td>"
        response.Write          referencia          response.Write " </td>"
                response.Write "</tr>"          end if          if ask_boleto_avion = 1 then
                response.Write "<tr class=texto_nivel_2>"          response.Write " <td>"
                response.Write "          Origen"          response.Write " </td>"
        response.Write " <td>"          response.Write          origen_boleto_avion
                        response.Write " </td>"          response.Write "</tr>"
        response.Write "<tr class=texto_nivel_2>"          response.Write " <td>"
        response.Write "          Destino"          response.Write " </td>"
        response.Write " <td>"          response.Write          destino_boleto_avion
                        response.Write " </td>"          response.Write "</tr>"
        response.Write "<tr class=texto_nivel_2>"          response.Write " <td>"
        response.Write "          Aerolinea"          response.Write " </td>"
        response.Write " <td>"          response.Write          aerolinea
                response.Write " </td>"          response.Write "</tr>"
        response.Write "<tr class=texto_nivel_2>"          response.Write " <td>"
        response.Write "          Hora de salida"          response.Write " </td>"
        response.Write " <td>"          response.Write          hora_salida
```

```
                response.Write " </td>"                         response.Write "</tr>"                  end if
        if ask_kilometraje = 1 then                     response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"                          response.Write "        Origen"
        response.Write " </td>"                         response.Write " <td>"
        response.Write                  origen                  response.Write " </td>"
        response.Write "</tr>"                          response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"                          response.Write "        Destino"
        response.Write " </td>"                         response.Write " <td>"
        response.Write                  destino                 response.Write " </td>"
        response.Write "</tr>"                          response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"                          response.Write "        Distancia"
        response.Write " </td>"                         response.Write " <td>"
        response.Write                  distancia               response.Write " </td>"
                response.Write "</tr>"                  response.Write "<tr class=texto_nivel_2>"
                response.Write " <td>"                  response.Write "        No. de viajes"
                response.Write " </td>"                 response.Write " <td>"
        response.Write                  no_viajes               response.Write " </td>"
                response.Write "</tr>"                  response.Write "<tr class=texto_nivel_2>"
                response.Write " <td>"                  response.Write "        Total de kilometros"
                        response.Write " </td>"                 response.Write " <td>"
        response.Write                  total_kilometros        response.Write " </td>"
                response.Write "</tr>"                  response.Write "<tr class=texto_nivel_2>"
                response.Write " <td>"                  response.Write "        Costo por kilometro"
                        response.Write " </td>"                 response.Write " <td>"
        response.Write                  formatcurrency(costo_por_kilometro,2)
        response.Write " </td>"                         response.Write "</tr>"          end if          if
ask_precio_unitario_cantidad = 1 then                   response.Write "<tr class=texto_nivel_2>"
                response.Write " <td>"                  response.Write "        Precio Unitario"
                response.Write " </td>"                 response.Write " <td>"
        response.Write                  formatcurrency(precio_unitario,2)
        response.Write " </td>"                         response.Write "</tr>"
        response.Write "<tr class=texto_nivel_2>"               response.Write " <td>"
        response.Write "        Unidades"                       response.Write " </td>"
        response.Write " <td>"                          response.Write                  unidades
                response.Write " </td>"                 response.Write "</tr>"          end if
        if ask_dias = 1 then                            response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"                          response.Write "        No. de d&iacute;as"
                response.Write " </td>"                 response.Write " <td>"
        response.Write                  dias                    response.Write " </td>"
        response.Write "</tr>"          end if          if ask_comensales = 1 then
        response.Write "<tr class=texto_nivel_2>"               response.Write " <td>"
        response.Write "        Comensales"                     response.Write " </td>"
        response.Write " <td>"                          response.Write                  comensales
                response.Write " </td>"                 response.Write "</tr>"          end if
        if (ask_dias = 1 or ask_comensales = 1) and monto_limite <> -1 then
        response.Write "<tr class=texto_nivel_2>"               response.Write " <td>"
        response.Write "        Subtotal diario"                response.Write " </td>"
        response.Write " <td>"                          response.Write
        formatcurrency(subtotal_diario,2)               response.Write " </td>"
        response.Write "</tr>"          end if          if monto_limite <> -1 then
        response.Write "<tr class=texto_nivel_2>"               response.Write " <td>"
        response.Write "        Monto limite permitido"                 response.Write " </td>"
                response.Write " <td>"                  response.Write
        formatcurrency(monto_limite,2)                  response.Write " </td>"
        response.Write "</tr>"          end if          response.Write "</table>" end function
        function muestra_distribucion(detalle)          'Esta funcion muestra la distribucion de esos
detalles en los         'diferentes centros de costo            'Recibe:        'detalle = numero del
detalle dentro de la solicitud          dim sql         dim res         dim arRes               dim
t_arRes         dim count               dim gran_subtotal               dim gran_iva            dim
gran_propina            dim gran_tua            dim gran_total          set res = connect.execute(sql)
                if not res.eof then             arRes = res.getrows                             t_arRes =
ubound(arRes,2)         else                    t_arRes = -1            end if          if t_arRes >=
```

```
0 then                  gran_subtotal = 0                  gran_iva = 0
        gran_propina = 0                gran_tua = 0                    gran_total = 0
        response.Write "  <tr>"                 response.Write "          <td class=tabla_nivel_4
colspan=3>Centro de costo</td>"                 response.Write "          <td class=tabla_nivel_4
colspan=6>Porcentaje</td>"                      response.Write "  </tr>"                for count = 0
to t_arRes                      gran_subtotal   = gran_subtotal + arRes(2,count)
                gran_iva        = gran_iva                      + arRes(3,count)
        gran_propina    = gran_propina  + arRes(4,count)                gran_tua
        = gran_tua                      + arRes(5,count)                gran_total              =
gran_total      + arRes(6,count)                response.Write "<tr class=texto_nivel_2>"
                response.Write "  <td colspan=3>"&arRes(0,count)&"</td>"
        response.Write "  <td>"&arRes(1,count)&"%</td>"                   response.Write "
        <td>"&formatcurrency(arRes(2,count),2)&"</td>"                  response.Write "
        <td>"&formatcurrency(arRes(3,count),2)&"</td>"                  response.Write "
        <td>"&formatcurrency(arRes(4,count),2)&"</td>"                  response.Write "
        <td>"&formatcurrency(arRes(5,count),2)&"</td>"                  response.Write "
        <td>"&formatcurrency(arRes(6,count),2)&"</td>"                  response.Write
"</tr>"                 next            end if  end function    function muestra_existentes(folio)
        'Esta funcion muestra los detalles capturados dentro de la solicitud            'Recibe:
        'folio = numero de folio de la solicitud                dim sql         dim res         dim arRes
                dim t_arRes             dim count               dim gran_subtotal               dim
gran_iva        dim gran_propina        dim gran_tua            dim gran_total          dim
asteriscos              dim clase               sql = "exec
[dbo].sp_muestra_datos_capturados_ver_detalle "&folio&",1 "     set res = connect.execute(sql)
        if not res.eof then             t_arRes = ubound(arRes,2)               else
        t_arRes = -1            end if          if t_arRes >= 0 then
        gran_subtotal = 0               gran_iva = 0                    gran_propina = 0
        gran_tua = 0                    gran_total = 0                  response.Write "<table
border=0 width=""100%"">"                       response.Write "  <tr>"
        response.Write "        <td class=tabla_nivel_2 colspan=9>Opciones de desplegado</td>"
                response.Write "  </tr>"                        response.Write "  <tr class=texto_nivel_2>"
                response.Write "        <td valign=top colspan=3>"
        response.Write "                Distribuci&oacute;n a centros de costo"
        response.Write "        </td>"                          response.Write "        <td valign=top
colspan=7>"                     if despliega_distribucion = 1 then
        response.Write "        <a
href=javascript:despliega_distribucion(0,'"&status&"','"&ver_menu&"');>Ocultar</a>"                      else
                                response.Write "        <a
href=javascript:despliega_distribucion(1,'"&status&"','"&ver_menu&"');>Desplegar</a>"                    end
if                      response.Write "        </td>"                  response.Write "  </tr>"
                response.Write "  <tr class=texto_nivel_2>"             response.Write "        <td
valign=top colspan=3>"                  response.Write "                Informacion detallada
(Adicional y Anexos)"                   response.Write "        </td>"
        response.Write "        <td valign=top colspan=7>"                      if
despliega_informacion = 1 then                  response.Write "        <a
href=javascript:despliega_informacion(0,'"&status&"','"&ver_menu&"');>Ocultar</a>"                       else
                                response.Write "        <a
href=javascript:despliega_informacion(1,'"&status&"','"&ver_menu&"');>Desplegar</a>"                     end
if                      response.Write "        </td>"                  response.Write "  </tr>"
                response.Write "  <tr>"                 response.Write "        <td
class=tabla_nivel_2 colspan=9>Detalle</td>"                     response.Write "  </tr>"
        if despliega_distribucion = 0 then                      response.Write "<tr
class=tabla_nivel_3>"                   response.Write "  <td>Detalle</td>"
        response.Write "  <td>Concepto</td>"                    response.Write "
        <td>Comentario</td>"                    response.Write "  <td>Fecha de gasto</td>"
                        response.Write "  <td>Subtotal</td>"
        response.Write "  <td>IVA</td>"                         response.Write "  <td>Propina</td>"
                        response.Write "  <td>TUA</td>"
        response.Write "  <td>Total</td>"                       response.Write "</tr>"
        end if                  for count = 0 to t_arRes                gran_subtotal   =
gran_subtotal + arRes(4,count)                  gran_iva        = gran_iva              +
arRes(5,count)                  gran_propina    = gran_propina  + arRes(6,count)
```

```
                gran_tua                = gran_tua              + arRes(7,count)
        gran_total              = gran_total    + arRes(8,count)                        if
despliega_distribucion = 1 then                                 response.Write "<tr
class=tabla_nivel_3>"                                   response.Write " <td>Detalle</td>"
                        response.Write " <td>Concepto</td>"
        response.Write " <td>Comentario</td>"                                   response.Write "
        <td>Fecha de gasto</td>"                        response.Write " <td>Subtotal</td>"
                        response.Write " <td>IVA</td>"
        response.Write " <td>Propina</td>"                                      response.Write "
        <td>TUA</td>"                                   response.Write " <td>Total</td>"
                        response.Write "</tr>"                          end if
        if arRes(9,count) = 0 then                              asteriscos = ""
                clase = ""                              else
        asteriscos = "**"                               clase = "aviso"                         end
if                              response.Write "<tr class=texto_nivel_2>"
        response.Write " <td class="&clase&">"&arRes(0,count)&" "&asteriscos&"</td>"
                response.Write " <td>"&arRes(1,count)&"</td>"
        response.Write " <td>"&arRes(2,count)&"</td>"                           response.Write "
        <td>"&formato_fecha(arRes(3,count))&"</td>"                             response.Write "
        <td>"&formatcurrency(arRes(4,count),2)&"</td>"                          response.Write "
        <td>"&formatcurrency(arRes(5,count),2)&"</td>"                          response.Write "
        <td>"&formatcurrency(arRes(6,count),2)&"</td>"                          response.Write "
        <td>"&formatcurrency(arRes(7,count),2)&"</td>"                          response.Write "
        <td>"&formatcurrency(arRes(8,count),2)&"</td>"                          response.Write
"</tr>"                         if despliega_distribucion = 1 then
                        call muestra_distribucion(arRes(0,count))
                                'Funcion que muestra la distribucion entre los
centros de costo                                                        'Localizacion
= esta misma pagina                     end if                          if
despliega_informacion = 1 then                          response.Write "<tr>"
                        response.Write " <td colspan=3 valign=top>"
        response.Write                  "<table border=0 width=""100%"">"
        response.Write "                <tr>"                           response.Write "
                        <td colspan=3>"
                                call muestra_informacion(arRes(0,count))

        'Funcion que muestra informacion adicional del detalle
                                                                'Localizacion = esta misma
pagina                                  response.Write "                        </td>"
                        response.Write "                <tr>"
        response.Write "        </table>"                               response.Write " <td
colspan=6 valign=top>"                                  response.Write          "<table
border=0 width=""100%"">"                                       response.Write "
        <tr>"                                   response.Write "                        <td
colspan=3>"
                        call muestra_anexos(arRes(0,count))
                                                                'Funcion que muestra los
anexos del detalle
                        'Localizacion = esta misma pagina
        response.Write "                                </td>"
        response.Write "                <tr>"                           response.Write "
        </table>"                                       response.Write " </td>"
                response.Write "</tr>"                          end if
        response.Write "<tr>"                                   response.Write " <td colspan=9><hr></td>"
                        response.Write "</tr>"                  next                            if
despliega_distribucion = 1 or despliega_informacion = 1 then
        response.Write "<tr class=tabla_nivel_3>"                               response.Write " <td
colspan=4>Totales</td>"                                 response.Write " <td>Subtotal</td>"
                        response.Write " <td>IVA</td>"
        response.Write " <td>Propina</td>"                                      response.Write "
        <td>TUA</td>"                                   response.Write " <td>Total</td>"
                        response.Write "</tr>"                          end if
```

```
        response.Write "<tr class=texto_nivel_2>"                    response.Write "  <td
colspan=4>Totales</td>"                    response.Write "
        <td>"&formatcurrency(gran_subtotal,2)&"</td>"                    response.Write "
        <td>"&formatcurrency(gran_iva,2)&"</td>"                    response.Write "
        <td>"&formatcurrency(gran_propina,2)&"</td>"                    response.Write "
        <td>"&formatcurrency(gran_tua,2)&"</td>"                    response.Write "
        <td>"&formatcurrency(gran_total,2)&"</td>"                    response.Write "</tr>"
                    response.Write "<tr class=texto_nivel_2>"
        response.Write "  <td colspan=9><hr></td>"                    response.Write "</tr>"
                    response.Write "<tr class=aviso>"                    response.Write "  <td
colspan=9>** Este detalle incumple pol&iacute;ticas</td>"                    response.Write
"</tr>"                    response.Write "</table>"            end if    end function    function global()
        'Funcion que presenta la informacion capturada para las solicitudes        response.Write
"<table border=0 width=""90%"" align=center>"                    call muestra_folio_relaciones(folio,"2")
        'Funcion que verifica con que otros folio se relaciona el folio propuesto        'Localizacion =
"../includes/muestr_folio_relac.asp"        response.Write "  <tr>"                    response.Write "            <td
class=tabla_nivel_2 colspan=2>Informaci&oacute;n General</td>"                    response.Write "  </tr>"
        response.Write "  <tr>"                response.Write "            <td valign=top width=""100%"">"
                                                                                    call
muestr_datos_grale(folio,"texto_nivel_2")
        'Funcion que muestra los datos generales de las solicitudes
                                'Localizacion = "../includes/muestr_datos_grale.asp"
        response.Write "            </td>"            response.Write "  </tr>"            response.Write "
        <tr>"                response.Write "            <td colspan=2>"
                                call muestra_existentes(folio)
                                'Funcion que muestra los detalles capturados de la solicitud
                                                'Localizacion = esta misma pagina
        response.Write "            </td>"            response.Write "  </tr>"            response.Write "
        <tr>"                response.Write "            <td colspan=2 valign=top align=center>"
                                                call
opciones_proveniencia(status,folio,proviene)
                'Funcion que muestra diversas opciones dependiendo de la proveniencia
                                                'de la peticion por ver el detalle de la solicitud
                                                        'Localizacion = esta misma
pagina            response.Write "            </td>"            response.Write "  </tr>"
        response.Write "</table>" end function%><html>    <head>            <title>Detalle de
solicitud</title>            <script src="../includes/open_windo.js"></script>                <script
src="../includes/open_anexo.js"></script>            <script src="../includes/valida_politicas.js"></script>
        <link rel="stylesheet" href="../stylesheet/style.css">            <!--                    function
despliega_distribucion(despliega_distribucion,status,ver_menu){                    //Funcion
que recarga la pagina para que se despliege la distribucion                    //a los centros de
costo de los detalles                    //Recibe:
        //despliega_distribucion = variable para desplegar u ocultar la distribucion (1,0;respectivamente)
                        //status = status de la solicitud, solo aplica cuando la peticion por ver el
                //                    detalle proviene de la pagina de AUtorizaciones
        document.form.despliega_distribucion.value = despliega_distribucion;
        document.form.action = 'solic_ver_detal.asp?status='+status+'&ver_menu='+ver_menu+";
                        document.form.submit();                    }                    function
despliega_informacion(despliega_informacion,status,ver_menu){                    //Funcion
que recarga la pagina para que se despliege informacion                    //adicional de los
detalles (informacion no relevante y anexos)                    //Recibe:
        //despliega_informacion = variable para desplegar u ocultar la informacion y anexos
                //(1,0;respectivamente)                    //status = status de la solicitud, solo
aplica cuando la peticion por ver el                //                    detalle proviene de la pagina
de AUtorizaciones                    document.form.despliega_informacion.value =
despliega_informacion;                    document.form.action =
'solic_ver_detal.asp?status='+status+'&ver_menu='+ver_menu+";
        document.form.submit();                    }                    function
regresar_a_pantalla_autoriza(){                    //Funcion a traves de la cual se puede regresar
a la pantalla de AUtorizaciones                    document.form.action =
'../flujo_autoriza/flujo_panta_autor.asp';                    document.form.submit();
        }                    function registra_aut_o_rec(opcion,status){
```

```
	//Esta funcion manda a llamar a la pagina que hace el registro de 	//la autorizacion o el rechazo como tal 	//Solo aplica cuando la peticion por ver el detalle proviene de la 	//pagina de AUtorizaciones
	tipo_operacion = document.form.tipo_operacion.value;
	numempleado = document.form.numempleado.value; 	idempresa = document.form.idempresa.value; 	folio = document.form.folio.value;
		comentario = document.form.comentario.value;
	document.form.action = '../flujo_autoriza/flujo_regis_autor.asp?status='+status+'&tipo_operacion='+tipo_operacion+'&opcion='+opcion
		document.form.submit(); 	} 	-->
	</script></head> <body>  <%if ver_menu<>"0" then%>	<!--#include file=../menu/menu.asp-->
	<%end if%>	<form name="form" method="post">	<%	%>	<input type=hidden name=idempresa 	value=<%=idempresa%>>	<input type=hidden name=numempleado 	value=<%=numempleado%>>
	<input type=hidden name=folio 		value=<%=folio%>>
	<input type=hidden name=tipo_operacion 		value=<%=tipo_operacion%>>
	<input type=hidden name=despliega_distribucion 	value=<%=despliega_distribucion%>>
	<input type=hidden name=despliega_informacion 	value=<%=despliega_informacion%>>
	<input type=hidden name=proviene 		value=<%=proviene%>>
	<input type=hidden name=tipo_autorizacion 	value=IN>		<input type=hidden name=ver_menu 	value=<%=ver_menu%>>	<!--Este campo especifica de donde si la autorizacion es global(GL) o individual(IN)-->	</form>  </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../solicitud/solic_manda_autor.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
	'Fte_ - -	'Fecha de ultima modificacion	: 6 de julio 2007	'Funcionalidad general	: Pagina un paso antes de mandar a autorizar%><!--#include file="../connect/conne_open.asp"--><!--	Este archivo contiene la conexion a la BD--><!--	#include file="../includes/muestr_boton.asp"--><!--	Este archivo incluye la funcion que presenta los botones de las paginas--><%
	'================================================================================	'Estas variables son comunes a todas las paginas	dim numempleado 	'Numero de empleado que esta firmado en la aplicacion	dim idempresa 	'Numero de empresa en la que se esta trabajando	numempleado = request("numempleado")
	idempresa = request("idempresa")
	'================================================================================
	'================================================================================	'Estas variables se usan nada mas en las paginas de solicitudes
	dim folio 	'Numero de folio de la solicitud
		folio = request("folio")
	'================================================================================
	'================================================================================	'Estas variables se usan nada mas en las paginas de solicitudes
	dim folio_principal 	'Numero de folio de la solicitud	folio_principal = request("folio_principal")
	'================================================================================
	'================================================================================	'Estas variables se usa para saber de donde proviene la peticion de esta pagina, se usa en	'la mayoria de las paginas	dim proviene 	proviene = request("proviene")
	'================================================================================	dim flujo	flujo = request("flujo")
	'================================================================================	'Variable que almacena el tipo de operacion (anticipo,pago,compra,etc)
		dim tipo_operacion	tipo_operacion = request("tipo_operacion")
```

```
'================================================================================
==============	function quita_ppto_principal(folio_principal)	dim sql
	sql = "exec [dbo].sp_elimina_ppto_solicitud "&folio_principal&""
	connect.execute(sql)	end function	function envio_mails(folio)
	dim sql	dim res	dim status	dim arRes
		dim t_arRes	dim count	sql = "select
[dbo].fn_obtiene_status_por_folio("&folio&")"	set res = connect.execute(sql)
		if not res.eof then	status = res(0)	else
		status = ""	end if	set res =
connect.execute(sql)	if not res.eof then	arRes = res.getrows
		t_arRes = ubound(arRes,2)	else
		t_arRes = -1	end if	if t_arRes >= 0 then
		response.write "<table width=""90%"" align=center>"
	response.Write "  <tr>"	response.Write "	<td
class=tabla_nivel_2 align=center>Se le ha enviado un correo de aviso a:</td>"
	response.Write "  </tr>"	response.Write "</table>"
	response.write "<table width=""90%"" align=center class=contenedor>"	for
count = 0 to t_arRes	response.Write "<tr>"
		response.Write "  <td class=texto_nivel_2 align=center>"&arRes(4,count)&"</td>"
		response.Write "</tr>"	next
		response.Write "</table>"	end if	end function
	function manda_autorizar(folio,numempleado,idempresa,status,desde_final)
	'Esta funcion manda a autorizar la solicitud	'Recibe:	'folio = folio
de la solicitud	'numempleado = numero de empleado que hace la solicitud
	'idempresa = empresa en la que se trabaja	'status = Status con que se registrara al
movimiento de la solicitud	dim sql	dim res	dim
folio_asignado	dim status_asignado	dim idstatus_asignado
		sql = "exec [dbo].sp_flujo_autoriza_mandar_autorizar "&folio&",
"&numempleado&","&idempresa&", "'&status&'", "&desde_final&" "	'response.write sql
		'response.end	set res = connect.execute(sql)
	if not res.eof then	folio_asignado = res(0)
	status_asignado = res(1)	idstatus_asignado = res(2)
	else	folio_asignado = "Indefinido"
	status_asignado = "Indefinido"	idstatus_asignado = "nu"
	end if	response.write "<table width=""90.5%"" align=center>"
	response.write "  <tr>"	response.write "	<td class=tabla_nivel_2
colspan=2 align=center>"	response.Write "	Solicitud enviada"
		response.write "	</td>"	response.write "  </tr>"
	response.write "</table>"	response.write "<table width=""90%"" align=center
class=contenedor>"	response.write "  <tr>"	response.write "
	<td class=texto_nivel_2 align=center>"	response.Write "	Su
solicitud a quedado registrada con el folio # "	response.Write
	folio_asignado	response.write "	</td>"
	response.write "  </tr>"	response.write "  <tr>"
	response.write "	<td class=texto_nivel_2 align=center>"
	response.Write "	Con el estado : "	response.Write
		status_asignado	response.write "	</td>"
	response.write "  </tr>"	response.write "</table>"
		call envio_mails(folio_asignado)
	response.write "<table width=""90%"" align=center>"	response.write "  <tr
height=15>"	response.write "  </tr>"	response.write "  <tr>"
		response.write "	<td colspan=2 align=center>"
			call muestra_boton("ver detalle","ver_detalle();")
		response.Write "	 "
			call muestra_boton("Ver flujo de
autorizacion","autorizaciones_pendientes(document.form.folio.value,'"&idstatus_asignado&"');")
			'Funcion que muestra los botones
			'Localizacion =
"../includes/muestr_boton.asp"	response.write "	</td>"
	response.write "  </tr>"	response.write "</table>"	end
function%><html>	<head>	<title>Mandar a autorizar</title>	<script
src="../includes/open_windo.js"></script>	<link rel="stylesheet" href="../stylesheet/style.css">
```

```
<!-- function ver_detalle(){ //Funcion a traves de la cual se accede al detalle de la solicitud //La solicitud tiene que estar en proceso de autorizacion para que //se tenga acceso a esta funcionalidad //Recibe: //folio = numero de folio de la solicitud //se asigna a proviene el valor US (USuario). Asi se restringen las operaciones //que el usuario tiene dentro del detalle de la solicitud folio = document.form.folio.value; tipo_operacion = document.form.tipo_operacion.value;
open_window('solic_ver_detal.asp?folio='+folio+'&tipo_operacion='+tipo_operacion+'&proviene=US','Detalle','status=no,resizable=yes,scrollbars=yes'); //Funcion para abrir ventanas //Localizacion = "../inlcudes/open_windo.js" }
function autorizaciones_pendientes(folio,status){
open_window('../reporte_autoriza/repor_autor_autor.asp?folio='+folio+'&status='+status+'','repor_autorizaciones','width=600,height=300,scrollbars=yes,resizable=yes,status=yes'); }
--> </script></head> <body> <!--#include file=../menu/menu.asp--> <form name="form" method="post"> <input type=hidden name=idempresa
value=<%=idempresa%>> <input type=hidden name=numempleado
value=<%=numempleado%>> <input type=hidden name=folio
value=<%=folio%>> <input type=hidden name=tipo_operacion
value=<%=tipo_operacion%>> <input type=hidden name=folio_principal
value=<%=folio_principal%>> <% if proviene = "QP" then
'response.Write "se quita presupuesto<br>" call quita_ppto_principal(folio_principal)
end if if flujo = "dp" then call manda_autorizar(folio,numempleado,idempresa,"nu","0") 'Funcion que manda a autorizar la solicitud 'Localizacion = esta misma pagina end if if flujo = "df" then call manda_autorizar(folio,numempleado,idempresa,"nu","1") end if %> </form> </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../solicitud/solic_detal_captu.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
'Fte_ - - 'Fecha de ultima modificacion : 6 de julio 2007 'Funcionalidad general : Pagina mediante la cual se captura el detalle de las '
solicitudes%><!-- #include file="../connect/conne_open.asp"--><!-- Este archivo contiene la conexion a la BD--><!-- #include file="../includes/muestr_folio.asp"--><!-- Este archivo incluye la funcion que muestra el folio de la solicitud al usuario--><!-- #include file="../includes/muestr_descr.asp"--><!-- Este archivo incluye la funcion que presenta la caja de texto de la descripcion--><!-- #include file="../includes/muestr_fecha.asp"--><!--
Este archivo incluye la funcion que presenta las cajas de texto para que se seleccione una fecha--><!-- #include file="../includes/muestr_texto.asp"--><!-- Este archivo incluye la funcion que presenta un cuadro de texto--><!-- #include file="../includes/muestr_boton.asp"--><!-- Este archivo incluye la funcion que presenta los botones--><!-- #include file="../includes/catal_conce_gasto.asp"--><!-- Este archivo incluye la funcion que muestra el catalogo de conceptos de gasto--><!-- #include file="../includes/catal_ivas.asp"--><!-- Este archivo incluye la funcion que muestra el catalogo de ivas--><!-- #include file="../includes/muestr_tipo_conce.asp"--><!-- Este archivo incluye la funcion que pide al usuario ciertos datos dependiendo del tipo de concepto del concepto--><!-- #include file="../includes/forma_fecha.asp"--><!-- Este archivo incluye la funcion que da formato a las fechas--><!--
#include file="../upload/masive_delete.asp"--><!-- Este archivo incluye la funcion que elimina archivos anexos en caso de que se elimine algun detalle--><!-- #include file="../includes/muestr_datos_grale.asp"--><!-- Este archivo incluye la funcion que presenta los datos generales capturados de la solicitud--><!-- #include file="../anticipo/antic_muestr_datos_parti.asp"--><!--
#include file="../pago/pago_muestr_datos_parti.asp"--><!-- #include file="../reservacion/reservacion_muestr_datos_parti.asp"--><!-- #include file="../compra/compra_muestr_datos_parti.asp"--><%
'================================================================================== 'Estas variables son comunes a todas las paginas dim numempleado 'Numero de empleado que esta firmado en la aplicacion dim idempresa 'Numero de empresa en la que se esta trabajando numempleado = request("numempleado")
idempresa = request("idempresa")
```

```
'================================================================================
===============
'================================================================================
===============	'Estas variables son comunes a todas las paginas de solicitudes
	dim idmoneda 'Moneda de la solicitud
	dim idmoneda_pago 'Moneda en que se pretende
se realice el pago	dim tipo_operacion 'Tipo de
solicitud (anticipo,pago,compra,etc)	dim idterritorio
	'Territorio en que se hizo o se hara el gasto	idmoneda = request("idmoneda")
	idmoneda_pago = request("idmoneda_pago")	tipo_operacion =
request("tipo_operacion")	idterritorio = request("idterritorio")
'================================================================================
===============
'================================================================================
===============	'Estas variables se usan nada mas en las paginas de solicitudes
	dim folio 'Numero de folio de la solicitud
		folio = request("folio")
'================================================================================
===============
'================================================================================
===============	'Estas variables se usan nada mas en las paginas de solicitudes
	dim folio_principal 'Numero de folio de la
solicitud principal a la	folio_principal = request("folio_principal")'que la solicitud va asociada
'================================================================================
===============	dim idproveedor_relacionado
'================================================================================
===============	'Estas variables se usan nada mas en las paginas de solicitudes
	dim tipo_operacion_principal 'Numero de
folio de la solicitud principal a la	tipo_operacion_principal = request("tipo_operacion_principal")'que
la solicitud va asociada
'================================================================================
===============
'================================================================================
===============	'Estas variables se usan nada mas en las paginas del detalle de las
solicitudes	dim detalle
	'Numero de detalle dentro de la solicitud	detalle = request("detalle")
'================================================================================
===============
'================================================================================
===============	dim incumplimiento
	'Variable que determina el cumplimiento o incumplimiento
'de politicas de ese detalle dentro de la solicitud
		incumplimiento = request("incumplimiento")
'================================================================================
===============
'================================================================================
===============	'Esta variable almacena la proveniencia de la peticion de las paginas
	dim proviene	'Variable que indica de donde viene la peticion de esta pagina
'IA = Insertar una nueva solicitud que viene de la pagina anterior
'MA = Modificar los datos de una solicitud
existente que viene de la pagina anterior 'GD =
Guardar un detalle para la solicitud en proceso que viene de esta pagina
	'MD = Modifica un detalle para la solicitud en proceso que viene de esta pagina
'ED = Elimina un detalle de la solicitud en proceso que
viene de esta pagina 'OD = Muestra los datos de un
detalle especifico de la solicitud, viene de esta pagina
	'TC = Busca las caracteristicas del tipo de concepto seleccionado, viene de esta pagina
	proviene =	request("proviene")
'================================================================================
===============
'================================================================================
===============	'Estas variables son las que almacenan datos especificos de los detalles
```

```
en proceso                    dim idconcepto                                                    'Variable que
almacena el concepto de gasto                dim comentario
        'Variable que almacena los comentarios de ese detalle                dim fecha_gasto
                                    'Variable que almacena el monto del detalle          idconcepto =
request("idconcepto")
        '================================================================================
===============
        '================================================================================
===============                    'Estas variables son las que almacenan datos especificos de los detalles
en proceso                    dim etiqueta_fecha                                                'Variable que
almacena el concepto de gasto                select case tipo_operacion                                    case
"anticipo","pago","comprobacion_anticipo","comprobacion_reservacion","comprobacion_gasto"
                                                                    : etiqueta_fecha = "Fecha de gasto"
                                    case "compra"                   : etiqueta_fecha = "Fecha de entrega"
                                    case "reservacion"              : etiqueta_fecha = "Fecha de salida"
                        end select
        '================================================================================
===============          function
valida_datos_cabeceros(folio_principal,idmoneda_relacionado,idterritorio_relacionado,idproveedor_relacionad
o,idempresa,numempleado,tipo_operacion)          dim sql          dim res          dim
idmoneda_principal                dim idterritorio_principal          dim idproveedor_principal          dim
territorios_diferentes                dim proveedores_diferentes                    sql = "exec
[dbo].sp_valida_datos_cabeceros_solicitudes "&folio_principal&" "                    set res = connect.execute(sql)
                        if not res.eof then                    idmoneda_principal = res(0)
            idterritorio_principal = res(1)                              idproveedor_principal = res(2)
            response.Write "<form name=form1 method=post>"                    response.Write "  <input
type=hidden name=proviene>"                    response.Write "  <input type=hidden
name=folio_principal>"                response.Write "  <input type=hidden
name=idtipo_operacion_principal>"                    response.Write "  <input type=hidden name=idempresa
value="&idempresa&">"                response.Write "  <input type=hidden name=numempleado
value="&numempleado&">"                    response.Write "  <input type=hidden
name=tipo_operacion value="&tipo_operacion&">"                    response.Write "</form>"
            response.write "<script language=javascript>"                    response.write "   function
regresar_datos_no_validos(){"                    response.write "
            document.form1.proviene.value = ';"                    response.write "
            document.form1.folio_principal.value = ';"                    response.write "
            document.form1.idtipo_operacion_principal.value = '';"                              response.write "
            document.form1.action = 'solic_grale_captu.asp';"                    response.write "
            document.form1.submit();"                    response.write "   }"
            response.write "</script>"                    response.Write "<link rel=stylesheet
href=../stylesheet/style.css>"                    response.Write "<table align=center width=""90%"">"
                        if trim(idmoneda_principal) <> trim(idmoneda_relacionado) then
            monedas_diferentes = 1                              response.Write "<tr>"
            response.Write "  <td><hr></td>"                              response.Write "</tr>"
                        response.Write "<tr>"                              response.Write "<td class=aviso
align=center>La moneda seleccionada no corresponde con la del folio principal"
            response.Write "</td>"                              response.Write "</tr>"                    end
if                              if trim(idterritorio_principal) <> trim(idterritorio_relacionado) then
                        territorios_diferentes = 1                              response.Write "<tr>"
                        response.Write "<td class=aviso align=center>El territorio seleccionado no corresponde con
el del folio principal"                    response.Write "</td>"
            response.Write "</tr>"                    end if                              if trim(idproveedor_principal)
<> trim(idproveedor_relacionado) then                              proveedores_diferentes = 1
                                    response.Write "<tr>"                              response.Write "<td
class=aviso align=center>El proveedor seleccionado no corresponde con el del folio principal"
                        response.Write "</td>"                              response.Write "</tr>"
            end if                          if monedas_diferentes = 1 or territorios_diferentes = 1 or
proveedores_diferentes then                              response.Write "<tr>"
            response.Write "  <td><hr></td>"                              response.Write "</tr>"
                        response.Write "<tr>"                              response.Write "  <td align=center>"
                                                                                    call
muestra_boton("Regresar","regresar_datos_no_validos();")                              response.write "
```

```
        </td>"                          response.Write "</tr>"
        response.Write "</table>"                       response.end                    else
                response.Write "</table>"               end if          end if  end function
        'Esta es la primera instruccion que se ejecuta y es la que compara los datos cabeceros con los datos
        'de los folios relacionados if folio_principal <> "" and (proviene = "IA" or proviene = "MA") then
        if request("idproveedor") = "" then             idproveedor_relacionado = 0             else
                idproveedor_relacionado = request("idproveedor")        end if          call
valida_datos_cabeceros(folio_principal,request("idmoneda"),request("idterritorio"),idproveedor_relacionado,ide
mpresa,numempleado,tipo_operacion)      end if  function carga_datos_detalle(detalle)
        'Esta funcion carga los datos capturados de un detalle cuando este se selecciona
        'para modificarlo               'Recibe         'detalle = numero del detalle que se selecciono
        dim sql         dim res         sql = "exec [dbo].sp_obtiene_datos_detalle "&detalle&" "
        set res = connect.execute(sql)          if not res.eof then             idconcepto =
res("idconcepto")               comentario = res("comentario")          fecha_gasto =
res("fecha_gasto")              incumplimiento = res("inc_pol")         else
        idconcepto = ""                 comentario = ""                 fecha_gasto = ""
        incumplimiento = 0              end if  end function    function muestra_existentes(folio)
        'Esta funcion muestra los detalles capturados para esa solicitud        'Recibe
        'folio = folio de la solicitud que se esta trabajando           dim sql         dim res         dim
arRes           dim t_arRes             dim count               dim gran_total          set res =
connect.execute(sql)            if not res.eof then             arRes = res.getrows
        t_arRes = ubound(arRes,2)               else                    t_arRes = -1            end
if              if t_arRes >= 0 then                    gran_total = 0
        response.Write "<table border=0 width=""100%"">"                response.Write " <tr>"
                response.Write "        <td class=tabla_nivel_2 colspan=7>Datos capturados</td>"
                response.Write " </tr>"                 response.Write " <tr>"
        response.Write "        <td colspan>"                   response.Write "
        <table border=0 width=""100%"" class=contenedor>"               response.Write "
                <tr class=tabla_nivel_3>"               response.Write "
        <td>Eliminar</td>"                      response.Write "
        <td>Detalle</td>"               response.Write "
        <td>Concepto</td>"                      response.Write "
        <td>Descripcion</td>"                   response.Write "
        <td>Distribuci&oacute;n</td>"                   response.Write "
        <td>Anexos</td>"                        response.Write "
        <td>Total</td>"                 response.Write "                        </tr>"
        for count = 0 to t_arRes                        gran_total = gran_total + arRes(4,count)
                response.Write "                        <tr class=texto_nivel_2>"
        response.Write "                        <td align=center><a
href=javascript:eliminar_detalle("&arRes(0,count)&");><img src=../images/cruz.gif border=0></a></td>"
                response.Write "                        <td><a
href=javascript:mostrar_detalle("&arRes(0,count)&");>"&arRes(0,count)&"</a></td>"
        response.Write "                        <td>"&arRes(1,count)&"</td>"
        response.Write "                        <td>"&arRes(2,count)&"</td>"
        response.Write "                        <td><a
href=javascript:mostrar_distribucion("&arRes(0,count)&");>"&arRes(3,count)&"</a></td>"
        response.Write "                        <td><a
href=javascript:anexar_documento("&arRes(0,count)&");><img src=../images/clip.gif border=0></a></td>"
                response.Write "
        <td>"&formatcurrency(arRes(4,count),2)&"</td>"                          response.Write "
                </tr>"                  next                    response.Write "
        <tr class=texto_nivel_1>"               response.Write "                                <td
colspan=6></td>"                response.Write "
        <td>"&formatcurrency(gran_total,2)&"</td>"              response.Write "
        </tr>"                  response.Write "                <table>"
        response.Write "        </td>"                  response.Write " </tr>"
        response.Write "</table>"       end if  end function    function
operaciones_detalle(folio_trabajar,idempresa,numempleado,idconcepto,comentario,fecha_gasto,subtotal,iva,v
alor_iva,propina,tua,total,folio_deposito,proviene)             'Esta funcion efectua varias operaciones sobre
los detalles de las solicitudes         'Recibe         'folio_a_trabajar = folio del detalle
        'idempresa = empresa en la que se esta trabajando       'numempleado = numero de empleado
que hace la solicitud           'idconcepto = concepto de gasto que se selecciono       'comentario
```

```
= comentario del detalle 'fecha_gasto = fecha en que se hizo o se hara el gasto
'subtotal = subtotal del detalle 'iva = iva del detalle 'valor_iva =
identificador de iva del detalle 'propina = propina del detalle 'tua = tua del detalle
'total = total del detalle 'proviene = accion a efectar '
i = insertar detalle nuevo ' u = actualizar detalle
' d = eliminar detalle dim sql dim folio_detalle
dim res dim idiva dim idiva_split if folio_deposito = "" then
folio_deposito = 0 end if if subtotal <> "" then
subtotal = subtotal else subtotal = 0 end if
if iva <> "" then iva = iva else iva = 0 end
if if valor_iva <> "0/0" and trim(valor_iva) <> "" then idiva_split =
split(valor_iva,"/") idiva = idiva_split(0) else idiva = "0"
end if if propina <> "" then propina = propina else
propina = 0 end if if tua <> "" then tua
= tua else tua = 0 end if if total <> "" then
total = total else total = 0 end if if proviene =
"i" then sql = "set dateformat dmy exec [dbo].sp_detalle_solicitud_inserta_modifica
"&folio_trabajar&","&idempresa&","&idconcepto&",
'"&comentario&'","'&fecha_gasto&'","&subtotal&","&iva&","'&idiva&'","&propina&","&tua&","&total&",1,"&incumpli
miento&",'"&proviene&'"" set res = connect.execute(sql) if not res.eof
then folio_detalle = res(0) end if
'Cada vez que se inserta un detalle nuevo, se hace una distribucion igual al 100% al centro
'de costo al que pertenece el empleado. Es una distribucion por default sql
= "exec [dbo].sp_distribucion_inserta_modifica 0,"&folio_detalle&",0,100,"&folio&","&idempresa&",
"&numempleado&", 'i' " connect.execute(sql) else
folio_detalle = folio_trabajar sql = "set dateformat dmy exec
[dbo].sp_detalle_solicitud_inserta_modifica "&folio_detalle&","&idempresa&","&idconcepto&",
'"&comentario&'","'&fecha_gasto&'","&subtotal&","&iva&","'&idiva&'","&propina&","&tua&","&total&",1,"&incumpli
miento&",'"&proviene&'"" connect.execute(sql) end if
'Dependiendo de las caracteristicas del concepto de gasto que se haya seleccionado 'se
inserta, actualiza o elimina en las tablas relacionadas a estas caracteristicas if
request("ask_boleto_avion") = 1 then sql = "set dateformat dmy exec
[dbo].sp_detalle_boleto_avion_inserta_modifica "&folio_detalle&",'"&left(replace(request("origen"),"'",""),50)&"',
'"&left(replace(request("destino"),"'",""),50)&"','"&left(replace(request("aerolinea"),"'",""),50)&"',
'"&left(replace(request("hora_salida"),"'",""),20)&"', '"&proviene&"' " connect.execute(sql)
end if if request("ask_kilometraje") = 1 then sql = "set
dateformat dmy exec [dbo].sp_detalle_kilometro_inserta_modifica "&folio_detalle&","&request("idkm")&",
"&request("no_viajes")&", '"&proviene&"' " connect.execute(sql) end if
if request("ask_precio_unitario_unidades") = 1 then sql = "set dateformat dmy
exec [dbo].sp_detalle_precio_unitario_unidades_inserta_modifica "&folio_detalle&",
"&request("precio_unitario")&", "&request("unidades")&", '"&proviene&'""
connect.execute(sql) end if if request("ask_dias") = 1 then
sql = "set dateformat dmy exec [dbo].sp_detalle_dias_inserta_modifica
"&folio_detalle&","&request("dias")&", '"&proviene&'"" connect.execute(sql) end
if if request("ask_comensales") = 1 then sql = "set dateformat dmy
exec [dbo].sp_detalle_comensales_inserta_modifica "&folio_detalle&","&request("comensales")&",
'"&proviene&'"" connect.execute(sql) end if if
request("ask_deposito") = 1 then sql = "set dateformat dmy exec
[dbo].sp_detalle_deposito_inserta_modifica "&folio_detalle&","&folio_deposito&", '"&proviene&'""
connect.execute(sql) end if 'Se actualiza el valor detalle = "" para que no
aparezca la opcion de eliminar 'despues que el detalle haya sido guardado, actualizado o
eliminado detalle = "" end function function
operaciones_solicitud(idempresa,numempleado,tipo_operacion,idstatus,idmoneda,idmoneda_pago,idterritorio,
descripcion,proviene) 'Esta funcion efectua varias operaciones sobre las solicitudes
'Recibe 'idempresa = empresa en la que se esta trabajando 'numempleado =
numero de empleado que hace la solicitud 'tipo_operacion = concepto de gasto que se selecciono
'idstatus = comentario del detalle 'idmoneda = fecha en que se hizo o se hara el
gasto 'idmoneda_pago = subtotal del detalle 'idterritorio = iva del detalle
'descripcion = identificador de iva del detalle 'proviene = accion a efectar '
gd = guardar un detalle para la solicitud en proceso que viene de esta pagina
' md = modifica un detalle para la solicitud en proceso que viene de esta
pagina ' ed = elimina un detalle de la solicitud en proceso que viene de
```

```
esta pagina                 '                       od = muestra los datos de un detalle especifico de la
solicitud, viene de esta pagina             dim sql         dim res         dim arRes               dim
t_arRes         dim count               if proviene = "i" then                  sql = "exec
[dbo].sp_solicitud_inserta_modifica
0,"&idempresa&","&numempleado&","'&tipo_operacion&"','&idstatus&","&idmoneda&","&idmoneda_pago&","&i
dterritorio&","'&descripcion&"','i'"                set res = connect.execute(sql)                  if
not res.eof then                        folio = res(0)                  end if                  if
trim(folio_principal) <> "" then                        sql = "insert into
folio_principal_folio_relacionado(folio_principal,folio_relacionado) values ("&folio_principal&","&folio&")"
                connect.execute(sql)                    if
trim(lcase(tipo_operacion_principal)) <> "anticipo"  then                               sql = "exec
[dbo].sp_asociar_principal_a_relacionado "&folio_principal&",0,"&folio&", "'&trim(tipo_operacion_principal)&"'"
                        connect.execute(sql)                    end if
        end if          sql = ""        else            sql = "exec
[dbo].sp_solicitud_inserta_modifica
"&folio&","&idempresa&","&numempleado&","'&tipo_operacion&"','&idstatus&","&idmoneda&","&idmoneda_pa
go&","&idterritorio&","'&descripcion&"','"&proviene&"'"         connect.execute(sql)
        sql = ""        end if          select case lcase(trim(tipo_operacion))
        case "anticipo"         : sql = "set dateformat dmy exec [dbo].sp_anticipo_inserta_modifica
        "&folio&", "'&request("fecha_ini")&"', "'&request("fecha_fin")&"','"&proviene&"'"
        case "pago"             : sql = "set dateformat dmy exec [dbo].sp_pago_inserta_modifica
                "&folio&", "&request("idproveedor")&",
"'&replace(request("no_factura"),"'","")&"','"&proviene&"'"             case "reservacion"      : sql
= "set dateformat dmy exec [dbo].sp_reservacion_inserta_modifica     "&folio&", "&request("idproveedor")&",
"'&proviene&"'"             case "compra"       : sql = "set dateformat dmy exec
[dbo].sp_compra_inserta_modifica        "&folio&", "&request("idproveedor")&",
"'&left(replace(request("depto"),"'",""),50)&"', "'&proviene&"'"        end select      end function
        '=============================================================================
=========================   if proviene = "IA" then         'Inserta Nueva solicitud        call
operaciones_solicitud(idempresa,numempleado,tipo_operacion,"nu",idmoneda,idmoneda_pago,idterritorio,repl
ace(left(request("descripcion"),250),"'"," "),"i")          'Localizacion = esta misma pagina end if    if
proviene = "MA" then            'Modifica Solicitud             call
operaciones_solicitud(idempresa,numempleado,tipo_operacion,"nu",idmoneda,idmoneda_pago,idterritorio,repl
ace(left(request("descripcion"),250),"'"," "),"u")          'Localizacion = esta misma pagina end if
        '=============================================================================
=========================
        '=============================================================================
=========================   if proviene = "GD" then         'Guarda Detalle del anticipo
        call
operaciones_detalle(folio,idempresa,numempleado,request("idconcepto"),replace(left(request("comentario"),2
50),"'","
"),request("fecha_gasto"),request("subtotal"),request("iva"),request("valor_iva"),request("propina"),request("tua
"),request("total"),request("folio_deposito"),"i")           'Localizacion = esta misma pagina end if    if
proviene = "MD" then            'Modifica Detalle de la solicitud               call
operaciones_detalle(request("detalle"),idempresa,0,request("idconcepto"),replace(left(request("comentario"),2
50),"'","
"),request("fecha_gasto"),request("subtotal"),request("iva"),request("valor_iva"),request("propina"),request("tua
"),request("total"),request("folio_deposito"),"u")           'Localizacion = esta misma pagina end if    if
proviene = "ED" then            'Elimina Detalle de la solicitud                call
masive_delete(0,request("detalle"))         'Se eliminan los archivos anexos relacionados a la solicitud
        'Localizacion = "../upload/masive_delete.asp"            call
operaciones_detalle(request("detalle"),idempresa,0,request("idconcepto"),replace(left(request("comentario"),2
50),"'","
"),request("fecha_gasto"),request("subtotal"),request("iva"),request("valor_iva"),request("propina"),request("tua
"),request("total"),request("folio_deposito"),"d")           'Funcion que elimina detalles de la solicitud
        'Localizacion = esta misma pagina    end if
        '=============================================================================
=========================
        '=============================================================================
=========================   if proviene = "OD" then         'mOstrar Detalle de la solicitud
        call carga_datos_detalle(request("detalle"))        'Localizacion = esta misma pagina    end if
        '=============================================================================
```

```
========================= function global() 'Funcion que muestra la informacion general
capturada y diversas opciones al usuario response.Write "<table border=0 width=""90%""
align=center>" response.Write " <tr>" response.Write " <td
class=tabla_nivel_2 colspan=2>Informaci&oacute;n General</td>" response.Write " </tr>"
response.Write " <tr>" response.Write " <td valign=top width=""100%"">"
response.Write " <table border=0 width=""100%"" align=center class=contenedor>"
response.Write " <tr>" response.Write "
<td valign=top width=""100%"">"
call muestr_datos_grale(folio,"texto_nivel_2")
response.Write " </td>" response.Write "
</tr>" response.Write " </table>"
response.Write " </td>" response.Write " </tr>" response.Write "
<tr>" response.Write " <td colspan=2>"
call muestra_existentes(folio) response.Write "
</td>" response.Write " </tr>" response.Write " <tr>"
response.Write " <td>" response.Write " <table border=0
width=""100%"">" response.Write " <tr>" response.Write "
<td class=tabla_nivel_2 colspan=5>Informaci&oacute;n del gasto<a
name=mostrar_detalle> </a></td>" response.Write " </tr>"
response.Write " <tr>" response.Write "
<td colspan=5>" response.Write "
<table border=0 width=""100%"" class=contenedor>" response.Write "
<tr>"
call catalogo_concepto_gasto(idempresa,"Concepto de
gasto","idconcepto",idconcepto,"onchange=javascript:tipo_concepto();",tipo_operacion,"texto_nivel_2","u",requ
est("idproveedor"))
'Funcion que muestra el catalogo de conceptos de gasto

'Localizacion = "../includes/catal_conce_gasto.asp" response.Write "
</tr>" response.Write "
<tr>"
call
muestra_descripcion("Descripcion","comentario",comentario,3,60,3,"texto_nivel_2")
'Funcion que
muestra la caja de texto para la desripcion del detalle
'Localizacion =
"../includes/muestr_descr.asp" response.Write "
</tr>" response.Write " <tr>"

call muestra_fecha(etiqueta_fecha,"fecha_gasto",fecha_gasto,"texto_nivel_2")
'Funcion que
muestra la caja de texto para seleccionar una fecha
'Localizacion = "../includes/muestr_fecha.asp"
response.Write " </tr>"
if proviene
<> "IA" and proviene <> "MA" and proviene <> "" then
call
muestra_tipo_concepto(idempresa,numempleado,idconcepto,idmoneda,idterritorio,tipo_operacion,detalle,"text
o_nivel_2")
'Funcion que muestra las diferentes opciones dependiendo del concepto de gasto
que se elija
'Localizacion = "../includes/muestr_tipo_conce.asp"
end if
response.Write " </table>"
response.Write " </tr>" response.Write "
</td>" response.Write " </tr>" response.Write " <tr>"
response.Write " <td align=center>" if detalle = "" then
call muestra_boton("Guardar","guardar_detalle('GD',0);")
else call
muestra_boton("Guardar","guardar_detalle('MD',document.form.detalle.value);") response.Write "
 " call
muestra_boton("Eliminar","eliminar_detalle(document.form.detalle.value);") end if
```

```
response.Write "                     "
        call muestra_boton("Terminar captura","terminar_captura(document.form.folio.value);")
                                                                    'Funcion que muestra los botones
                                                                    'Localizacion =
"../includes/muestr_boton.asp"          response.Write "            </td>"          response.Write "
    </tr>"          response.Write "</table>" end function%><html>      <head>
    <title>Detalle de solicitud</title>         <script src="../includes/open_windo.js"></script>
    <script src="../includes/suma_total.js"></script>           <script
src="../includes/valida_politicas.js"></script>         <script src="../includes/calen_picke.js"></script>
    <link rel="stylesheet" href="../stylesheet/style.css">      <script language=javascript>
    <!--                    //Funcion que valida los datos capturados antes de guardar un
detalle                 //Recibe:                       //valor = valor de proveniencia
para ver que se hace                //                  GD = Guarda detalle
        //                  MD = Modifica detalle                   //detalle = folio del
detalle de la solicitud                 if(document.form.idconcepto.value==0){
            alert('Favor de seleccionar un concepto de gasto');
    document.form.idconcepto.focus();                       return false;
        }                   if(document.form.comentario.value==''){
            alert('Favor de introducir una breve descripcion');
    document.form.comentario.focus();                       return false;
        }                   if(document.form.comentario.value.length>250){
                alert('La descripcion no puede ser mayor a 250 caracteres');
            document.form.comentario.focus();                       return false;
                }
    if(document.form.fecha_gasto.value==''){                    alert('Favor de
introducir la fecha de gasto');                 document.form.fecha_gasto.focus();
                    return false;                   }
        if(document.form.total.value==0){                       alert('El total no
puede ser cero');               return false;                       }
            //Funcion que valida el incumplimiento de politicas
    //Localizacion = "../includes/valida_politicas.js"
    if(valida_politicas()==1){                  if(detalle==0)
            document.form.detalle.value = '';                       else
                document.form.detalle.value = detalle;
    if(document.form.ask_boleto_avion.value==1){
    if(document.form.origen.value==''){                     alert('Favor de
introducir el origen');                 document.form.origen.focus();
                    return false;                       }
            }               if(document.form.ask_boleto_avion.value==1){
                if(document.form.destino.value==''){
                alert('Favor de introducir el destino');
        document.form.destino.focus();                          return false;
                }               }
    if(document.form.ask_boleto_avion.value==1){
    if(document.form.hora_salida.value==''){                        alert('Favor
de introducir la hora aproximada de salida');
    document.form.hora_salida.focus();                      return false;
                }               }
    if(document.form.ask_kilometraje.value==1){
    if(document.form.kilometraje_costo.value==0){
    alert('No hay un costo por kilometro definido, favor de notificar al administrador');
                return false;                       }
        }                   //Se guarda el detalle de la solicitud
                document.form.proviene.value = valor;
    document.form.action = 'solic_detal_captu.asp';
    document.form.submit();                 }               }
    function eliminar_detalle(detalle){                 //Recibe:
    //detalle = folio del detalle de la solicitud                   //Se asigna un valor de
proviene = ED (Elimina Detalle)                 //y un valor de 0 al incumplimiento de politicas
(como se va a eliminar no tiene caso                    //el valor que tenga)
        document.form.incumplimiento.value = 0;
    document.form.detalle.value = detalle;                  document.form.proviene.value
```

```
= 'ED';                                 document.form.action = 'solic_detal_captu.asp';
        document.form.submit();                 }                               function
mostrar_detalle(detalle){                       //Funcion a traves de la cual se recarga la pagina para
mostrar los datos                       //capturados del detalle                        //Recibe:
                                        //detalle = folio del detalle de la solicitud                   //Se
asigna un valor de proviene = MO (MOstrar detalle)                      document.form.detalle.value =
detalle;                                document.form.proviene.value = 'OD';
        document.form.action = 'solic_detal_captu.asp#mostrar_detalle';
        document.form.submit();                 }                               function
terminar_captura(folio){                        //Funcion a traves de la cual se indica que la captura ha
terminado                               document.form.folio.value = folio;
        document.form.action = 'solic_datos_autor.asp';
        document.form.submit();                 }                               function tipo_concepto(){
                        //Funcion a traves de la cual se recarga la pagina para mostrar las
                //caracteristicas del concepto seleccionado                             //Recibe:
                        //Se asigna un valor de proviene = TC (Tipo de Concepto)
        document.form.detalle.value = '';               document.form.proviene.value = 'TC';
                                document.form.action = 'solic_detal_captu.asp';
        document.form.submit();                 }                               function
mostrar_distribucion(detalle){                          //Funcion a traves de la cual se abre la pagina
que muestra las diferentes distribuciones               //Recibe:
        //detalle = folio del detalle de la solicitud                   folio =
document.form.folio.value;                     idempresa = document.form.idempresa.value;

        open_window('../distribucion/distr_detal.asp?folio='+folio+'&idempresa='+idempresa+'&detalle='+detal
le+'','','status=no,scrollbars=yes,resizable=yes,height=400,width=700');                        //Funcion
para abrir una pagina nueva                             //Localizacion = "..includes/open_windo.js"
                }                               function anexar_documento(detalle){
        //Funcion a traves de la cual se abre la pagina que muestra los archivos anexos
        //Recibe:                               //detalle = folio del detalle de la solicitud
                folio = document.form.folio.value;                      idempresa =
document.form.idempresa.value;
        open_window('../anexo/anexo_detal.asp?folio='+folio+'&idempresa='+idempresa+'&detalle='+detalle+'
','','status=yes,scrollbars=yes,resizable=yes,height=400,width=700');                          //Funcion
para abrir una pagina nueva                             //Localizacion = "..includes/open_windo.js"
                }               -->             </script></head> <body>  <!--#include
file=../menu/menu.asp-->  <form name="form" method="post">      <%              'Funcion que muestra
los datos capturados de la solicitud            'Localizacion = esta misma pagina %>            <input
type=hidden name=idempresa              value=<%=idempresa%>>           <input type=hidden
name=numempleado                value=<%=numempleado%>>         <input type=hidden
name=idmoneda           value=<%=idmoneda%>>            <input type=hidden name=idterritorio
        value=<%=idterritorio%>>        <input type=hidden name=tipo_operacion
        value=<%=tipo_operacion%>>              <input type=hidden name=proviene
        value=<%=request("proviene")%>>         <input type=hidden name=incumplimiento
        value=<%=incumplimiento%>>              <input type=hidden name=folio
        value=<%=folio%>>               <input type=hidden name=detalle
        value=<%=detalle%>>             <input type=hidden name=folio_principal
        value=<%=folio_principal%>>             <input type=hidden name=idproveedor
        value=<%=request("idproveedor")%>>      </form>  </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../solicitud/solic_datos_autor.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -               'Fecha de ultima modificacion       : 6 de julio 2007           'Funcionalidad
general                         : Pagina un paso antes de mandar a autorizar                    '
                                                        Se presentan las opciones de: regresar, ver detalle,
        '                                                               dejar pendiente y mandar a
autorizar
        '================================================================================
===============                 'Estas variables son comunes a todas las paginas            dim
```

```
numempleado                                                    'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                                  'Numero de
empresa en la que se esta trabajando            numempleado = request("numempleado")
        idempresa = request("idempresa")
        '================================================================================
==============
        '================================================================================
==============        'Estas variables son comunes a todas las paginas de solicitudes
        dim idmoneda                                                   'Moneda de la solicitud
        dim idmoneda_pago                                              'Moneda en que se pretende
se realice el pago      dim tipo_operacion                                             'Tipo de
solicitud (anticipo,pago,compra,etc)        dim idterritorio
        'Territorio en que se hizo o se hara el gasto       idmoneda = request("idmoneda")
        idmoneda_pago = request("idmoneda_pago")            tipo_operacion =
request("tipo_operacion")       idterritorio = request("idterritorio")
        '================================================================================
==============
        '================================================================================
==============        'Estas variables se usan nada mas en las paginas de solicitudes
        dim folio                                                      'Numero de folio de la solicitud
                folio = request("folio")
        '================================================================================
==============
        '================================================================================
==============        'Estas variables se usan nada mas en las paginas de solicitudes
        dim folio_principal                                                'Numero de folio de la
solicitud       folio_principal = request("folio_principal")
        '================================================================================
==============
        '================================================================================
==============        'Estas variables se usa para saber de donde proviene la peticion de esta
pagina, se usa en       'la mayoria de las paginas      dim proviene            proviene =
request("proviene")
        '================================================================================
==============
        '================================================================================
==============        'Estas variables se usa para saber si todos los detalles estan distribuidos
al 100% en      'centros de costo       'Se inicializa la variable en cero      dim
distribucion_valida             distribucion_valida = 0
        '================================================================================
==============%><!--#include file="../connect/conne_open.asp"--><!--     Este archivo contiene la
conexion a la BD--><!--     #include file="../includes/muestr_folio.asp"--><!--     Este archivo incluye la funcion
que muestra el folio de la solicitud al usuario--><!--   #include file="../includes/muestr_datos_grale.asp"--><!--
        Archivo que incluye la funcion que muestra los datos generales de las solicitudes--><!-- #include
file="../includes/muestr_boton.asp"--><!--     Este archivo incluye la funcion que presenta los botones--><!--
        #include file="../includes/forma_fecha.asp"--><!--     Este archivo incluye la funcion que da formato a
las fechas--><!--  #include file="../anticipo/antic_muestr_datos_parti.asp"--><!-- #include
file="../pago/pago_muestr_datos_parti.asp"--><!--     #include
file="../reservacion/reservacion_muestr_datos_parti.asp"--><!--          #include
file="../compra/compra_muestr_datos_parti.asp"--><%         function revisa_flujo(folio)            dim sql
        dim res         dim t_arRes             sql = "exec [dbo].sp_reporte_consulta_autorizadores
"&folio&",'nu','p',0,1"             set res = connect.execute(sql)              if not res.eof then
        arRes = res.getrows                     t_arRes = ubound(arRes,2)               else
                t_arRes = -1            end if          if t_arRes >= 0 then
        revisa_flujo = true             else                    revisa_flujo = false                end
if      end function    function tablas_iguales(folio_principal,folio_relacionado)              dim sql
        dim res         dim iguales             dim flujo       sql = "exec [dbo].sp_compara_tablas
"&folio_principal&", "&folio_relacionado&""          set res = connect.execute(sql)          if not res.eof
then                    iguales = res(0)                flujo = res(1)          else
        iguales = 0                     flujo = "dp"            end if          tablas_iguales =
iguales & "/" & flujo       end function        function
revisa_saldo_en_contra(folio_principal,folio,tipo_operacion)            dim sql        dim res        dim
```

```
saldo                sql = "select
[dbo].fn_revisa_saldos_en_contra(''&folio_principal&'',''&folio&'',''&lcase(trim(tipo_operacion))&'')"            set
res = connect.execute(sql)            if not res.eof then                saldo = res(0)                else
                saldo = 0                end if            if saldo = 0 then
        revisa_saldo_en_contra = "MI/0"            end if            if saldo < 0 then
        revisa_saldo_en_contra = "SC/"&saldo                end if            if saldo > 0 then
        revisa_saldo_en_contra = "SF/"&saldo                end if    end function        function
revisa_distribucion_por_folio(folio)            'Esta funcion revisa que todos los detalles este distribuidos
        'al 100% entre los centros de costo            dim sql            dim res            sql = "exec
[dbo].sp_revisa_distribucion_por_folio ''&folio&'' "            set res = connect.execute(sql)            if
not res.eof then            distribucion_valida = res(0)                else
        distribucion_valida = 0            end if    end function        call revisa_distribucion_por_folio(folio)
        'Funcion que comprueba que la solicitud este distribuida en un 100% en los     'centros de costo
        'Localizacion = esta misma pagina function global()            dim proviene_sig_pagina            dim
flujo            dim split_tablas_iguales            dim split_tipo_saldo_saldo            dim tipo_saldo
        dim saldo                'Funcion que presenta diversas opciones al usuario antes de mandar a
autorizar            'Si la solicitud ni esta distribuida al 100%, no se puede mandar a autorizar                'Si
la solicitud esta distribuida al 100%, si se puede mandar a autorizar            response.Write "<table
border=0 width=""90.5%"" align=center>"            response.Write " <tr>"            response.Write "
        <td class=tabla_nivel_2 colspan=2>Informaci&oacute;n General</td>"            response.Write "
        </tr>"            response.Write "</table>"            response.Write "<table border=0 width=""90%""
class=contenedor>"            response.Write " <tr>"            response.Write "            <td
valign=top width=""100%"">"                                                                                call
muestr_datos_grale(folio,"texto_nivel_2")
        'Funcion que muestra los datos capturados de la solicitud
                        'Localizacion = "../includes/muestr_datos_grale.asp"
        response.Write "            </td>"            response.Write " </tr>"            response.Write
"</table>"            if distribucion_valida <> 1 and tipo_operacion <> "comprobacion_reservacion" then
                        response.Write "<table border=0 width=""90%"">"
        response.Write " <tr>"            response.Write "            <td colspan=4><hr></td>"
                response.Write " </tr>"                response.Write " <tr>"
        response.Write "            <td colspan=2 align=center class=aviso>"
        response.Write                "En este momento no puede mandar a autorizar su
solicitud porque faltan detalles por distribuir entre los centros de costo o no est&aacute;n distribuidos al 100%"
                        response.Write "            </td>"                response.Write " </tr>"
                response.Write "</table>"            end if            proviene_sig_pagina = ""            flujo
= "dp"            if trim(folio_principal) <> "" then                response.Write "<table border=0
width=""90%"">"                split_tablas_iguales = split(tablas_iguales(folio_principal,folio),"/")
                flujo = split_tablas_iguales(1)                if trim(lcase(tipo_operacion)) =
"comprobacion_anticipo" or trim(lcase(tipo_operacion)) = "comprobacion_reservacion" then
                split_tipo_saldo_saldo =
split(revisa_saldo_en_contra(folio_principal,folio,trim(tipo_operacion)),"/")
        saldo = abs(split_tipo_saldo_saldo(1))                                select case
split_tipo_saldo_saldo(0)                                case "MI"        :        tipo_saldo = "Saldo "
                                response.Write "<tr>"
                                        response.Write " <td colspan=2
align=center class=aviso>"
        response.Write "            El monto de su solicitud corresponde al de la solicitud asociada"
                                        response.Write " </td>"
                                case "SF"        :        tipo_saldo = "Saldo a Favor "
                                response.Write "<tr>"
                        response.Write " <td colspan=2 align=center class=aviso>"
                                        response.Write "            Tiene registrado un
saldo a favor de "&formatcurrency(saldo,2)&""
                        response.Write " </td>"
        response.Write "</tr>"                                case "SC"        :        tipo_saldo =
"Saldo en Contra "
        response.Write "<tr>"
        response.Write " <td colspan=2 align=center class=aviso>"
                        response.Write "            Tiene registrado un saldo en contra de
"&formatcurrency(saldo,2)&".<br> "
```

```
response.Write "  Debe hacer un deposito a favor de la empresa "
response.Write "  Y usar dicho concepto de gasto para la comprobacion "
response.Write " </td>"
response.Write "</tr>"
distribucion_valida = 0 end select else proviene_sig_pagina = "QP" if (split_tablas_iguales(0)) <> 1 then response.Write "<tr>"
response.Write " <td colspan=2 align=center class=aviso>"
response.Write "  Los conceptos o montos o distribuciones de este folio con el folio al que esta "
response.Write "  asociado no corresponden. Si decide mandar a autorizar, el flujo de su solicitud "
response.Write "  comenzara desde el principio"
response.Write " </td>"
response.Write "</tr>" else flujo = "df"
response.Write "<tr>"
response.Write " <td colspan=2 align=center class=aviso>"
response.Write "  La informacion de los dos folio corresponden.<br>"
response.Write "  Su solicitud solo pasara por el ultimo nivel de autorizacion."
response.Write " </td>"
response.Write "</tr>" end if end if
response.Write "</table>"  end if  response.Write "<table border=0 width=""90%"">"  response.Write " <tr height=30>"  response.Write "  <td colspan=2 align=center>"  call muestra_boton("Dejar pendiente","dejar_pendiente();")  response.Write "   "  call muestra_boton("Ver resumen de captura","ver_detalle(document.form.folio.value);")
response.Write "   "
call muestra_boton("Regresar al detalle","regresar_detalle(document.form.folio.value);")
response.Write "  </td>"  response.Write " </tr>"  if distribucion_valida = 1 then  response.Write "<tr>"  response.Write " <td colspan=2 align=center>"  if revisa_flujo(folio) then
call muestra_boton("Mandar a autorizar","mandar_autorizar('"&proviene_sig_pagina&"','"&flujo&"');")
response.Write " "  end if
call muestra_boton("Ver flujo de autorizacion","autorizaciones_pendientes(document.form.folio.value,'nu');")
'Funcion que presenta los botones
'Localizacion = "../includes/muestr_boton.asp"
response.Write " </td>"  response.Write "</tr>"  end if
response.Write "</table>" end function%><html>  <head>  <title>Captura de Solicitud</title>  <script src="../includes/open_windo.js"></script>  <script src="../includes/suma_total.js"></script>  <script src="../includes/valida_politicas.js"></script>
<link rel="stylesheet" href="../stylesheet/style.css">  <!--  function ver_detalle(){  //Funcion que permite ver el detalle de la solicitud
folio = document.form.folio.value;  tipo_operacion = document.form.tipo_operacion.value;  proviene = 'US';
//Variable que indica que la peticion de ver el detalle proviene del USuario
//esto es para que no muestre opciones a las que el USuario no tiene acceso
open_window('solic_ver_detal.asp?folio='+folio+'&tipo_operacion='+tipo_operacion+'&proviene='+proviene+'&ver_menu=0','Detalle','status=yes,resizable=yes,scrollbars=yes');  }
function mandar_autorizar(proviene,flujo){
document.form.proviene.value = proviene;  document.form.flujo.value = flujo;  document.form.action = 'solic_manda_autor.asp';
document.form.submit();  }  function autorizaciones_pendientes(folio,status){
open_window('../reporte_autoriza/repor_autor_autor.asp?folio='+folio+'&status='+status+'&ver_menu=0','repor_autorizaciones','width=600,height=300,scrollbars=yes,resizable=yes,status=no');
}  function dejar_pendiente(){  //Funcion que deja pendiente la solicitud, para mandarla a autorizar despues
document.form.folio.value = '';  document.form.proviene.value = '';
document.form.action = 'solic_grale_captu.asp';
document.form.submit();  }  function regresar_detalle(){
```

```
                                        //Funcion que permite al usuario regresar al detalle de la solicitud para
hacer alguna                                    //correccion
        document.form.proviene.value = '';                              document.form.action =
'solic_detal_captu.asp';                                document.form.submit();                         }
        -->             </script></head> <body>  <!--#include file=../menu/menu.asp-->       <form
name="form" method="post">              <input type=hidden name=idempresa
        value=<%=idempresa%>>           <input type=hidden name=numempleado
        value=<%=numempleado%>>                 <input type=hidden name=folio
        value=<%=folio%>>               <input type=hidden name=idmoneda
        value=<%=idmoneda%>>            <input type=hidden name=tipo_operacion
        value=<%=tipo_operacion%>>              <input type=hidden name=proviene
        value=<%=proviene%>>            <input type=hidden name=folio_principal
        value=<%=folio_principal%>>             <input type=hidden name=flujo
        value=""> <%            call global()           'Funcion que presenta diversas
opciones al usuario antes de mandar a autorizar             'Localizacion = esta misma pagina %>
        </form>  </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../anticipo/antic_pide_datos_parti.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion        : 5 de julio 2007   'Funcionalidad general
        : Esta funcion permite al usuario capturar los datos particulares       '
                                        de las solicitudes de anticipo          'Recibe  'Fecha_ini =
valor de fecha de inicio en caso de que sea un anticipo ya capturado   'Fecha_fin = valor de fecha de fin en
caso de que sea un anticipo ya capturado   'Clase = clase de estilo con que se presentara el texto
        'Funciones relacionadas   'muestra_fecha = funcion que muestra el cuadro de texto y el calendario
para poder seleccionar una fecha        function antic_pide_datos_parti(fecha_ini,fecha_fin,clase)
        response.Write " <tr>"                                                                  call
muestra_fecha("Fecha de inicio","fecha_ini",fecha_ini,clase)        response.Write " </tr>"
                response.Write " <tr>"
        call muestra_fecha("Fecha de fin","fecha_fin",fecha_fin,clase)      response.Write " </tr>"
                                        response.Write "</table>" end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../anticipo/antic_carga_datos_parti.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion        : 11 de julio 2007 'Funcionalidad general
        : Esta funcion carga los datos particulares de los      '
                        anticipos cuando son seleccionados      'Recibe: 'folio = numero de folio de la
solicitud function carga_datos_anticipo(folio)          dim sql         sql = "exec
[dbo].sp_anticipo_muestra_datos_parti "&folio&", 1"         set res = connect.execute(sql)          if
not res.eof then            fecha_ini = res(0)          fecha_fin = res(1)      else
                fecha_ini = ""          fecha_fin = ""          end if     end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../anticipo/catal_antic_auto.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -          'Fecha de ultima modificacion        : 6 de julio 2007          'Funcionalidad
general                 : Pagina que presenta las solicitudes de anticipo que       '
                                        pueden ser asociadas a una comprobacion de
anticipo%><!--  #include file="../connect/conne_open.asp"--><!--     Este archivo contiene la conexion a la
BD--><!--       #include file="../includes/forma_fecha.asp"--><!--     Este archivo incluye la funcion que da
formatoa las fechas--><!-- #include file="../includes/muestr_boton.asp"--><!--   Este archivo incluye la funcion
que presenta los botones--><%
        '=============================================================================
==============          'Estas variables son comunes a todas las paginas        dim
numempleado                                             'Numero de empleado que esta firmado
en la aplicacion        dim idempresa                                           'Numero de
```

```
empresa en la que se esta trabajando                    numempleado = request("numempleado")
        idempresa = request("idempresa")
        '================================================================================
==============                  function catalogo_anticipos_pagados(idempresa,numempleado)
                'Esta funcion presenta las solicitudes de anticipo que ya han sido pagadas
        'y que pueden ser asociadas a una comprobacion de anticipo                  'idempresa =
empresa a la que pertenece la solicitud                 'numempleado = numero de empleado que hace
la solicitud                    dim sql                 dim res                 dim arRes
                dim count               dim gran_total                  dim seleccionar
                dim valor               sql = "exec [dbo].sp_anticipos_pagados_catalogo
"&idempresa&","&numempleado&" "                 set res = connect.execute(sql)
        if not res.eof then             arRes = res.getrows
        t_arRes = ubound(arRes,2)                       else                            t_arRes = -1
                        end if          response.Write "<table border=0 width=""90%"">"
                response.Write " <tr>"                  response.Write "        <td
class=tabla_nivel_2 colspan=9>Anticipos pagados</td>"                   response.Write " </tr>"
                if t_arRes >= 0 then                    gran_total = 0
        response.Write " <tr class=tabla_nivel_3>"                      response.Write "
        <td>Folio</td>"                 response.Write "        <td>Descripcion</td>"
                        response.Write "        <td>Monto</td>"
        response.Write "        <td>Moneda</td>"                                response.Write "
        <td>Moneda del pago</td>"                       response.Write "
        <td>Territorio</td>"                    response.Write "        <td>Fecha de
solicitud</td>"                 response.Write "        <td>Asociar</td>"
                response.Write "        <td>Ver detalle</td>"
        response.Write " </tr>"                 for count = 0 to t_arRes
                response.Write "<tr class=texto_nivel_2>"
        response.Write " <td>"&arRes(0,count)&"</td>"
        response.Write " <td>"&arRes(1,count)&"</td>"
        response.Write " <td>"&formatcurrency(arRes(2,count),2)&"</td>"
        response.Write " <td>"&arRes(3,count)&"</td>"
        response.Write " <td>"&arRes(5,count)&"</td>"
        response.Write " <td>"&arRes(7,count)&"</td>"
        response.Write " <td>"&formato_fecha(arRes(9,count))&"</td>"
        response.Write " <td><a
href=javascript:asocia_compra_pago("&arRes(0,count)&","&arRes(4,count)&","&arRes(6,count)&","&arRes(8,c
ount)&");>Asociar</a></td>"                     response.Write " <td><a
href=../solicitud/solic_ver_detal.asp?folio="&arRes(0,count)&"&proviene=US&ver_menu=0>ver
detalle</a></td>"               response.Write "</tr>"                          next
                        else                    response.Write " <tr class=aviso colspan=9
align=center>"                  response.Write "        <td>Por el momento, no hay anticipos
disponibles para asociar</td>"                  response.Write " </tr>"                 end
if                      response.Write "<tr class=texto_nivel_2>"               response.Write " <td
colspan=9><hr></td>"                    response.Write "</tr>"                  response.Write "<tr>"
                        response.Write " <td colspan=9 align=center>"
                        call muestra_boton("Cerrar","window.close();")
        response.Write "         "
        call muestra_boton("Imprimir","window.print();")                        response.Write " </td>"
                response.Write "</tr>"                  response.Write "</table>"       end
function%><html>        <head>          <title>Anticipos pagados</title>                <script
src="../includes/open_window.js"></script>              <script src="../includes/no_right_mouse.js"></script>
        <link rel="stylesheet" href="../stylesheet/style.css">          <script language=javascript>
        <!--                    function
asocia_compra_pago(folio_principal,idmoneda,idmoneda_pago,idterritorio,idproveedor){
        //Esta funcion ingresa los datos correspondientes a la pagina que la abrio
(../solicitud/solic_grale_captu.asp)                    //Recibe:
        //folio_principal = folio del anticipo al que pertence esta comprobacion
        //idmoneda = moneda en la que se pidio el anticipo                      //idmoneda_pago =
moneda en la que se pago el anticipo                    //idterritorio = territorio en el que se
gasto el anticipo                       //tipo_operacion_principal = tipo de operacion de anticipo
                        //Si los datos introducidos estan correctos,                    //se copian
estos datos a la pagina que abrio este catalogo
```

```
        //(../solicitud/solic_grale_captu.asp)
        window.opener.document.form.folio_principal.value = folio_principal;
        window.opener.document.form.idmoneda.value = idmoneda;
        window.opener.document.form.idterritorio.value = idterritorio;
        window.opener.document.form.tipo_operacion_principal.value = 'anticipo';                                }
                -->                </script></head> <body>  <form name="form" method="post">
        <input type=hidden name=idempresa          value=<%=idempresa%>>          <input type=hidden
name=numempleado       value=<%=numempleado%>>       <%                call
catalogo_anticipos_pagados(idempresa,numempleado)                'Esta funcion obtiene los anticipos
pagados que pueden ser asociados a una comprobacion              'Localizacion = esta misma pagina  %>
        </form>  </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../anticipo/antic_muestr_datos_parti.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion       : 5 de julio 2007  'Funcionalidad general
        : Esta funcion muestra al usuario los datos particulares de un '
                                anticipo ya capturado     'Recibe: 'Folio = Folio de la solicitud capturada
        'Clase = clase de estilo con que se presentara el texto          'Funciones relacionadas
        'Formato_fecha = funcion que da formato a la fecha que reciba          function
antic_muestr_datos_parti(folio,clase)                dim sql             dim res             dim reg_fecha_ini
                dim reg_fecha_fin                    sql = "exec [dbo].sp_anticipo_muestra_datos_parti
"&folio&", 0"                set res = connect.execute(sql)                if not res.eof then
        reg_fecha_ini = res(0)                       reg_fecha_fin = res(1)                else
        reg_fecha_ini = ""                      reg_fecha_fin = ""                    end if
        response.Write "<table border=0 width=""100%"">"             response.Write "  <tr class="&clase&">"
                response.Write "             <td>Fecha de inicio</td>"             response.Write "
        <td>"&formato_fecha(reg_fecha_ini)&"</td>"                   response.Write "  </tr>"
        response.Write "  <tr class="&clase&">"                  response.Write "             <td>Fecha de
fin</td>"             response.Write "             <td>"&formato_fecha(reg_fecha_fin)&"</td>"
        response.Write "  </tr>"             response.Write "</table>" end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../inicio/inicio_inici.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declarar variables%><%
        'Fte_ - -            'Fecha de ultima modificacion       : 6 de julio 2007            'Funcionalidad
general                      : Pagina de bienvenida
        '=============================================================================
===============              'Estas variables son comunes a todas las paginas              dim
numempleado                                                  'Numero de empleado que esta firmado
en la aplicacion             dim idempresa                                                 'Numero de
empresa en la que se esta trabajando                 numempleado = request("numempleado")
        idempresa = request("idempresa")
        '=============================================================================
===============%><!--#include file="../connect/conne_open.asp"--><!--      Este archivo contiene la
conexion a la BD--><html>          <head>          <title>Bienvenida</title>          <link rel="stylesheet"
media=screen href="../stylesheet/style.css">          <link rel="stylesheet" media=print
href="../stylesheet/print.css">                <script src="../includes/no_right_mouse.js"></script>
        <script language=javascript>                 <!--                -->                </script></head>
        <body>  <!--#include file=../menu/menu.asp-->       <p> </p>  <table align=center
width=89% border=0>              <tr>                        <td width=60%>
        <table align=center width=100% border=0>                                    <tr>
                                <td class=tabla_nivel_2>
        Bienvenida                                          </td>
        </tr>                           </table>            </td>                        <td>
                        <table align=center width=100% border=0>                                    <tr>
                                <td class=tabla_nivel_2>
                        Soporte                                          </td>
                </tr>                   </table>            </td>          </tr>
```

```
        <tr>                    <td valign=top>                 <table align=center
width=99% class=contenedor>                                     <tr>
                <td class=texto_nivel_2>                                        Bienvenido
al sistema de solicitud de: <br>                                        <ul>
                                        <li>Anticipos</li>
                        <li>Comprobacion de Anticipos</li>
                <li>Compras</li>
        <li>Comprobacion de Gastos y</li>
        <li>Pagos a Proveedores</li>                                            </ul>
                                        </td>                                   </tr>
                </table>                </td>                   <td valign=top>
        <table align=center width=99% class=contenedor>                         <tr>
                                        <td class=texto_nivel_2>
                Para soporte tecnico, no dude en llamar a la extension 3124
                </td>                                   </tr>                   </table>
                </td>           </tr>   </table> </body></html>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../comprobacion_anticipo/comprobacion_anticipo_pide_datos_parti.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion        : 11 de julio 2007 'Funcionalidad general
        : Esta funcion permite al usuario capturar los datos particulares       '
                                        de las solicitudes de pago      'Recibe  'idempresa =
empresa en la que se esta trabajando        'numempleado = empleado que solicita la comprobacion de
anticipo  'Clase = clase de estilo con que se presentara el texto       'Funciones relacionadas
        'catalogo_proveedor = funcion que muestra el catalgo de proveedores 'muestra_texto = funcion que
muestra un cuadro de texto          function
comprobacion_anticipo_pide_datos_parti(idempresa,numempleado,clase)             response.Write
"<table border=0 width=""100%"">"       if folio = "" then                      response.Write " <tr>"
                response.Write "        <td class="&clase&" colspan=2>"
        response.Write "                <a
href=javascript:open_window('../anticipo/catal_antic_auto.asp?idempresa="&idempresa&"&numempleado="&n
umempleado&"','','width=900,height=300,scrollbars=yes,status=yes,resizable=yes');>ver anticipos
autorizados</a>"                response.Write "        </td>"                  response.Write "
        </tr>"          end if          response.Write "</table>" end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../comprobacion_anticipo/comprobacion_anticipo_carga_datos_parti.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion        : 11 de julio 2007 'Funcionalidad general
        : Esta funcion carga los datos particulares de las comprobaciones de anticipos          'Recibe:
        'folio = numero de folio de la solicitud            function
carga_datos_comprobacion_anticipo(folio)                                end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../comprobacion_anticipo/comprobacion_anticipo_muestr_datos_parti.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion        : 11 de julio 2007 'Funcionalidad general
        : Esta funcion muestra al usuario los datos particulares de un '
                                        una comprobacion de anticipos ya capturada      'Recibe  'Folio = Folio
de la solicitud capturada   'Clase = clase de estilo con que se presentara el texto     'Funciones
relacionadas        'Formato_fecha = funcion que da formato a la fecha que reciba       function
comprobacion_anticipo_muestr_datos_parti(folio,clase)       end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/funci_exist_eleme.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-
```

```
<%	Function existe_elemento(idelemento,tabla,where)		dim sql		dim res
		sql = "exec [dbo].sp_elemento_ha_sido_utilizado '"&trim(idelemento)&"','"&trim(tabla)&"',
'"&trim(where)&"' "		set res = connect.execute(sql)		if not reseof then
	existe_elemento = res(0)		else		existe_elemento = true		end
if		End Function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_conce_gasto.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%	'Fte_ - -  'Fecha de ultima modificacion	: 5 de julio 2007	'Funcionalidad general
	: Esta funcion muestra el catalgo de conceptos de gasto de la aplicacion		'Recibe
	'idempresa = empresa en la que se esta trabajando	'mensaje = Leyenda que antecede al cuadro de
texto	'nombre_objeto = nombre del objeto		'valor = Valor a presentar en caso de que haya un
registro capturado		'funcion_javascript = funcion javascript que se ejecutara cuando se cambie el
concepto de gasto		'tipo_operacion = tipo de operacion que se esta realizando (anticipo, pago, compra,
etc)	'clase = clase de estilo con que se presentara el texto		function
catalogo_concepto_gasto(idempresa,mensaje,nombre_objeto,valor,funcion_javascript,tipo_operacion,clase,pr
oviene,proveedor)		dim sql		dim res
		dim arRes		dim t_arRes		dim
count		dim seleccionar		sql = "exec
[dbo].sp_concepto_gasto_catalogo "&idempresa&",'"&tipo_operacion&"','"&proviene&"','"&proveedor&""""
		set res = connect.execute(sql)		if not res.eof then		t_arRes =
ubound(arRes,2)		else		t_arRes = -1		end if
	if t_arRes >= 0 then		response.Write "<td class="&clase&">"
	response.Write " <span>"&mensaje&"</span>"
	response.Write "</td>"		response.Write "<td colspan=3>"
	response.Write " <select class=caja_texto name="&nombre_objeto&" "&funcion_javascript&">"
		response.Write "		<option value=0>[SELECCIONE UN CONCEPTO DE
GASTO]</option>"		for count = 0 to t_arRes
	if trim(arRes(0,count)) = trim(valor) then		seleccionar =
"selected"		else		seleccionar = ""
		end if		response.Write " <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
	next		response.Write " </select>"
	response.Write "</td>"		else		response.Write "<td class="&clase&">"
		response.Write " <span>"&mensaje&"</span>"
	response.Write "</td>"		response.Write "<td class=aviso>"
	response.Write " No hay conceptos de gasto registrados"		response.Write "
	<input type=hidden name="&nombre_objeto&" value="""">"
	response.Write "</td>"		end if	end
function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/muestr_descr.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%		'Fte_ - -  'Fecha de ultima modificacion	: 5 de julio 2007	'Funcionalidad general
		: Esta funcion muestra al usuario el cuadro de texto  '
			para incluir una descripcion		'Recibe  'mensaje = Leyenda
que antecede al cuadro de texto	'nombre_objeto = nombre del objeto		'valor = Valor a presentar en
caso de que haya un registro capturado	'rows = numero de filas de la caja de texto	'cols = numero de
columnas de la caja de texto		'colspan = numero de columnas en las que se extiende dentro de una tabla
	'clase = clase de estilo con que se presentara el texto		function
muestra_descripcion(mensaje,nombre_objeto,valor,rows,cols,colspan,clase)		response.Write "<td
class="&clase&">"		response.Write " <span>"&mensaje&"</span>"
	response.Write "</td>"		response.Write "<td colspan="&colspan&">"
	response.Write " <textarea rows="&rows&" cols="&cols&" class=caja_texto
name="&nombre_objeto&">"&valor&"</textarea>"		response.Write "</td>"		end
function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-
```

../includes/funci_exist_folde.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<% Function ExisteFolder(fldr) Dim fso Set fso =
CreateObject("Scripting.FileSystemObject") If (fso.FolderExists(fldr)) Then
ExisteFolder = true Else ExisteFolder = false End
If End Function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/menu_objet_exist.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<% function obtiene_menu_especifico(idempresa,numempleado,idobjeto,proviene) dim sql
dim res sql = "exec [dbo].sp_existe_perfil_objeto
"&idempresa&","&numempleado&","&idobjeto&","&proviene&"" set res = connect.execute(sql)
obtiene_menu_especifico = trim(res(0)) end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_terri.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<% 'Fte_ - - 'Fecha de ultima modificacion : 5 de julio 2007 'Funcionalidad general
: Esta funcion muestra el catalogo de territorios de la aplicacion 'Recibe
'mensaje = Leyenda que antecede al cuadro de texto'nombre_objeto = nombre del objeto
'valor = Valor a presentar en caso de que haya un registro capturado 'clase = clase de estilo con
que se presentara el texto function catalogo_territorio(mensaje,nombre_objeto,valor,clase)
dim sql dim res dim arRes
dim t_arRes dim count
dim seleccionar sql = "exec [dbo].sp_territorio_catalogo
" set res = connect.execute(sql) if not res.eof then t_arRes =
ubound(arRes,2) else t_arRes = -1 end if
if t_arRes >= 0 then response.Write "<td class="&clase&">"
response.Write " "&mensaje&""
response.Write "</td>" response.Write "<td>"
response.Write " <select class=caja_texto name="&nombre_objeto&">"
response.Write " <option value=0>[SELECCIONE UN TERRITORIO]</option>"
for count = 0 to t_arRes if arRes(0,count) = valor then
seleccionar = "selected" else
seleccionar = "" end if
response.Write " <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
next response.Write " </select>"
response.Write "</td>" else response.Write "<input
type=hidden name="&nombre_objeto&" value="""">" end if end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/muestr_folio.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<% 'Fte_ - - 'Fecha de ultima modificacion : 5 de julio 2007 'Funcionalidad general
: Esta funcion muestra al usuario el folio de la solicitud 'Recibe 'mensaje =
texto que se presentara ante el cuadro de texto 'folio = folio de la solicitud capturada 'Clase =
clase de estilo con que se presentara el texto function
muestra_folio(mensaje,valor_folio,clase) response.Write "<td class="&clase&">"
response.Write " "&mensaje&"" response.Write "</td>"
if valor_folio <> "" then response.Write "<td
class=texto_nivel_2>" response.Write valor_folio
response.Write "</td>" else response.Write "<td
class=texto_nivel_2>" response.Write " Indefinido"
response.Write "</td>" end if end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_tipo_usuar.asp

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

```
<%       'Fte_ - -  'Fecha de ultima modificacion     : 27 de julio 2007 'Funcionalidad general
         : Esta funcion muestra el catalgo de monedas de la aplicacion     'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto   'nombre_objeto = nombre del objeto     'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
         function catalogo_tipo_usuario(mensaje,nombre_objeto,valor,clase)     dim sql
               dim res                     dim arRes
         dim t_arRes                  dim count                    dim
seleccionar                        sql = "exec [dbo].sp_tipo_usuario_catalogo "
     set res = connect.execute(sql)          if not res.eof then          t_arRes =
ubound(arRes,2)        else          t_arRes = -1          end if
     if t_arRes >= 0 then          response.Write "<td class="&clase&">"
     response.Write "  <span>"&mensaje&"</span>"
     response.Write "</td>"          response.Write "<td>"
     response.Write "  <select class=caja_texto name="&nombre_objeto&">"
     response.Write "        <option value=0>[SELECCIONE UN TIPO DE USUARIO]</option>"
               for count = 0 to t_arRes               if arRes(0,count) =
valor then                    seleccionar = "selected"                    else
                    seleccionar = ""               end if
          response.Write "  <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
     next          response.Write "  </select>"
     response.Write "</td>"          else          response.Write "<input type=hidden
name="&nombre_objeto&" value="""">"          end if   end function%>
```

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/muestr_folio_relac.asp

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

```
<%       function muestra_folio_relaciones(folio,colspan)          dim sql       dim res
     dim arRes          dim t_arRes          dim count                    sql
= "exec [dbo].sp_obtiene_folios_relacionados "&folio&",'fr'"          set res = connect.execute(sql)
     if not res.eof then          t_arRes = ubound(arRes,2)          else
     t_arRes = -1          end if          if t_arRes >=0 then
     response.Write "<tr>"          response.Write "  <td class=tabla_nivel_2
colspan="&colspan&">Esta solicitud se relaciona con:</td>"          response.Write "</tr>"
          response.Write "<tr>"          response.Write "  <td colspan=2>"
     response.Write "          <table border=0 width=""100%"">"          response.Write "
          <tr class=texto_nivel_1>"          response.Write "
     <td>Folio</td>"          response.Write "               <td>Tipo de
operacion</td>"          response.Write "          <td>Ver detalle</td>"
          response.Write "          </tr>"          for count = 0 to t_arRes
               if ucase(trim(arRes(2,count))) = "P" then
     folio_principal = arRes(0,count)               end if
     response.Write "          <tr class=texto_nivel_2>"          response.Write "
          <td>"&arRes(0,count)&"</td>"          response.Write "
     <td>"&arRes(1,count)&"</td>"          response.Write "
     <td>"          response.Write "               <a
href=javascript:open_window('../solicitud/solic_ver_detal.asp?folio="&arRes(0,count)&"&proviene=US&ver_me
nu=0','Detalle','status=no,scrollbars=yes,resizable=yes');>Ver detalle</a>"
     response.Write                    "</td>"               response.Write "
     </tr>"                    next
          response.Write "          </table>"          response.Write "  </td>"
          response.Write "</tr>"          else          folio_principal = ""
               end if          end function     %>
```

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_cuent_conta.asp

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

```
<%		'Fte_ - -  'Fecha de ultima modificacion		: 21 de julio 2007 'Funcionalidad general
		: Esta funcion muestra el catalgo de cuentas contables disponibles por empresa
	'Recibe  'empresa = Empresa en la que se esta trabajando	'mensaje = Leyenda que antecede al
cuadro de texto	'nombre_objeto = nombre del objeto		'valor = Valor a presentar en caso de que haya
un registro capturado	'clase = clase de estilo con que se presentara el texto		function
catalogo_cuenta_contable(idempresa,mensaje,nombre_objeto,valor,clase)		dim sql
		dim res		dim arRes
	dim t_arRes		dim count		dim
seleccionar		sql = "exec [dbo].sp_cuenta_contable_catalogo
"&idempresa&", 'u' "		set res = connect.execute(sql)		if not res.eof then
	arRes = res.getrows		t_arRes = ubound(arRes,2)		else
		t_arRes = -1		end if		if t_arRes >= 0 then
		response.Write "<td class="&clase&">"		response.Write "
	<span>"&mensaje&"</span>"		response.Write "</td>"
		response.Write "<td>"		response.Write " <select class=caja_texto
name="&nombre_objeto&">"		response.Write "	<option
value=0>[SELECCIONE UNA CUENTA CONTABLE]</option>"		for
count = 0 to t_arRes		if arRes(count) = valor then
		seleccionar = "selected"		else
	seleccionar = ""		end if		response.Write "
	<option "&seleccionar&" value="&arRes(0,count)&">"&trim(arRes(0,count))&" --
"&arRes(1,count)&"</option>"		next
	response.Write " </select>"		response.Write "</td>"		else
		response.Write "<input type=hidden name="&nombre_objeto&" value="""">"
	end if	end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/pide_datos_grale.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%	'Fte_ - -  'Fecha de ultima modificacion		: 5 de julio 2007	'Funcionalidad general
	: Esta funcion permite al usuario capturar los datos generales '
		de todas las solicitudes		'Recibe  'Folio = Folio de la solicitud
capturada		'idmoneda = idmoneda de la solicitud		'idmoneda_pago = idmoneda en que se
pretende que se haga el pago		'idterritorio = territorio en donde se hara o se hizo el gasto,
dependiendo del tipo de solicitud		'descripcion = descripcion general del gasto
	'Funciones relacionadas	'muestra_folio = funcion que muestra el numero de folio de la solicitud
	'catalogo_moneda = funcion que muestra el catalogo de monedas de la aplicacion
	'catalogo_territorio = funcion que muestra el catalogo de territorios de la aplicacion
	'muestra_descripcion = funcion que da opcion al usuario a ingresar una descripcion general del gasto
	function
pide_datos_grale(folio,idmoneda,idmoneda_pago,idterritorio,descripcion,tipo_operacion,clase)
	response.Write "<table border=0 align=center width=""100%"">"		response.Write "
	<tr>"		call
muestra_folio("Folio",folio,clase)		'Localizacion
= "../includes/muestr_folio.asp"		response.Write " </tr>"
	response.Write " <tr>"		call
catalogo_moneda("Moneda de la solicitud","idmoneda",idmoneda,clase,tipo_operacion)
		'Localizacion = "../includes/catal_moned.asp"
	response.Write " </tr>"		response.Write " <tr>"
		call catalogo_moneda("Moneda del
pago","idmoneda_pago",idmoneda_pago,clase,tipo_operacion)
		'Localizacion = "../includes/catal_moned.asp"		response.Write "
	</tr>"		response.Write " <tr>"
		call catalogo_territorio("Territorio","idterritorio",idterritorio,clase)
			'Localizacion = "../includes/catal_terri.asp"
	response.Write " </tr>"		response.Write " <tr>"
		call muestra_descripcion("Descripcion general del
gasto","descripcion",descripcion,5,35,3,clase)
	'Localizacion = "../includes/muestr_descr.asp"		response.Write " </tr>"
		response.Write "</table>" end function	%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-
```

../includes/catal_area_respo.asp

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

```
<%		'Fte_ - -  'Fecha de ultima modificacion	: 21 de julio 2007 'Funcionalidad general
		: Esta funcion muestra el catalgo de areas de responsabilidad disponibles por empresa
	'Recibe	'empresa = Empresa en la que se esta trabajando	'mensaje = Leyenda que antecede al
cuadro de texto	'nombre_objeto = nombre del objeto		'valor = Valor a presentar en caso de que haya
un registro capturado	'clase = clase de estilo con que se presentara el texto			function
catalogo_area_responsabilidad(idempresa,mensaje,nombre_objeto,valor,clase,liga)		dim sql
				dim res					dim arRes
		dim t_arRes				dim count					dim
seleccionar				dim liga_abre				sql = "exec
[dbo].sp_area_responsabilidad_catalogo "&idempresa&" "		set res = connect.execute(sql)
	if not res.eof then			arRes = res.getrows				t_arRes =
ubound(arRes,2)		else			t_arRes = -1			end if
	if t_arRes >= 0 then				response.Write "<td class="&clase&">"
	if liga = 1  then				liga_abre = "<a
href=javascript:catalogo_area_autoriza("&idempresa&");>(ver)</a>"				end if
	response.Write "  <span>"&mensaje&" "&liga_abre&"</span>"
	response.Write "</td>"				response.Write "<td>"
	response.Write "  <select class=caja_texto name="&nombre_objeto&">"
	response.Write "		<option value=0>[SELECCIONE UN AREA DE
RESPONSABILIDAD]</option>"				for count = 0 to t_arRes
		if arRes(count) = valor then						seleccionar =
"selected"				else						seleccionar = ""
			end if				response.Write "  <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
	next				response.Write "  </select>"
	response.Write "</td>"			else				response.Write "<td class="&clase&">"
			response.Write "  <span>"&mensaje&"</span>"
	response.Write "</td>"				response.Write "<td class=aviso>"
	response.Write "  No hay areas de responsabilidad registradas"
	response.Write "  <input type=hidden name="&nombre_objeto&" value="""">"
		response.Write "</td>"				end if	end function%>
```

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_prove.asp

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

```
<%		'Fte_ - -  'Fecha de ultima modificacion	: 11 de julio 2007 'Funcionalidad general
		: Esta funcion muestra el catalogo de proveedores de la empresa en cuestion
	'Recibe	'empresa = empresa en la que se esta trabajando	'mensaje = Leyenda que antecede al
cuadro de texto	'nombre_objeto = nombre del objeto		'valor = Valor a presentar en caso de que haya
un registro capturado	'clase = clase de estilo con que se presentara el texto			function
catalogo_proveedor(idempresa,mensaje,nombre_objeto,valor,clase)			dim sql
		dim res				dim arRes					dim
t_arRes				dim count					dim seleccionar
			sql = "exec [dbo].sp_proveedor_catalogo "&idempresa&" "			set res =
connect.execute(sql)		if not res.eof then			arRes = res.getrows
	t_arRes = ubound(arRes)		else			t_arRes = -1			end if
	if t_arRes >= 0 then				response.Write "<td class="&clase&">"
	response.Write "  <span>"&mensaje&"</span>"
	response.Write "</td>"				response.Write "<td>"
	response.Write "  <select class=caja_texto name="&nombre_objeto&">"
	response.Write "		<option value=0>[SELECCIONE UN PROVEEDOR]</option>"
				for count = 0 to t_arRes					if trim(arRes(count)) =
trim(valor) then				seleccionar = "selected"					else
					seleccionar = ""					end if
		response.Write "  <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
	next				response.Write "  </select>"
```

```
        response.Write "</td>"                  else                            response.Write "<td class="&clase&">"
                        response.Write "  <span>"&mensaje&"</span>"
        response.Write "</td>"                  response.Write "<td class=aviso>"
        response.Write "  No hay proveedores registrados"                       response.Write "  <input
type=hidden name="&nombre_objeto&" value="""">"                                 response.Write
"</td>"                 end if     end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_objet.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion        : 27 de julio 2007 'Funcionalidad general
                : Esta funcion muestra el catalgo de monedas de la aplicacion           'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto           'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
                function catalogo_objeto(mensaje,nombre_objeto,valor,clase)           dim sql
                        dim res                                 dim arRes
        dim t_arRes                             dim count                                       dim
seleccionar                                     sql = "exec [dbo].sp_objeto_catalogo"                   set
res = connect.execute(sql)          if not res.eof then                 arRes = res.getrows
        t_arRes = ubound(arRes)             else                        t_arRes = -1                end if
                        if t_arRes >= 0 then                            response.Write "<td class="&clase&">"
                        response.Write "  <span>"&mensaje&"</span>"
        response.Write "</td>"                  response.Write "<td>"
        response.Write "  <select class=caja_texto name="&nombre_objeto&">"
        response.Write "            <option value=0>[SELECCIONE UN OBJETO]</option>"
                        for count = 0 to t_arRes                        if arRes(0,count) = valor then
                                seleccionar = "selected"                                    else
                                seleccionar = ""                        end if
        response.Write "  <option "&seleccionar&" value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
                                                        next                    response.Write "
        </select>"                                      response.Write "</td>"          else
        response.Write "<input type=hidden name="&nombre_objeto&" value="""">"          end if     end
function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_statu.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion        : 5 de julio 2007   'Funcionalidad general
                : Esta funcion muestra el catalgo de centros de costo de la aplicacion          'Recibe
        'idempresa = empresa en la que se esta trabajando  'mensaje = Leyenda que antecede al cuadro de
texto      'nombre_objeto = nombre del objeto           'valor = Valor a presentar en caso de que haya un
registro capturado          'clase = clase de estilo con que se presentara el texto          function
catalogo_status(mensaje,nombre_objeto,proviene,valor,clase)             dim sql
                dim res                                 dim arRes                                       dim
t_arRes                         dim count                               dim seleccionar
                'Si la peticion viene del Reporte de Autorizaciones             if proviene = "ra" then
                dim selecciona_n                        dim selecciona_r                        dim
selecciona_e                    dim selecciona_a                                        select case
trim(lcase(valor))                      case "n" : selecciona_n = "selected"
        case "r"  : selecciona_r = "selected"                   case "e" : selecciona_e = "selected"
                        case "a" : selecciona_a = "selected"                            end select
                                                        response.Write "<td class="&clase&">"
                response.Write "  <span>"&mensaje&"</span>"
        response.Write "</td>"                  response.Write "<td>"
        response.Write "  <select class=caja_texto name="&nombre_objeto&">"
        response.Write "            <option                                 value=0>[SELECCIONE UN
ESTADO]</option>"                                       response.Write "            <option
"&selecciona_n&" value=n>Solicitudes nuevas</option>"
        response.Write "            <option "&selecciona_e&" value=e>En espera de autorizacion</option>"
                                                        response.Write "            <option
```

```
"&selecciona_r&" value=r>Rechazadas</option>"
        response.Write "        <option "&selecciona_a&" value=a>Autorizadas</option>"
                response.Write "  </select>"
        response.Write "</td>"                                  end if                                  if
proviene = "us" then                    dim selecciona_ac                       dim selecciona_ba
                                                                select case trim(lcase(valor))
                        case "a" : selecciona_ac = "selected"
        case "b" : selecciona_ba = "selected"                   end select
                        response.Write "<td class="&clase&">"
        response.Write "  <span>"&mensaje&"</span>"
        response.Write "</td>"                  response.Write "<td>"
        response.Write "  <select class=caja_texto name="&nombre_objeto&">"
        response.Write "        <option                                 value=0>[SELECCIONE UN
ESTADO]</option>"                                       response.Write "        <option
"&selecciona_ac&" value=a>Activo</option>"
        response.Write "        <option "&selecciona_ba&" value=b>Baja</option>"
                response.Write "  </select>"
        response.Write "</td>"                                  end if                          if proviene =
"ap" then                       sql = "exec [dbo].so_status_catalogo 'a' "
                                set res = connect.execute(sql)                          if
not res.eof then                        arRes = res.getrows                             t_arRes =
ubound(arRes,2)                 else                    t_arRes = -1                    end
if                                      if t_arRes >= 0 then
        response.Write "<td class="&clase&">"                   response.Write "
        <span>"&mensaje&"</span>"                               response.Write
"</td>"                         response.Write "<td>"                   response.Write "
        <select class=caja_texto name="&nombre_objeto&"
onchange=javascript:selecciona_status(this.value);>"                    response.Write "
        <option value=0>[SELECCIONE UN STATUS]</option>"
        for count = 0 to t_arRes                                if trim(arRes(0,count)) = trim(valor) then
                                        seleccionar = "selected"
        else                                    seleccionar = ""
        end if                          response.Write "  <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
                next                    response.Write "  </select>"
                response.Write "</td>"          else
        response.Write "<td class="&clase&">"                   response.Write "
        <span>"&mensaje&"</span>"                               response.Write
"</td>"                         response.Write "<td class=aviso>"
        response.Write "  No hay monedas registradas"                   response.Write "
        <input type=hidden name="&nombre_objeto&" value="""">"
        response.Write "</td>"                  end if                                  end if
                end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/forma_fecha.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion         : 5 de julio 2007   'Funcionalidad general
        : Esta funcion da formato a las fechas                  'Recibe  'fecha = fecha a la que se le va a dar
formato. las fechas se reciben en formato dd/mm/aaaa            function formato_fecha(fecha)                   dim
dia                     dim mes         dim anio        dim split_fecha                 if trim(fecha) <> ""
then                            split_fecha = split(fecha)                                                      dia
= split_fecha(0)                                                select case cint(split_fecha(1))
                        case "1" : mes = "enero"                        case "2" : mes = "febrero"
                        case "3" : mes = "marzo"                        case "4" : mes = "abril"
                        case "5" : mes = "mayo"                         case "6" : mes = "junio"
                        case "7" : mes = "julio"                        case "8" : mes = "agosto"
                        case "9" : mes = "septiembre"                           case "10"       : mes =
"octubre"                                       case "11"       : mes = "noviembre"
        case "12"       : mes = "diciembre"                     end select
                        anio = split_fecha(2)
```

```
        formato_fecha = dia & "/" & mes & "/" & anio        else                        formato_fecha =
"Indefinida"                end if                        end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/muestr_boton.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion      : 5 de julio 2007   'Funcionalidad general
                : Esta funcion presenta los botones que se usaran dentro de las paginas
        'Recibe  'mensaje = Leyenda que aparece dentro del boton    'funcion_javascript = funcion javsacript
que se ejecutara al hacer clic en el boton              function muestra_boton(mensaje,funcion_javascript)
        response.Write "<input type=button class=boton value=""&mensaje&"""
onclick=javascript:"&funcion_javascript&">"  end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/muestr_fecha.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion      : 5 de julio 2007   'Funcionalidad general
                : Esta funcion muestra al usuario el cuadro de texto y el calendario      '
                                                        para seleccionar una fecha                  'Recibe
        'mensaje = Leyenda que antecede al cuadro de texto'nombre_objeto = nombre del objeto
        'valor = Valor a presentar en caso de que haya un registro capturado   'clase = clase de estilo con
que se presentara el texto                  'Funciones relacionadas    'NewCal = funcion javascript que
presenta el calendario       function muestra_fecha(mensaje,nombre_objeto,valor,clase)
        response.Write "<td class="&clase&">"               response.Write "  <span>"&mensaje&"<span>"
                                response.Write "</td>"              response.Write "<td colspan=3>"
        response.Write "  <input readonly class=caja_texto type=text id="&nombre_objeto&"
name="&nombre_objeto&" value="&valor&">"                    response.Write "  <a
href=javascript:New('"&nombre_objeto&"','ddmmmyyyy',false,24);><img src=../images/calen.gif border=0
alt=""Seleccione una fecha""></a>"
        'Localizacion = "../includes/calen_picke.js"        response.Write "</td>"      end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/muestr_texto.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion      : 11 de julio 2007 'Funcionalidad general
                : Esta funcion muestra al usuario un cuadro de texto                'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto        'valor = Valor a
presentar en caso de que haya un registro capturado'colspan = numero de columnas en las que se extiende
dentro de una tabla        'clase = clase de estilo con que se presentara el texto                'size =
tamano de la caja de texto'maxlength = maximo de caracteres permitidos              function
muestra_texto(mensaje,nombre_objeto,valor,clase,size,maxlength)         response.Write "<td>"
        response.Write "  <span class="&clase&">"&mensaje&"</span>"
        response.Write "</td>"          response.Write "<td>"          response.Write "  <input
type=text class=caja_texto name="&nombre_objeto&" size="&size&" maxlength="&maxlength"
value="""&valor&""">"          response.Write "</td>"          end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_usuar.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion      : 27 de julio 2007 'Funcionalidad general
                : Esta funcion muestra el catalgo de monedas de la aplicacion       'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto        'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
                function
catalogo_usuario(idempresa,mensaje,nombre_objeto,valor,clase,javascript,proviene)          dim sql
                                dim res                                 dim arRes
                dim t_arRes                             dim count                                   dim
seleccionar                                     sql = "exec [dbo].sp_usuario_catalog
```

```
"&idempresa&","'&proviene&"' "                set res = connect.execute(sql)                if not res.eof then
                arRes = res.getrows                        t_arRes = ubound(arRes,2)                else
                        t_arRes = -1                end if                        if t_arRes >= 0 then
                        response.Write "<td class="&clase&">"                        response.Write "
        <span>"&mensaje&"</span>"                                        response.Write "</td>"
                        response.Write "<td>"                response.Write " <select class=caja_texto
name="&nombre_objeto&" "&javascript&">"                response.Write "        <option
value=0>[SELECCIONE UN EMPLEADO]</option>"                                        for count = 0 to
t_arRes                        if arRes(0,count) = valor then
        seleccionar = "selected"                        else
        seleccionar = ""                        end if                        response.Write "
        <option "&seleccionar&" value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
                                        next                response.Write " </select>"
                        response.Write "</td>"                else                        response.Write
"<input type=hidden name="&nombre_objeto&" value="""">"        end if    end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_opera_usuar.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%        'Fte_ - -  'Fecha de ultima modificacion        : 5 de julio 2007  'Funcionalidad general
        : Esta funcion muestra el catalgo de centros de costo de la aplicacion        'Recibe
        'idempresa = empresa en la que se esta trabajando  'mensaje = Leyenda que antecede al cuadro de
texto    'nombre_objeto = nombre del objeto        'valor = Valor a presentar en caso de que haya un
registro capturado        'clase = clase de estilo con que se presentara el texto        function
catalogo_operaciones_usuario(idempresa,numempleado,mensaje,nombre_objeto,valor,clase)        dim
sql                        dim res                        dim arRes
                dim t_arRes                        dim count
        dim seleccionar                        sql = "exec
[dbo].sp_obtiene_tipos_operacion_por_empleado "&idempresa&", "&numempleado&""        set res =
connect.execute(sql)        if not res.eof then                t_arRes = ubound(arRes,2)
        else                t_arRes = -1                end if                        if t_arRes >=
0 then                response.Write "<td class="&clase&">"                        response.Write "
        <span>"&mensaje&"</span>"                                        response.Write "</td>"
                response.Write "<td>"                response.Write " <select class=caja_texto
name="&nombre_objeto&" onchange=javascript:selecciona_tipo_operacion(this.value);>"
        response.Write "        <option value=0>[SELECCIONE UN TIPO DE OPERACION]</option>"
                        for count = 0 to t_arRes                        if
lcase(trim(arRes(0,count))) = lcase(trim(valor)) then                        seleccionar =
"selected"                        else                        seleccionar = ""
                end if                        response.Write " <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
        next                response.Write " </select>"
        response.Write "</td>"                else                response.Write "<input type=hidden
name="&nombre_objeto&" value="""">"        end if    end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_banco.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%        'Fte_ - -  'Fecha de ultima modificacion        : 27 de julio 2007 'Funcionalidad general
        : Esta funcion muestra el catalgo de monedas de la aplicacion        'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto        'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
                function catalogo_banco(mensaje,nombre_objeto,valor,clase)        dim sql
                dim res                        dim arRes
        dim t_arRes                        dim count                        dim
seleccionar                        sql = "exec [dbo].sp_banco_catalogo 'u' "        set
res = connect.execute(sql)        if not res.eof then                arRes = res.getrows
        t_arRes = ubound(arRes,2)                else                t_arRes = -1                end
if                        if t_arRes >= 0 then                response.Write "<td
class="&clase&">"                        response.Write " <span>"&mensaje&"</span>"
```

```
                    response.Write "</td>"                    response.Write "<td>"
        response.Write "  <select class=caja_texto name="&nombre_objeto&">"
        response.Write "            <option value=0>[SELECCIONE UN BANCO]</option>"
                    for count = 0 to t_arRes                    if arRes(count) = valor then
                                seleccionar = "selected"                    else
                                seleccionar = ""                    end if
        response.Write "  <option "&seleccionar&" value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
                                                    next                    response.Write "
        </select>"                    response.Write "</td>"                    else
        response.Write "<input type=hidden name="&nombre_objeto&" value="""">"                    end if     end
function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_perfi.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%          'Fte_ - -  'Fecha de ultima modificacion        : 27 de julio 2007 'Funcionalidad general
            : Esta funcion muestra el catalgo de monedas de la aplicacion          'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto          'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
            function catalogo_perfil(idempresa,mensaje,nombre_objeto,valor,clase)                    dim
sql                    dim res                    dim arRes
            dim t_arRes                    dim count
        dim seleccionar                    sql = "exec [dbo].sp_perfil_catalogo
"&idempresa&" "        set res = connect.execute(sql)          if not res.eof then
        arRes = res.getrows                    t_arRes = ubound(arRes,2)                    else
            t_arRes = -1          end if                    if t_arRes >= 0 then
            response.Write "<td class="&clase&">"                    response.Write "
        <span>"&mensaje&"</span>"                    response.Write "</td>"
            response.Write "<td>"                    response.Write "  <select class=caja_texto
name="&nombre_objeto&">"                    response.Write "            <option
value=0>[SELECCIONE UN PERFIL]</option>"                    for count = 0 to
t_arRes                    if arRes(0,count) = valor then
        seleccionar = "selected"                    else
        seleccionar = ""                    end if                    response.Write "
        <option "&seleccionar&" value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
                    next                    response.Write "  </select>"
            response.Write "</td>"          else                    response.Write
"<input type=hidden name="&nombre_objeto&" value="""">"          end if     end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_tipos_opera.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%          'Fte_ - -  'Fecha de ultima modificacion        : 21 de julio 2007 'Funcionalidad general
            : Esta funcion muestra el catalgo de tipos de operacion disponibles por empresa
        'Recibe  'empresa = Empresa en la que se esta trabajando     'mensaje = Leyenda que antecede al
cuadro de texto    'nombre_objeto = nombre del objeto          'valor = Valor a presentar en caso de que haya
un registro capturado        'clase = clase de estilo con que se presentara el texto                    function
catalogo_tipo_operacion(idempresa,mensaje,nombre_objeto,valor,clase,idconcepto,proviene,dependencia)
            dim sql                    dim res                    dim
arRes                    dim t_arRes                    dim count
            dim seleccionar                    sql = "exec
[dbo].sp_obtiene_tipos_de_operacion "&idempresa&", "'&proviene&"' "          set res = connect.execute(sql)
            if not res.eof then                    arRes = res.getrows                    t_arRes =
ubound(arRes,2)          else                    t_arRes = 1                    end if
        if t_arRes >= 0 then                    response.Write "<td class="&clase&">"
        response.Write "  <span>"&mensaje&"</span>"
        response.Write "</td>"                    response.Write "  <td>"                    for count = 0
to t_arRes                    if isnumeric(idconcepto) then
        sql = "select [dbo].fn_obtiene_concepto_tipo_operacion
("&idempresa&","&idconcepto&","&arRes(0,count)&")"                    else
```

```
                    sql = "select [dbo].fn_obtiene_alta_tipo_operacion
("&idempresa&","&arRes(0,count)&")"                    end if                    set
res = connect.execute(sql)                    if not res.eof then                    if
res(0) = 1 then                    seleccionar = "checked"
          else                    seleccionar = ""
          end if                    else                    seleccionar
= ""                    end if                    response.Write "  <input
"&seleccionar&" type=checkbox name=tipo_operacionMJ"&arRes(1,count)&" value="&arRes(0,count)&"
onclick=javascript:selecciona_tipo_operacion("'&trim(lcase(arRes(1,count)))&"','"&dependencia&"');><span
class="&clase&">" & arRes(2,count) & "</span><br>"                    next
     response.Write "  </td>"          else          response.Write "<input type=hidden
name="&nombre_objeto&" value="""">"          end if     end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/muestr_datos_grale.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%     'Fte_ - -  'Fecha de ultima modificacion     : 5 de julio 2007   'Funcionalidad general
     : Esta funcion muestra al usuario los datos generales capturados     '
                    de las solicitudes     '
                    Tambien presenta los datos particulares de cada tipo de solicitud
     'Recibe  'folio = Leyenda que antecede al cuadro de texto     'clase = clase de estilo con que se
presentara el texto                    'Funciones relacionadas   'antic_muestr_datos_parti = funcion
que muestra datos particulares de los anticipos capturados    'muestra_folio = funcion que muestra el folio de
la solicitud capturada     function muestr_datos_grale(folio,clase)          response.Write "<script
language=javascript>"          response.Write "  function ver_historial(folio){"
     response.Write "
     open_window('../historial/repor_histo.asp?folio='+folio+'','historial','width=900,height=300,scrollbars=y
es,resizable=yes,status=yes');"          response.Write "  }"          response.Write "</script>"
     dim sql          dim res          dim reg_folio          dim reg_solicitante          dim
reg_monto          dim reg_moneda          dim reg_moneda_pago          dim reg_status
     dim reg_territorio          dim reg_descripcion          dim reg_tipo_operacion
          sql = "exec [dbo].sp_muestra_datos_grale "&folio&" "          set res = connect.execute(sql)
          if not res.eof then          reg_folio = res(0)          reg_solicitante =
res(1)          reg_moneda = res(3)          reg_moneda_pago = res(4)
          reg_status = res(5)          reg_territorio = res(6)
     reg_tipo_operacion = res(7)          reg_descripcion = res(8)          else
          reg_folio = "Indefinido"          reg_solicitante = "No registrado"
     reg_monto = 0          reg_moneda = "No registrada"
     reg_moneda_pago = "No registrada"          reg_status = "Indefinido"
     reg_territorio = "No registrado"          reg_tipo_operacion = "Indefinido"
     reg_descripcion = "Indefinida"          end if          response.Write
"<table border=0 width=""100%"">"          response.Write "  <tr class="&clase&">"
     response.Write "          <td width=""25%"">Tipo de operacion</td>"          response.Write "
     <td>"&reg_tipo_operacion&"</td>"          response.Write "          <td valign=top rowspan=8
valign=top>"          response.Write "          <table border=0 width=""100%"" valign=top>"
          response.Write "          <tr valign=top>"          response.Write "
                    <td valign=top>"
                    select case tipo_operacion
                                        case "anticipo"
     : call antic_muestr_datos_parti(folio,clase)

     'Localizacion = "../anticipo/antic_muestr_datos_parti.asp"
                                        case "pago"
     : call pago_muestr_datos_parti(folio,clase)

     'Localizacion = "../pago/pago_muestr_datos_parti.asp"
                                        case "reservacion"          :
call reservacion_muestr_datos_parti(folio,clase)

     'Localizacion = "../reservacion/reservacion_muestr_datos_parti.asp"
```

```
                case "compra"                   : call compra_muestr_datos_parti(folio,clase)

                                                'Localizacion =
"../compra/compra_muestr_datos_parti.asp"



                        end select              response.Write "
        </td>"          response.Write "                        </tr>"          response.Write "
                </table>"               response.Write "        </td>"
        response.Write " </tr>"                                 response.Write " <tr>"
                                                        call muestra_folio("Folio",reg_folio,clase)
                                                                'Localizacion =
"../includes/muestr_folio.asp"
                response.Write " </tr>"                         response.Write " <tr class="&clase&">"
                response.Write "        <td>Solicitante</td>"   response.Write "
        <td>"&reg_solicitante&"</td>"   response.Write " </tr>"         response.Write " <tr
class="&clase&">"               response.Write "        <td>Monto</td>"         response.Write "
        <td>"&formatcurrency(reg_monto,2)&"</td>"       response.Write " </tr>"
        response.Write " <tr class="&clase&">"          response.Write "        <td>Moneda de la
solicitud</td>"         response.Write "        <td>"&reg_moneda&"</td>"
        response.Write " </tr>"                 response.Write " <tr class="&clase&">"
        response.Write "        <td>Moneda del pago</td>"       response.Write "
        <td>"&reg_moneda_pago&"</td>"           response.Write " </tr>"
        response.Write "        <td>Estado</td>"        response.Write "
        <td>"&reg_status&"</td>"        response.Write " </tr>"                 response.Write " <tr
class="&clase&">"               response.Write "        <td>Territorio</td>"
        response.Write "        <td>"&reg_territorio&"</td>"            response.Write " </tr>"
                response.Write " <tr class="&clase&" valign=top>"       response.Write "
        <td>Descripcion</td>"           response.Write "        <td
colspan=3>"&reg_descripcion&"</td>"             response.Write " </tr>"
        response.Write " <tr class="&clase&" valign=top>"       response.Write "
        <td>Historial</td>"             response.Write "        <td><a
href=javascript:ver_historial("&folio&");>Ver historial</a></td>"       response.Write " </tr>"
                        response.Write "</table>"                       end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_moned.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion        : 5 de julio 2007   'Funcionalidad general
                : Esta funcion muestra el catalgo de monedas de la aplicacion           'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto           'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
                function catalogo_moneda(mensaje,nombre_objeto,valor,clase,tipo_operacion)              dim
sql                                             dim res                                 dim arRes
                        dim t_arRes                             dim count
        dim seleccionar                                 sql = "exec [dbo].sp_moneda_catalogo 'u' "
        set res = connect.execute(sql)          if not res.eof then             arRes = res.getrows
                else                            t_arRes = -1            end if                          if
t_arRes >= 0 then                       response.Write "<td class="&clase&">"
        response.Write " <span>"&mensaje&"</span>"
        response.Write "</td>"                  response.Write "<td>"
        response.Write " <select class=caja_texto name="&nombre_objeto&"
onchange=javascript:selecciona_moneda("'&tipo_operacion&"',this.value);>"
        response.Write "        <option value=0>[SELECCIONE UNA MONEDA]</option>"
                        for count = 0 to t_arRes                        if trim(arRes(0,count)) =
trim(valor) then                        seleccionar = "selected"                                        else
                                        seleccionar = ""                        end if
                response.Write " <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
```

```
        next                    response.Write " </select>"
        response.Write "</td>"          else                    response.Write "<td class="&clase&">"
                        response.Write " <span>"&mensaje&"</span>"
        response.Write "</td>"                  response.Write "<td class=aviso>"
        response.Write " No hay monedas registradas"                    response.Write " <input
type=hidden name="&nombre_objeto&" value="""">"                                 response.Write
"</td>"                 end if  end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_empre.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion      : 27 de julio 2007 'Funcionalidad general
                : Esta funcion muestra el catalgo de monedas de la aplicacion           'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto           'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
                function catalogo_empresa(mensaje,nombre_objeto,valor,clase)                    dim sql
                        dim res                                 dim arRes
                dim t_arRes                             dim count                                       dim
seleccionar                             sql = "exec [dbo].sp_empresa_catalogo 'u' "             set
res = connect.execute(sql)          if not res.eof then                 arRes = res.getrows
        t_arRes = ubound(arRes,2)               else                    t_arRes = -1                end
if                              if t_arRes >= 0 then                    response.Write "<td
class="&clase&">"                       response.Write " <span>"&mensaje&"<span>"
                response.Write "</td>"                  response.Write "<td>"
        response.Write " <select class=caja_texto name="&nombre_objeto&">"
        response.Write "            <option value=0>[SELECCIONE UNA EMPRESA]</option>"
                for count = 0 to t_arRes                        if arRes(0,count) = valor then
                        seleccionar = "selected"                        else
                        seleccionar = ""                        end if
        response.Write " <option "&seleccionar&" value="arRes(0,count)&">"&arRes(1,count)&"</option>"
                                                next                    response.Write "
        </select>"                              response.Write "</td>"          else
        response.Write "<input type=hidden name="&nombre_objeto&" value="""">"          end if     end
function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_ivas.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion      : 5 de julio 2007   'Funcionalidad general
                : Esta funcion muestra el catalgo de ivas de la aplicacion              'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto           'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
                function catalogo_ivas(mensaje,nombre_objeto,valor)                     dim sql
                        dim res                                 dim arRes
        dim t_arRes                             dim count                                       dim
seleccionar                             sql = "exec [dbo].sp_iva_catalogo 'u' "                 if
not res.eof then                arRes = res.getrows                     t_arRes = ubound(arRes,2)
                else                    t_arRes = -1                end if                          if
t_arRes >= 0 then                       response.Write "<td class=texto_nivel_2>"
        response.Write " <span>"&mensaje&"</span>"
        response.Write "</td>"                  response.Write "<td>"
        response.Write " <select class=caja_texto name="&nombre_objeto&"
onchange=javascript:suma_total();>"                     response.Write "            <option
value=0/0>[SELECCIONE UN IVA]</option>"                                 for count = 0 to
t_arRes                         if ucase(trim(arRes(0,count))) = ucase(trim(valor)) then
                        seleccionar = "selected"                        else
                seleccionar = ""                        end if
        response.Write " <option "&seleccionar&"
value="""&arRes(0,count)&"/"&arRes(1,count)&""">"&arRes(2,count)&"</option>"
                                        next                    response.Write " </select>"
```

```
                response.Write "</td>"          else                    response.Write "<input
type=hidden name="&nombre_objeto&" value="""">"          end if     end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_tipo_conce.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%              'Fte_ - -  'Fecha de ultima modificacion        : 21 de julio 2007 'Funcionalidad general
                : Esta funcion muestra el catalgo de tipos de concepto disponibles por empresa
        'Recibe  'empresa = Empresa en la que se esta trabajando    'mensaje = Leyenda que antecede al
cuadro de texto   'nombre_objeto = nombre del objeto          'valor = Valor a presentar en caso de que haya
un registro capturado       'clase = clase de estilo con que se presentara el texto                 function
catalogo_tipo_concepto(idempresa,mensaje,nombre_objeto,valor,clase,liga)          dim sql
                dim res                                 dim arRes
        dim t_arRes                             dim count                               dim
seleccionar                     dim liga_abre                                           sql = "exec
[dbo].sp_tipo_concepto_catalogo "&idempresa&", 'u' "            set res = connect.execute(sql)
        if not res.eof then             arRes = res.getrows             t_arRes = ubound(2)
                else                    t_arRes = -1            end if                          if
t_arRes >= 0 then                       response.Write "<td class="&clase&">"                   if
liga = 1  then                          liga_abre = "<a
href=javascript:catalogo_tipo_concepto("&idempresa&");>(ver)</a>"                       end if
        response.Write "  <span>"&mensaje&" "&liga_abre&"</span>"
        response.Write "</td>"                  response.Write "<td>"
        response.Write "  <select class=caja_texto name="&nombre_objeto&">"
        response.Write "          <option value=0>[SELECCIONE UN TIPO DE CONCEPTO]</option>"
                                for count = 0 to t_arRes                                if arRes(0,count) =
valor then                              seleccionar = "selected"                                else
                                seleccionar = ""                        end if
                response.Write "  <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
        next                    response.Write "  </select>"
        response.Write "</td>"          else                    response.Write "<td class="&clase&">"
                        response.Write "  <span>"&mensaje&"</span>"
        response.Write "</td>"                  response.Write "<td class=aviso>"
        response.Write "  No hay tipos de conceptos registrados"                response.Write "
        <input type=hidden name="&nombre_objeto&" value="""">"
        response.Write "</td>"          end if     end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/funci_pagin.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      sub button(intPrev,intNext,TotalPages)          dim Counter
        response.Write "<script language=javascript>"           response.Write "  function index(index){
                response.Write "        document.form.idempresa.value = '"&request("idempresa")&"';"
                response.Write "        document.form.numempleado.value =
'"&request("numempleado")&"';"                          response.Write "
        document.form.tipo_operacion.value = '"&request("tipo_operacion")&"';"
        response.Write "        document.form.idmoneda.value = '"&request("idmoneda")&"';"
        response.Write "        document.form.fecha_ini.value = '"&request("fecha_ini")&"';"
        response.Write "        document.form.fecha_fin.value = '"&request("fecha_fin")&"';"
        response.Write "        document.form.folio.value = '"&request("folio")&"';"
        response.Write "        document.form.no_reg.value = '"&no_reg&"';"
        response.Write "        if(document.form.idproveedor)"          response.Write "
        document.form.idproveedor.value = '"&request("idproveedor")&"';"        response.Write "
        if(document.form.depto)"        response.Write "                document.form.depto.value =
'"&request("depto")&"';"                        response.Write "        if(document.form.factura)"
        response.Write "                document.form.factura.value = '"&request("factura")&"';"
                                                                        response.Write "
        document.form.action = 'repor_autor_usuar.asp?PageIndex='+index+";"
        response.Write "        document.form.submit();"        response.Write "  }"
```

```
        response.Write "</script>"                                                  response.Write "<tr
class=texto_nivel_2>"           response.Write "  <td colspan=8 align=right>"              if intPrev <>
0 then                  response.Write "  <a href=javascript:index("&intPrev&");>&lt;&lt;</a>"
            end if          if rst.AbsolutePage <> -3 then                  response.Write "  <a
href=javascript:index("&intNext&");>&gt;&gt;</a>"                           end if
        response.Write "  </td>"            response.Write "</tr>"            Counter=1
        response.Write "<tr class=texto_nivel_2>"          response.Write "  <td colspan=8 align=right>"
            response.Write "          Ir a la hoja: "           while Counter <= TotalPages
            if cint(Counter)=cint(PageIndex) then                   Response.Write
Counter                 else                    Response.Write "<a
href=javascript:index("&counter");>"&counter&"</a>"
                                        end if                                      if
cint(Counter) <> cint(TotalPages) then                  Response.Write " | "
                                                                    end if
            Counter = Counter + 1           wend                response.Write "  </td>"
                                    response.Write "</tr>"                          end
sub%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/muestr_tipo_conce.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion      : 5 de julio 2007   'Funcionalidad general
        : Esta funcion presenta las diferentes caracteristicas que le corresponde          '
                                        a cada concepto de gasto. Por ejemplo: si se
selecciona el concepto    '                                                         de
kilometraje puede presentar ciertos campos para que el usuario        '
                                    ingrese informacion que no le pedira cuando seleccione      '
                                            el concepto de comidas  '
                                        Si se trata de un concepto de gasto seleccionado no
capturado, la funcion       '                                                       pide al
usuario ingresar ciertos datos      '                                                   Si
se selecciona un detalle, la funcion cargara los datos existentes       '
                                    previamente capturados para ese detalle         'Recibe  'idempresa =
empresa en la que se esta trabajando      'numempleado = numero de empleado que hace la operacion
        'idconcepto = concepto de gasto seleccionado      'idmoneda = moneda de la solicitud'idterritorio =
territorio en que se hizo o se hara el gasto, dependiendo del tipo de solicitud    'tipo_operacion = tipo de
solicitud (anticipo, pago, compra, etc)      'folio_detalle = Folio del detalle en el que se esta trabajando
        'clase = Clase en la que se presentara el texto             'Funciones relacionadas
        'catalogo_ivas = funcion que presenta el catalogo de ivas de la aplicacion      function
muestra_tipo_concepto(idempresa,numempleado,idconcepto,idmoneda,idterritorio,tipo_operacion,folio_detalle
,clase)             dim sql         dim res         dim ask_boleto_avion            dim origen
            dim destino             dim hora_salida         dim aerolinea           dim
ask_kilometraje         dim total_kilometros            dim costo_por_kilometro         dim
ask_precio_unitario_cantidad            dim idkm            dim no_viajes           dim
precio_unitario         dim unidades            dim subtotal            dim ask_iva             dim
iva             dim idiva           dim ask_propina         dim propina             dim ask_tua
            dim tua         dim total           dim ask_dias            dim dias
        dim ask_comensales          dim comensales          dim ask_deposito        dim
folio_deposito                      dim subtotal_diario             dim monto_limite
            dim f_detalle           dim subtotal_readonly           dim titulo              dim
tabla_adicional                     tabla_adicional = 0                             if
folio_detalle <> "" then            f_detalle = folio_detalle           else
        f_detalle = 0           end if                  subtotal_readonly = ""
        sql = "exec [dbo].sp_tipo_concepto_define
"&idempresa&","&numempleado&","&idconcepto&","&idmoneda&","&idterritorio&","&tipo_operacion&","&f_deta
lle&" "         set res = connect.execute(sql)          if not res.eof then
        ask_boleto_avion = res("ask_boleto_avion")          origen = res("origen")
        destino = res("destino")            hora_salida = res("hora_salida")
        aerolinea = res("aerolinea")            ask_kilometraje = res("ask_kilometraje")
            total_kilometros = res("total_kilometros")          costo_por_kilometro =
res("costo_por_kilometro")          idkm = res("idkm")              no_viajes =
```

```
res("no_viajes")                    ask_precio_unitario_cantidad = res("ask_precio_unitario_cantidad")
            precio_unitario = res("precio_unitario")                    unidades = res("unidades")
                  subtotal = res("subtotal")                    ask_iva = res("ask_iva")
      iva = res("iva")                    idiva = res("idiva")                    ask_propina =
res("ask_propina")                    propina = res("propina")                    ask_tua =
res("ask_tua")                    tua = res("tua")                    total = res("total")
      ask_dias = res("ask_dias")                    dias = res("dias")                    ask_comensales =
res("ask_comensales")                    comensales = res("comensales")
      ask_deposito = res("ask_deposito")                    folio_deposito = res("folio_deposito")
            subtotal_diario = res("subtotal_diario")                    monto_limite =
res("monto_limite")                    else                    ask_boleto_avion = 0
      origen = ""                    destino = ""                    hora_salida = ""
      aerolinea = ""                    ask_kilometraje = 0                    total_kilometros = 0
                  costo_por_kilometro = 0                    ask_precio_unitario_cantidad = 0
            idkm = 0                    no_viajes = 0                    precio_unitario = 0
            unidades = 1                    subtotal = 0                    ask_iva = 0
            iva = 0                    idiva = "0/0"                    ask_propina = 0
      propina = 0                    ask_tua = 0                    tua = 0                    total
= 0                  ask_dias = 0                    dias = 1                    ask_comensales = 0
                  comensales = 1                    ask_deposito = 0
      folio_deposito = 0                    subtotal_diario = 0                    monto_limite
= -1            end if                    if ask_boleto_avion = 1 then
      response.Write "<tr>"                    call
muestra_texto("Origen","origen",origen,clase,20,50)
      call muestra_texto("Destino","destino",destino,clase,20,50)
                  response.Write "</tr>"
      response.Write "<tr>"                    call
muestra_texto("Aerolinea (opcional)","aerolinea",aerolinea,clase,20,50)
                  call muestra_texto("Hora de salida","hora_salida",hora_salida,clase,20,50)
                  response.Write "</tr>"
      response.Write "<input type=hidden name=ask_boleto_avion value=1>"
                  else                    response.Write
"<input type=hidden name=origen value="""">"                    response.Write "<input type=hidden
name=destino value="""">"                    response.Write "<input type=hidden name=hora_salida
value="""">"                    response.Write "<input type=hidden name=aerolinea value="""">"
            response.Write "<input type=hidden name=ask_boleto_avion value=0>"
            end if                    if ask_kilometraje = 1 then
      subtotal_readonly = "readonly"                    response.Write "<tr class="&clase&">"
            response.Write "  <td>"                    response.Write "
      Kilometraje"                    response.Write "  </td>"
      response.Write "  <td>"                    response.Write "            <a
href=javascript:open_window('../kilometro/catal_kilom.asp?idempresa="&idempresa&"&idkm="&idkm&"&no_vi
ajes="&no_viajes&"&idmoneda="&idmoneda&"','Kilometraje','width=500,height=300,resizable=yes,status=yes')
;>Ver catalogo de distancias</a>"                    response.Write "  </td>"
      response.Write "</tr>"                    response.Write "<tr>"
      response.Write "  <td class="&clase&">"                    response.Write "
      Total de kilometros"                    response.Write "  </td>"
      response.Write "  <td>"                    response.Write "
      <input "&subtotal_readonly&" class=caja_texto_numeros type=text name=kilometraje_reco
value="&total_kilometros&" onblur=javascript:suma_total();>"                    response.Write "  </td>"
            response.Write "</tr>"                    response.Write "<tr>"
      response.Write "  <td class="&clase&">"                    response.Write "
      Costo por kilometro"                    response.Write "  </td>"
      response.Write "  <td>"                    response.Write "
      <span class="&clase&">$ </span><input "&subtotal_readonly&" class=caja_texto_numeros type=text
name=kilometraje_costo value="&costo_por_kilometro&" onblur=javascript:suma_total();>"
      response.Write "  </td>"                    response.Write "</tr>"
      response.Write "<input type=hidden name=idkm value="&idkm&">"
      response.Write "<input type=hidden name=no_viajes value="&no_viajes&">"
      response.Write "<input type=hidden name=ask_kilometraje value=1>"
            else                    response.Write "<input type=hidden name=kilometraje
value=0>"                    response.Write "<input type=hidden name=kilometraje_reco value=0>"
```

```
                response.Write "<input type=hidden name=kilometraje_costo value=0>"
        response.Write "<input type=hidden name=ask_kilometraje value=0>"           end if
                        if ask_precio_unitario_cantidad = 1 then                    subtotal_readonly =
"readonly"                      response.Write "<tr>"                   response.Write " <td
class="&clase&">"                                       response.Write "        Unidades"
                response.Write " </td>"                                 response.Write " <td>"
                        response.Write "        <input class=caja_texto_numeros type=text
name=unidades value="&unidades&" onblur=javascript:suma_total();>"              response.Write "
        </td>"                  response.Write "</tr>"
        response.Write "<tr>"                           response.Write "        Precio Unitario"
        response.Write " </td>"                         response.Write " <td>"
                        response.Write "        <span class="&clase&">$ </span><input
class=caja_texto_numeros type=text name=precio_unitario value="&precio_unitario&"
onblur=javascript:suma_total();>"               response.Write " </td>"
        response.Write "</tr>"                          response.Write "<input type=hidden
name=ask_precio_unitario_unidades value=1>"             else                    response.Write
"<input type=hidden name=precio_unitario value=0>"              response.Write "<input type=hidden
name=unidades value=0>"                         response.Write "<input type=hidden
name=ask_precio_unitario_unidades value=0>"             end if                                  if
ask_deposito = 1 then                   subtotal_readonly = "readonly"
        response.Write "<tr class="&clase&">"                   response.Write " <td>"
                        response.Write "        Folio de deposito"
        response.Write " </td>"                         response.Write " <td>"
                        response.Write "        <a
href=javascript:open_window('../deposito/catal_depos.asp?idempresa="&idempresa&"&numempleado="&num
empleado&"&folio_deposito="&folio_deposito&"&idmoneda="&idmoneda&"','Kilometraje','width=500,height=30
0,resizable=yes,status=yes');>Ver depositos realizados</a>"              response.Write " </td>"
                response.Write "</tr>"                  response.Write "<input type=hidden
name=folio_deposito value="&folio_deposito&">"                  response.Write "<input
type=hidden name=ask_deposito value=1>"                                 else
                response.Write "<input type=hidden name=folio_deposito value=0>"
        response.Write "<input type=hidden name=ask_deposito value=0>"          end if
                        response.Write "<tr>"           response.Write " <td
class="&clase&">"               response.Write "        Subtotal"               response.Write "
        </td>"                  response.Write " <td>"
        response.Write "        <span class="&clase&">$ </span><input "&subtotal_readonly&"
class=caja_texto_numeros type=text name=subtotal value="&subtotal&" onblur=javascript:suma_total();>"
        response.Write " </td>"         response.Write "</tr>"                                  if
ask_iva = 1 then                response.Write "<tr>"                   response.Write " <td
class="&clase&">"                                       response.Write "        IVA"
                response.Write " </td>"                 response.Write " <td>"
        response.Write "        <span class="&clase&">$ </span><input class=caja_texto_numeros
readonly type=text name=iva value="&iva&">"                     response.Write " </td>"
                                                          call
catalogo_ivas("Porcentaje","valor_iva",idiva)
                'Localizacion = "../includes/catal_ivas.asp"            response.Write "</tr>"
        response.Write "<input type=hidden name=ask_iva value=1>"       else
        response.Write "<input type=hidden name=iva value=0>"                   response.Write
"<input type=hidden name=ask_iva value=0>"              end if
                        if ask_propina = 1 then                 response.Write "<tr>"
        response.Write " <td class="&clase&">"
        response.Write "        Propina"                response.Write " </td>"
        response.Write " <td>"                  response.Write "        <span class="&clase&">$
</span><input class=caja_texto_numeros type=text name=propina value="&propina&"
onblur=javascript:suma_total();>"               response.Write " </td>"
        response.Write "</tr>"                  response.Write "<input type=hidden name=ask_propina
value=1>"               else                    response.Write "<input type=hidden name=propina
value=0>"                       response.Write "<input type=hidden name=ask_propina value=0>"
        end if                                  if ask_tua = 1 then                     response.Write "<tr>"
                        response.Write " <td class="&clase&">"
        response.Write "        TUA"                    response.Write " </td>"
        response.Write " <td>"                  response.Write "        <span class="&clase&">$
```

```
</span><input class=caja_texto_numeros type=text name=tua value="&tua&"
onblur=javascript:suma_total();>"                    response.Write " </td>"
        response.Write "</tr>"                       response.Write "<input type=hidden name=ask_tua
value=1>"                   else                     response.Write "<input type=hidden name=tua value=0>"
                            response.Write "<input type=hidden name=ask_tua value=0>"          end if
                            response.Write " <tr>"           response.Write "          <td
class="&clase&">"                                    response.Write "          Total"
        response.Write "          </td>"             response.Write "          <td>"
                            response.Write "          <span class="&clase&">$ </span><input
class=caja_texto_numeros readonly type=text name=total value="&total&">"          response.Write "
        </td>"              response.Write " </tr>"          response.Write "</table>"
                                  if (ask_dias = 1 or ask_comensales = 1) and monto_limite = -1 then
                                  titulo = "Informaci&oacute;n adicional"          else
        if monto_limite = -1 then                        titulo = ""                        else
                            titulo = "Evaluaci&oacute;n de pol&iacute;ticas de montos l&iacute;mite"
                end if            end if                 if trim(titulo) <> "" then
                tabla_adicional = 1                                 response.Write
"<table width=""100%"" align=center>"                               response.Write " <tr
height=5>"                        response.Write " </tr>"
        response.Write " <tr>"                   response.Write "          <td colspan=5
class=tabla_nivel_2>"                                response.Write
        titulo                    response.Write "          </td>"
        response.Write " </tr>"                          response.Write " <tr height=1>"
        response.Write " </tr>"                          response.Write "</table>"
                            end if                       if tabla_adicional = 1 then
        response.Write "<table width=""100%"" align=center class=contenedor>"
                                                         end if                    if ask_dias =
1 then                      response.Write "<tr>"        response.Write " <td class="&clase&"
width=220>"                                          response.Write "          No. de d&iacute;as"
                response.Write " </td>"                  response.Write " <td>"
                            response.Write "          <input class=caja_texto_numeros type=text
name=dias value="&dias&" onblur=javascript:suma_total();>"          response.Write " </td>"
                response.Write "</tr>"                   response.Write "<input type=hidden
name=ask_dias value=1>"                  else            response.Write "<input type=hidden
name=dias value=1>"                      response.Write "<input type=hidden name=ask_dias value=0>"
                            end if
        if ask_comensales = 1 then                       response.Write "<tr>"
        response.Write " <td class="&clase&" width=220>"
        response.Write "          Comensales"            response.Write " </td>"
                response.Write " <td>"                                      response.Write "
        <input class=caja_texto_numeros type=text name=comensales value="&comensales&"
onblur=javascript:suma_total();>"                    response.Write " </td>"
        response.Write "</tr>"                           response.Write "<input type=hidden
name=ask_comensales value=1>"                            else                response.Write
"<input type=hidden name=comensales value=1>"                    response.Write "<input type=hidden
name=ask_comensales value=0>"                            end if
                if (ask_dias = 1 or ask_comensales = 1) and monto_limite <> -1 then
        response.Write "<tr>"                    response.Write " <td class="&clase&" width=220>"
                            response.Write "          Subtotal diario"
        response.Write " </td>"                          response.Write " <td>"
                            response.Write "          <span class="&clase&">$ </span><input
class=caja_texto_numeros readonly type=text name=subtotal_diario value="&subtotal_diario&">"
        response.Write " </td>"                  response.Write "</tr>"          else
        response.Write "          <input class=caja_texto_numeros readonly type=hidden
name=subtotal_diario value="&subtotal_diario&">"                                          end if
                            if monto_limite <> -1 then   response.Write "<tr>"
        response.Write " <td class="&clase&" width=220>"
        response.Write " </td>"                          response.Write " <td>"
                            response.Write "          <span class="&clase&">$ </span><input
class=caja_texto_numeros readonly type=text name=monto_limite value="&monto_limite&">"
        response.Write " </td>"                                     response.Write "</tr>"
        else                      response.Write "          <input class=caja_texto_numeros readonly
```

```
type=hidden name=monto_limite value="&monto_limite&">"                end if
        if tabla_adicional = 1 then                    response.Write "</table>"
                                                                end if                                                  end
function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_centr_costo.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%      'Fte_ - -  'Fecha de ultima modificacion        : 5 de julio 2007   'Funcionalidad general
        : Esta funcion muestra el catalgo de centros de costo de la aplicacion  'Recibe  'idempresa =
empresa en la que se esta trabajando        'mensaje = Leyenda que antecede al cuadro de texto
        'nombre_objeto = nombre del objeto          'valor = Valor a presentar en caso de que haya un
registro capturado          'clase = clase de estilo con que se presentara el texto          function
catalogo_centro_costo(idempresa,mensaje,nombre_objeto,valor,clase,proviene)                dim sql
        dim res             dim arRes               dim t_arRes             dim count                   dim
seleccionar             sql = "exec [dbo].sp_centro_costo_catalogo "&idempresa&", '"&proviene&"'"
        set res = connect.execute(sql)              if not res.eof then                 arRes = res.getrows
                else                        t_arRes = -1            end if              if t_arRes >= 0 then
                        response.Write "<td class="&clase&">"                   response.Write "
        <span>"&mensaje&"</span>"                   response.Write "</td>"
        response.Write "<td>"                       response.Write "  <select class=caja_texto
name="&nombre_objeto&">"                    response.Write "            <option
value=0>[SELECCIONE UN CENTRO DE COSTO]</option>"                   for count = 0 to t_arRes
                        if arRes(0,count) = valor then                                  seleccionar
= "selected"                                else                                        seleccionar = ""
                        end if                                  response.Write "  <option "&seleccionar&"
value="&arRes(0,count)&">"&arRes(1,count)&" -- "&arRes(2,count)&"</option>"                 next
                response.Write "  </select>"                    response.Write "</td>"              else
                        response.Write "<input type=hidden name="&nombre_objeto&" value="""">"
        end if      end function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/catal_nivel.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

<%                  'Fte_ - -  'Fecha de ultima modificacion        : 27 de julio 2007 'Funcionalidad general
                    : Esta funcion muestra el catalgo de monedas de la aplicacion           'Recibe  'mensaje =
Leyenda que antecede al cuadro de texto    'nombre_objeto = nombre del objeto          'valor = Valor a
presentar en caso de que haya un registro capturado'clase = clase de estilo con que se presentara el texto
                    function catalogo_nivel(idempresa,mensaje,nombre_objeto,valor,clase)                    dim
sql                                     dim res                                     dim arRes
                    dim count                                   dim seleccionar
                    sql = "exec [dbo].sp_nivel_catalogo "&idempresa&" "             set res = connect.execute(sql)
                    if not res.eof then                 arRes = res.getrows                             t_arRes =
ubound(arRes,2)             else                        t_arRes = -1                end if
        if t_arRes >= 0 then                            response.Write "  <span>"&mensaje&"</span>"
                                    response.Write "</td>"                  response.Write "<td>"
                    response.Write "  <select class=caja_texto name="&nombre_objeto&">"
        response.Write "            <option value=0>[SELECCIONE UN NIVEL]</option>"
                    for count = 0 to t_arRes                                if arRes(0,count) = valor then
                                    seleccionar = "selected"                                    else
                    seleccionar = ""                                        end if
        response.Write "  <option "&seleccionar&" value="&arRes(0,count)&">"&arRes(1,count)&"</option>"
                                                            next                    response.Write "
        </select>"                                  response.Write "</td>"          else
        response.Write "<input type=hidden name="&nombre_objeto&" value="""">"                  end if     end
function%>
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../distribucion/distr_detal.asp
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-
```

```
<%@language="vbscript"%><%Option Explicit 'Esta sentencia obliga a declar variables%><%
	'Fte_ - -		'Fecha de ultima modificacion	: 5 de julio 2007		'Funcionalidad general		: Pagina mediante la cual se distribuyen los gastos		'
en uno o mas centros de costo%><!--	#include file="../connect/conne_open.asp"--><!--	Este archivo contiene la conexion a la BD--><!--	#include file="../includes/catal_centr_costo.asp"--><!--	Este archivo incluye la funcion que muestra el catalogo de centros de costo--><!-- #include file="../includes/muestr_boton.asp"--><!--	Este archivo incluye la funcion que los botones--><%
	'=============================================================================
===============	'Estas variables son comunes a todas las paginas	dim numempleado		'Numero de empleado que esta firmado en la aplicacion	dim idempresa
	'Numero de empresa en la que se esta trabajando
	numempleado = request("numempleado")	idempresa = request("idempresa")
	'=============================================================================
===============
	'=============================================================================
===============	'Estas variables se usan nada mas en las paginas de solicitudes
	dim folio		'Numero de folio de la solicitud	folio = request("folio")
	'=============================================================================
===============
	'=============================================================================
===============	'Estas variable guarda el valor del identificador de la distribucion en la tabla 'distribucion_solicitud	dim iddistribucion
	'Identificador de la distribucion	iddistribucion = request("iddistribucion")

	'=============================================================================
===============
	'=============================================================================
===============	'Estas variables se usan nada mas en las paginas del detalle de las solicitudes	dim detalle
	'Numero de detalle dentro de la solicitud	detalle = request("detalle")
	'=============================================================================
===============
	'=============================================================================
===============	'Estas variables se usa para saber de donde proviene la peticion de esta pagina, se usa en	'la mayoria de las paginas	dim proviene	proviene = request("proviene")
	'=============================================================================
===============	function guarda_distribucion(iddistribucion,detalle,centro_costo,porcentaje,folio,idempresa,numempleado)
	'Esta funcion inserta en la tabla distribucion_solicitud las distribuciones de los detalles
	'a los diferentes centros de costo		'Recibe:		'iddistribucion = identificador de la distribucion		'detalle = identificador del detalle que se va a distribuir
	'centro_costo = centro de costo al que se hace la distribucion
	'porcentaje = porcentaje a distribuir		'folio = folio al que pertenece el detalle que se esta distribuyendo		'idempresa = empresa en la que se esta operando
	'numempleado = numero de empleado que hace la distribucion		dim sql
		dim res		sql = "exec [dbo].sp_distribucion_inserta_modifica "&iddistribucion&","&detalle&","&centro_costo&","&porcentaje&","&folio&","&idempresa&","&numempleado&",'i'"
		set res = connect.execute(sql)		if not res.eof then
		response.Write "<table align=center>"		response.Write "	<tr>"		response.Write "	<td class=aviso align=center>"&res(0)&"</td>"		response.Write "	</tr>"
		response.Write "</table>"		else		end if		end function
		function elimina_distribucion(iddistribucion,detalle,centro_costo,porcentaje,folio,idempresa,numempleado)
	'Esta funcion elimina la distribucion de ese detalle		'Recibe:
	'iddistribucion = identificador de la distribucion		'detalle = identificador del detalle que se va a distribuir		'centro_costo = centro de costo al que se hace la
```

```
distribucion                    'porcentaje = porcentaje a distribuir                    'folio = folio al que
pertenece el detalle que se esta distribuyendo                    'idempresa = empresa en la que se
esta operando                   'numempleado = numero de empleado que hace la distribucion
        dim sql                 sql = "exec [dbo].sp_distribucion_inserta_modifica
"&iddistribucion&","&detalle&","&centro_costo&","&porcentaje&","&folio&","&idempresa&","&numempleado&",'d
' "                     connect.execute(sql)            end function            function
muestra_distribuciones_existentes(detalle)              'Esta funcion muestra las distribuciones de ese
detalle                 'Recibe:                'detalle = identificador del detalle en que se estra
trabajando              dim sql                 dim res                 dim arRes
        dim t_arRes                     dim count                       dim centro
        dim total_porcentaje                    set res = connect.execute(sql)
    if not res.eof then                 arRes = res.getrows
    t_arRes = ubound(arRes,2)                   else                            t_arRes = -1
                    end if                  if t_arRes >= 0 then
    total_porcentaje = 0                        response.Write "<tr class=tabla_nivel_3>"
                    response.Write "  <td>Clave</td>"           response.Write "
    <td>Centro de costo</td>"                   response.Write "  <td>Porcentaje</td>"
                    response.Write "  <td>Eliminar</td>"
    response.Write "</tr>"                      for count = 0 to t_arRes
            total_porcentaje = total_porcentaje + arRes(2,count)
    response.Write "<tr class=texto_nivel_2>"                       response.Write "
    <td>"&arRes(0,count)&"</td>"                            response.Write "
    <td>"&arRes(1,count)&"</td>"                            response.Write "
    <td>"&arRes(2,count)&" %</td>"                          response.Write "  <td><a
href=javascript:eliminar_distribucion("&arRes(8,count)&")><img src=../images/cruz.gif border=0></a></td>"
                                    response.Write "</tr>"                  next
                    if t_arRes = 0 then                     centro = "centro"
                    else                            centro = "centros"
            end if                  response.Write "<tr>"
    response.Write "  <td colspan=4><hr></td>"              response.Write "</tr>"
                    response.Write "<tr class=texto_nivel_2>"
    response.Write "  <td>"&t_arRes + 1 &" "& centro & "  de costo</td>"
    response.Write "  <td></td>"                            response.Write "
    <td>"&total_porcentaje&" %</td>"                        response.Write "  <td></td>"
                    response.Write "</tr>"                  end if          end
function%><html>        <head>          <title>Cat&aacute;logo de centros de costo</title>
    <script language=javascript src="../includes/open_window.js"></script>          <script
src="../includes/no_right_mouse.js"></script>                   <link rel="stylesheet"
href="../stylesheet/style.css">                 <script language=javascript>            <!--
    function guardar_distribucion(){                        //Esta funcion valida que se
introduzcan datos validos antes de mandar a                     //distribuir
                    if(document.form.centro_costo.value==0){                        alert('Favor
de seleccionar un centro de costo');
    document.form.centro_costo.focus();                                             return false;
                    }
    if((isNaN(document.form.porcentaje.value)||(document.form.porcentaje.value<0)||(document.form.por
centaje.value==""))){                                   alert('Valor no valido para el porcentaje');
                                document.form.porcentaje.focus();
    return false;                           }                                       //Se manda a llamar
a si misma con el valor GD (Guarda distribucion) como                               //valor proviene. Este
valor se usa para hacer la insercion del registro de                            //distribucion
                    document.form.proviene.value = 'GD';                            document.form.action
= 'distr_detal.asp'                 document.form.submit();                         }
    function eliminar_distribucion(iddistribucion){                                 //Se manda a llamar
a si misma con el valor ED (Elimina distribucion) como                              //valor proviene. Este
valor se usa para hacer la eliminacion del registro                             //o los registros de distribucion
                                //iddistribucion = numero identificador del registro de distribucion
                    if (iddistribucion==0)                                          mensaje = '¿Esta
seguro de eliminar la distribucion totalmente?';                                    else
                    mensaje = '¿Esta seguro de eliminar este porcentaje de distribucion?'
    if(confirm(mensaje)){                                   document.form.proviene.value = 'ED';
                                document.form.iddistribucion.value = iddistribucion;
```

```
                document.form.action = 'distr_detal.asp'
        document.form.submit();                 }                 }                 -->
                </script></head> <body>  <form name="form" method="post">                 <input
type=hidden name=idempresa              value=<%=idempresa%>>           <input type=hidden
name=numempleado            value=<%=numempleado%>>         <input type=hidden
name=detalle                value=<%=detalle%>>             <input type=hidden name=proviene
        value=<%=proviene%>>            <input type=hidden name=folio
        value=<%=folio%>>               <input type=hidden name=iddistribucion
        value=<%=iddistribucion%>>              <table border=0 width=90.5%>                    <tr>
                <td colspan=3 class=tabla_nivel_2>
        Distribucion a centros de costo                 </td>                   </tr>
        </table>        <table border=0 width=90% class=contenedor>                     <tr>
                <%                              call
catalogo_centro_costo(idempresa,"Centros de Costo","centro_costo","","texto_nivel_2","u")
                'Funcion que muestra el catalogo de centros de costo
                'Localizacion = "../includes/muestr_catal_centr_costo.asp"                      %>
                </tr>                   <tr>                    <td
class=texto_nivel_2>                    Porcentaje
        </td>                   <td>                    <input type=text
class=caja_texto name=porcentaje size=6 maxlength=5>                    </td>
        </tr>           </table>        <table border=0 width=90%>                      <tr>
                <td colspan=4>                  <%                      if
proviene = "GD" then                    call
guarda_distribucion("0",detalle,request("centro_costo"),request("porcentaje"),folio,idempresa,0)
                'Funcion que guarda el registro de distribucion
        'Localizacion = esta misma pagina                end if                  if
proviene = "ED" then                    call
elimina_distribucion(request("iddistribucion"),detalle,0,0,0,0,0)                       'Funcion que
elimina el registro de distribucion             'Localizacion = esta misma pagina
                end if                  %>                      </td>
                </tr>                   <tr>                    <td colspan=4 align=center>
                        <%                      call
muestra_boton("Guardar","guardar_distribucion();")                      response.Write
" "                call muestra_boton("Eliminar
distribucion","eliminar_distribucion(0);")              response.Write " "
                        call muestra_boton("Cerrar","window.close();")
                'Funcion que muestra los botones de operacion
        'Localizacion = "../includes/muestr_boton.asp"                  %>
                </td>                   </tr>                   <tr>                    <td
colspan=4 align=center><hr></td>                <tr>                    <%
        call muestra_distribuciones_existentes(detalle)                 'Funcion que muestra
las distribuciones para el detalle en cuestion          'Localizacion = esta misma pagina
                %>              </table> </form>  </body></html>
```

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/open_windo.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

```
// JScript source code

//Esta funcion es para abrir ventanas
        function open_window(url,title,charac){
                window.open(url,title,charac);
        }
```

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/valida_politicas.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

```
// JScript source code

//Esta funcion es la que valida las politicas corporativas de los montos limite

	function valida_politicas(){
		//Evalua el subtotal diario solicitado vs. el monto limite permitido
		confirmacion = 0;
		subtotal_diario = eval(document.form.subtotal_diario.value);
		monto_limite_permitido = eval(document.form.monto_limite.value);

		if(monto_limite_permitido==-1){
			confirmacion = 1;
			document.form.incumplimiento.value = 0;
		}
		else{
			if(subtotal_diario>monto_limite_permitido){
				//Si el subtotal es mayor al monto limite permitido, se le pide el
consentimiento
				//del usuario para guardar el detalle, a expensas de que se esta
incumpliendo
				//la politica
				if(confirm("Con dicha informacion, se incurre en incumplimiento de
politicas...\n¿Desea continuar?")){
					confirmacion = 1;
					document.form.incumplimiento.value = 1;
				}
				else{
					confirmacion = 0 ;
					document.form.incumplimiento.value = 0;
				}
			}
			else{
				confirmacion = 1;
				document.form.incumplimiento.value = 0;
			}
		}
		return confirmacion;
		//Este valor es el determinante para guardar o no el detalle
		//1 = se guarda (con o sin incumplimiento de politicas)
		//0 = no se guarda
	}

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/valida_tipos_opera.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

// JScript source code

		function selecciona_tipo_operacion(tipo_operacion,dependencia){
			//Funcion que valida la seleccion de los tipos de operacion
			//Recibe
			//tipo_operacion = tipo de operacion que se selecciona

			var tipo_operacion_seleccionado =
'document.form.tipo_operacionMJ'+tipo_operacion+'';

			//Si se selecciona anticipo
			//Regla : Todos los conceptos en anticipo deben existir en las comprobaciones de
anticipo
			if (tipo_operacion=='anticipo'){
				if(dependencia=='cd'){
```

```
                                        if(eval(tipo_operacion_seleccionado).checked==true){

        if(document.form.tipo_operacionMJcomprobacion_anticipo){

        eval(document.form.tipo_operacionMJcomprobacion_anticipo).checked = true;
                                                }
                                        }
                                        else{

        if(document.form.tipo_operacionMJcomprobacion_anticipo){

        if(eval(document.form.tipo_operacionMJcomprobacion_anticipo).checked==true){

        eval(document.form.tipo_operacionMJanticipo).checked = false;


        eval(document.form.tipo_operacionMJcomprobacion_anticipo).checked = false;

                                                        }
                                                }
                                        }
                                }//Si no hay dependencia, es decir dependencia = sd
                                else{
                                        if(eval(tipo_operacion_seleccionado).checked==true){

        if(document.form.tipo_operacionMJcomprobacion_anticipo){

        //eval(document.form.tipo_operacionMJcomprobacion_anticipo).checked = true;
                                                }
                                        }
                                        else{

        if(document.form.tipo_operacionMJcomprobacion_anticipo){

        if(eval(document.form.tipo_operacionMJcomprobacion_anticipo).checked==false){

        //eval(document.form.tipo_operacionMJanticipo).checked = false;

                                                        }
                                                }
                                        }
                                }
                        }

                        //Si se selecciona comprobacion de anticipo
                        //Regla : No todos los conceptos de comprobacion de anticipos deben existir en
anticipos
                        if (tipo_operacion=='comprobacion_anticipo'){
                                if(dependencia=='cd'){
                                        if(eval(tipo_operacion_seleccionado).checked==true){
                                                if(document.form.tipo_operacionMJanticipo){

        eval(document.form.tipo_operacionMJanticipo).checked = true;
                                                }
                                        }
                                        else{
                                                if(document.form.tipo_operacionMJanticipo){

        if(eval(document.form.tipo_operacionMJanticipo).checked==true){

        eval(document.form.tipo_operacionMJanticipo).checked = false;
```

```
eval(document.form.tipo_operacionMJcomprobacion_anticipo).checked = false;

                                                }
                                        }
                                }
                        }//Si no hay dependencia, es decir dependencia = sd
                        else{
                                if(eval(tipo_operacion_seleccionado).checked==true){
                                        if(document.form.tipo_operacionMJanticipo){

//eval(document.form.tipo_operacionMJanticipo).checked = true;
                                        }
                                }
                                else{
                                        if(document.form.tipo_operacionMJanticipo){

if(eval(document.form.tipo_operacionMJanticipo).checked==true){

//eval(document.form.tipo_operacionMJanticipo).checked = false;

                                                }
                                        }
                                }
                        }
                }


                //Si se selecciona compra
                //Regla : Todos los conceptos en compra deben existir en los pagos

                if (tipo_operacion=='compra'){
                        if(eval(tipo_operacion_seleccionado).checked==true){
                                if(document.form.tipo_operacionMJpago){
                                        eval(document.form.tipo_operacionMJpago).checked =
true;
                                }
                        }
                        else{
                                if(document.form.tipo_operacionMJpago){

if(eval(document.form.tipo_operacionMJpago).checked==true){

eval(document.form.tipo_operacionMJcompra).checked = false;

                                        }
                                }
                        }
                }

                //Si se selecciona contrato
                //Regla : Todos los conceptos en contrato deben existir en los pagos

                if (tipo_operacion=='contrato'){
                        if(eval(tipo_operacion_seleccionado).checked==true){
                                if(document.form.tipo_operacionMJpago){
                                        eval(document.form.tipo_operacionMJpago).checked =
true;
                                }
                        }
                        else{
                                if(document.form.tipo_operacionMJpago){
```

```
if(eval(document.form.tipo_operacionMJpago).checked==true){

eval(document.form.tipo_operacionMJcontrato).checked = false;

                                                }
                                        }
                                }
                        }


                        //Si se selecciona pago
                        //Regla : No Todos los conceptos en pago deben existir en compras y contratos

                        if (tipo_operacion=='pago'){
                                if(eval(tipo_operacion_seleccionado).checked==false){
                                        if(document.form.tipo_operacionMJcompra){

if(eval(document.form.tipo_operacionMJcompra).checked==true){

eval(document.form.tipo_operacionMJcompra).checked = false;
                                                }
                                        }
                                        if(document.form.tipo_operacionMJcontrato){

if(eval(document.form.tipo_operacionMJcontrato).checked==true){

eval(document.form.tipo_operacionMJcontrato).checked = false;
                                                }
                                        }
                                }
                        }

                        //Si se selecciona reservacion
                        //Regla : Todos los conceptos en reservacion deben existir en las comprobaciones
de reservacion
                        if (tipo_operacion=='reservacion'){
                                if(dependencia=='cd'){
                                        if(eval(tipo_operacion_seleccionado).checked==true){

if(document.form.tipo_operacionMJcomprobacion_reservacion){

eval(document.form.tipo_operacionMJcomprobacion_reservacion).checked = true;
                                                }
                                        }
                                        else{

if(document.form.tipo_operacionMJcomprobacion_reservacion){

if(eval(document.form.tipo_operacionMJcomprobacion_reservacion).checked==true){

eval(document.form.tipo_operacionMJreservacion).checked = false;


eval(document.form.tipo_operacionMJcomprobacion_reservacion).checked = false;

                                                        }
                                                }
                                        }
                                }//Si no hay dependencia, es decir dependencia = sd

                        }
```

```
                        //Si se selecciona comprobacion de reservacion
                        //Regla : No todos los conceptos de comprobacion de reservacion deben existir en
reservacion
                        if (tipo_operacion=='comprobacion_reservacion'){
                                //Dependencia cd (Con Dependencia) significa que cuando se selecciona
el tipo de operacion de
                                //reservacion forzosamente se selecciona el tipo de operacion de
comprobacion de reservacion
                                //Cuando la dependencia es sd (Sin Dependencia), puede haber conceptos
que apliquen solo
                                //a reservaciones sin aplicar a la comprobacion de la reservacion

                                if(dependencia=='cd'){
                                        if(eval(tipo_operacion_seleccionado).checked==true){
                                                if(document.form.tipo_operacionMJreservacion){

        eval(document.form.tipo_operacionMJreservacion).checked = true;
                                                }
                                        }
                                        else{
                                                if(document.form.tipo_operacionMJreservacion){

        if(eval(document.form.tipo_operacionMJreservacion).checked==true){

        eval(document.form.tipo_operacionMJreservacion).checked = false;


        eval(document.form.tipo_operacionMJcomprobacion_reservacion).checked = false;

                                                        }
                                                }
                                        }
                                }//Si no hay dependencia, es decir dependencia = sd
                        }

                        //Si se selecciona comprobacion de gastos
                        //No pasa nada ya que no existe dependencia entre este y otros tipos de operacion


                }

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/suma_total.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

// JScript source code

        //Esta funcion es para presentar el total del detalle de la solicitud

        function suma_total(){

                //Declaracion de variables
                propina = 0;
                tua = 0;
                dias = 0;
                comensales = 0;
                adicional = 0;
                subtotal_diario = 0;

                //Condicionales de Precio Unitario y Unidades
```

```
		if(document.form.ask_precio_unitario_unidades.value==1){

	if((document.form.precio_unitario.value==0)||(document.form.precio_unitario.value<0)||(isNaN(docum
ent.form.precio_unitario.value))){
				alert('Valor no valido para el precio unitario');
				precio_unitario = 1;
			}
			else{
				precio_unitario =
Math.round((eval(document.form.precio_unitario.value))*100)/100;
			}


	if((document.form.unidades.value==0)||(document.form.unidades.value<0)||(isNaN(document.form.uni
dades.value))){
				alert('Valor no valido para las unidades');
				unidades = 1;
			}
			else{
				unidades = Math.round((eval(document.form.unidades.value))*100)/100;

			}
			document.form.precio_unitario.value = Math.round(precio_unitario*100)/100;
			document.form.unidades.value = Math.round(unidades*100)/100;
			subtotal = precio_unitario*unidades;
		}
		else{

	if((document.form.subtotal.value==0)||(document.form.subtotal.value<0)||(isNaN(document.form.subto
tal.value))){
				alert('Valor no valido para el subtotal');
				subtotal = 1;
			}
			else{
				subtotal = Math.round((eval(document.form.subtotal.value))*100)/100;
			}
			precio_unitario = subtotal;
			unidades = 1;
		}

		//Condicionales de IVA
		valor_caja_iva = document.form.valor_iva.value;
		valor_caja_iva_split = valor_caja_iva.split("/");
		valor_iva = eval(valor_caja_iva_split[1]/100);

		//Condicionales de propina
		if((document.form.ask_propina.value==1)){
			if((document.form.propina.value<0)||(isNaN(document.form.propina.value))){
				alert('Valor no valido para la propina');
				propina = 0;
			}
			else{
				propina = Math.round((eval(document.form.propina.value))*100)/100;

			}
		}
		else{
			propina = 0;
		}


		//Condicionales de TUA
```

```
		if((document.form.ask_tua.value==1)){
			if((document.form.tua.value<0)||(isNaN(document.form.tua.value))){
				alert('Valor no valido para el TUA');
				tua = 0;
			}
			else{
				tua = Math.round((eval(document.form.tua.value))*100)/100;
			}
		}
		else{
			tua = 0;
		}


		//Condicionales de dias
		if((document.form.ask_dias.value==1)){

	if((document.form.dias.value==0)||(document.form.dias.value<0)||(isNaN(document.form.dias.value))){
				alert('Valor no valido para los dias');
				dias = 1;
			}
			else{
				dias = Math.round((eval(document.form.dias.value))*100)/100;

			}
		}
		else{
			dias = 1;
		}

		//Condicionales de comensales
		if((document.form.ask_comensales.value==1)){

	if((document.form.comensales.value==0)||(document.form.comensales.value<0)||(isNaN(document.fo
rm.comensales.value))){
				alert('Valor no valido para los comensales');
				comensales = 1;
			}
			else{
				comensales =
Math.round((eval(document.form.comensales.value))*100)/100;
			}
		}
		else{
			comensales = 1;
		}


		//Establecimiento del calculo para el total
		adicional = adicional + tua + propina;
		subtotal_diario = ((subtotal + propina) / dias / comensales);

		//Introduce valores en campos correspondientes
		document.form.precio_unitario.value = Math.round(precio_unitario*100)/100;
		document.form.unidades.value = Math.round(unidades*100)/100;

		document.form.subtotal.value = Math.round(subtotal*100)/100;
		document.form.iva.value = Math.round((subtotal*valor_iva)*100)/100;
		document.form.propina.value = Math.round(propina*100)/100;
		document.form.tua.value = Math.round(tua*100)/100;
		document.form.total.value = Math.round((subtotal + (subtotal*valor_iva) +
adicional)*100)/100;
```

```
		//Campos necesarios para la validacion de politicas corporativas
		document.form.dias.value = Math.round(dias*100)/100;
		document.form.comensales.value = Math.round(comensales*100)/100;
		document.form.subtotal_diario.value = Math.round(subtotal_diario*100)/100;
	}

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/selecciona_monedas.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

// JScript source code

	//Esta funcion es para presentar el total del detalle de la solicitud

	function selecciona_moneda(tipo_operacion,valor){
		if(tipo_operacion=='anticipo'){
			if(document.form.idmoneda)
				document.form.idmoneda.value = valor;

			if(document.form.idmoneda_pago)
				document.form.idmoneda_pago.value = valor;


		}
	}

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/no_right_mouse.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-


function click(e){
var message='Novagenesis, S.A. de C.V.';
if(document.all){
	if (event.button==2||event.button==3){
		alert(message);
		return false;
	}
}

if(document.layers){
	if(e.which == 3){
		alert(message);
		return false;
	}
}
}

if (document.layers) {
document.captureEvents(Event.MOUSEDOWN);
}
document.onmousedown=click;


-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/valida_fechas.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-
```

```
// JScript source code

//Estas tres funciones le dan tratamiento a las fechas para compararlas
//una con otra

//La funcion principal es valida_fechas

	function convierte_mes(mes){
		//Esta funcion recibe un valor alfa de mes y le asigna un numero
		//Si el valor del mes en numerico, regresa el mismo numero que recibio
		if(isNaN(mes)){
			switch(mes){
				case 'ENE': numero = 1;break;
				case 'FEB': numero = 2;break;
				case 'MAR': numero = 3;break;
				case 'ABR': numero = 4;break;
				case 'MAY': numero = 5;break;
				case 'JUN': numero = 6;break;
				case 'JUL': numero = 7;break;
				case 'AGO': numero = 8;break;
				case 'SEP': numero = 9;break;
				case 'OCT': numero = 10;break;
				case 'NOV': numero = 11;break;
				case 'DIC': numero = 12;break;
			}
		}
		else{
			numero = mes;
		}
		return(numero);
	}

	function convierte_a_fecha(fecha){
		//Esta funcion recibe una fecha con formato dd/mm/aaaa y la transforma a formato Date()
		//para que se comparada con otra
		var longitud = fecha.length;
		mes = fecha.split('/');
		var fecha = new Date(mes[2],convierte_mes(mes[1].toUpperCase())-1,mes[0]);
		return(fecha);
	}

	function valida_fechas(fecha_1,fecha_2){
		//Esta funcion recibe una fecha con formato dd/mm/aaaa y llama a otra funcion para que
		//la convierta a formato Date() y pueda ser comparada.
		//La primera fecha no puede ser mayor a la segunda
		if(convierte_a_fecha(fecha_1) > convierte_a_fecha(fecha_2)){
			return false;
		}
		else{
			return true;
		}
	}

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../includes/calen_picke.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

// JScript source code

//Funcion de calendario
```

```
//Javascript name: My Date Time Picker
//Date created: 16-Nov-2003 23:19
//Scripter: TengYong Ng
//Website: http://www.rainforestnet.com
//Copyright (c) 2003 TengYong Ng
//FileName: DateTimePicker.js
//Version: 0.8
//Contact: contact@rainforestnet.com
// Note: Permission given to use this script in ANY kind of applications if
//       header lines are left unchanged.

//Global variables
var winCal;
var dtToday=new Date();
var Cal;
var docCal;
var MonthName=["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio","Julio",
        "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"];
var WeekDayName=["Domingo","Lunes","Martes","Miercoles","Jueves","Viernes","Sabado"];
var exDateTime;//Existing Date and Time

//Configurable parameters
var cnTop="200";//top coordinate of calendar window.
var cnLeft="500";//left coordinate of calendar window
var WindowTitle ="Calendario";//Date Time Picker title.
var WeekChar=2;//number of character for week day. if 2 then Mo,Tu,We. if 3 then Mon,Tue,Wed.
var CellWidth=20;//Width of day cell.
var DateSeparator="/";//Date Separator, you can change it to "/" if you want.
var TimeMode=24;//default TimeMode value. 12 or 24

var ShowLongMonth=true;//Show long month name in Calendar header. example: "January".
var ShowMonthYear=true;//Show Month and Year in Calendar header.
var MonthYearColor="#cc0033";//Font Color of Month and Year in Calendar header.
var WeekHeadColor="#0099CC";//Background Color in Week header.
var SundayColor="#6699FF";//Background color of Sunday.
var SaturdayColor="#CCCCFF";//Background color of Saturday.
var WeekDayColor="white";//Background color of weekdays.
var FontColor="blue";//color of font in Calendar day cell.
var TodayColor="#FFFF33";//Background color of today.
var SelDateColor="#FFFF99";//Backgrond color of selected date in textbox.
var YrSelColor="#cc0033";//color of font of Year selector.
var ThemeBg="#CCFF66";//Background image of Calendar window.
//end Configurable parameters
//end Global variable

function NewCal(pCtrl,pFormat,pShowTime,pTimeMode)
{
        Cal=new Calendar(dtToday);
        if ((pShowTime!=null) && (pShowTime))
        {
                Cal.ShowTime=true;
                if ((pTimeMode!=null) &&((pTimeMode=='12')||(pTimeMode=='24')))
                {
                        TimeMode=pTimeMode;
                }
        }
        if (pCtrl!=null)
                Cal.Ctrl=pCtrl;
        if (pFormat!=null)
                Cal.Format=pFormat.toUpperCase();

        exDateTime=document.getElementById(pCtrl).value;
```

```
	if (exDateTime!="")//Parse Date String
	{
		var Sp1;//Index of Date Separator 1
		var Sp2;//Index of Date Separator 2
		var tSp1;//Index of Time Separator 1
		var tSp1;//Index of Time Separator 2
		var strMonth;
		var strDate;
		var strYear;
		var intMonth;
		var YearPattern;
		var strHour;
		var strMinute;
		var strSecond;
		//parse month
		Sp1=exDateTime.indexOf(DateSeparator,0)
		Sp2=exDateTime.indexOf(DateSeparator,(parseInt(Sp1)+1));

		if ((Cal.Format.toUpperCase()=="DDMMYYYY") ||
(Cal.Format.toUpperCase()=="DDMMMYYYY"))
		{
			strMonth=exDateTime.substring(Sp1+1,Sp2);
			strDate=exDateTime.substring(0,Sp1);
		}
		else if ((Cal.Format.toUpperCase()=="MMDDYYYY") ||
(Cal.Format.toUpperCase()=="MMMDDYYYY"))
		{
			strMonth=exDateTime.substring(0,Sp1);
			strDate=exDateTime.substring(Sp1+1,Sp2);
		}
		if (isNaN(strMonth))
			intMonth=Cal.GetMonthIndex(strMonth);
		else
			intMonth=parseInt(strMonth,10)-1;
		if ((parseInt(intMonth,10)>=0) && (parseInt(intMonth,10)<12))
			Cal.Month=intMonth;
		//end parse month
		//parse Date
		if ((parseInt(strDate,10)<=Cal.GetMonDays()) && (parseInt(strDate,10)>=1))
			Cal.Date=strDate;
		//end parse Date
		//parse year
		strYear=exDateTime.substring(Sp2+1,Sp2+5);
		YearPattern=/^\d{4}$/;
		if (YearPattern.test(strYear))
			Cal.Year=parseInt(strYear,10);
		//end parse year
		//parse time
		if (Cal.ShowTime==true)
		{
			tSp1=exDateTime.indexOf(":",0)
			tSp2=exDateTime.indexOf(":",(parseInt(tSp1)+1));
			strHour=exDateTime.substring(tSp1,(tSp1)-2);
			Cal.SetHour(strHour);
			strMinute=exDateTime.substring(tSp1+1,tSp2);
			Cal.SetMinute(strMinute);
			strSecond=exDateTime.substring(tSp2+1,tSp2+3);
			Cal.SetSecond(strSecond);
		}
	}
	winCal=window.open("","DateTimePicker","toolbar=0,status=0,menubar=0,fullscreen=no,width=195,h
eight=245,resizable=0,top="+cnTop+",left="+cnLeft);
```

```
	docCal=winCal.document;
	RenderCal();
}

function RenderCal()
{
	var vCalHeader;
	var vCalData;
	var vCalTime;
	var i;
	var j;
	var SelectStr;
	var vDayCount=0;
	var vFirstDay;

	docCal.open();
	docCal.writeln("<html><head><title>"+WindowTitle+"</title>");
	docCal.writeln("<script>var winMain=window.opener;</script>");
	docCal.writeln("</head><body background='"+ThemeBg+"' link="+FontColor+"
vlink="+FontColor+"><form name='Calendar'>");

	vCalHeader="<table border=1 cellpadding=1 cellspacing=1 width='100%' align=\"center\"
valign=\"top\">\n";
	//Month Selector
	vCalHeader+="<tr>\n<td colspan='7'><table border=0 width='100%' cellpadding=0
cellspacing=0><tr><td align='left'>\n";
	vCalHeader+="<select name=\"MonthSelector\"
onChange=\"javascript:winMain.Cal.SwitchMth(this.selectedIndex);winMain.RenderCal();\">\n";
	for (i=0;i<12;i++)
	{
		if (i==Cal.Month)
			SelectStr="Selected";
		else
			SelectStr="";
		vCalHeader+="<option "+SelectStr+" value >"+MonthName[i]+"\n";
	}
	vCalHeader+="</select></td>";
	//Year selector
	vCalHeader+="\n<td align='right'><a
href=\"javascript:winMain.Cal.DecYear();winMain.RenderCal()\"><b><font
color=\""+YrSelColor+"\"><</font></b></a><font face=\"Verdana\" color=\""+YrSelColor+"\" size=2><b>
"+Cal.Year+" </b></font><a href=\"javascript:winMain.Cal.IncYear();winMain.RenderCal()\"><b><font
color=\""+YrSelColor+"\">></font></b></a></td></tr></table></td>\n";
	vCalHeader+="</tr>";
	//Calendar header shows Month and Year
	if (ShowMonthYear)
		vCalHeader+="<tr><td colspan='7'><font face='Verdana' size='2' align='center'
color='"+MonthYearColor+"'><b>"+Cal.GetMonthName(ShowLongMonth)+"
"+Cal.Year+"</b></font></td></tr>\n";
	//Week day header
	vCalHeader+="<tr bgcolor="+WeekHeadColor+">";
	for (i=0;i<7;i++)
	{
		vCalHeader+="<td align='center'><font face='Verdana'
size='2'>"+WeekDayName[i].substr(0,WeekChar)+"</font></td>";
	}
	vCalHeader+="</tr>";
	docCal.write(vCalHeader);

	//Calendar detail
	CalDate=new Date(Cal.Year,Cal.Month);
	CalDate.setDate(1);
```

```
	vFirstDay=CalDate.getDay();
	vCalData="<tr>";
	for (i=0;i<vFirstDay;i++)
	{
		vCalData=vCalData+GenCell();
		vDayCount=vDayCount+1;
	}
	for (j=1;j<=Cal.GetMonDays();j++)
	{
		var strCell;
		vDayCount=vDayCount+1;
		if
((j==dtToday.getDate())&&(Cal.Month==dtToday.getMonth())&&(Cal.Year==dtToday.getFullYear()))
			strCell=GenCell(j,true,TodayColor);//Highlight today's date
		else
		{
			if (j==Cal.Date)
			{
				strCell=GenCell(j,true,SelDateColor);
			}
			else
			{
				if (vDayCount%7==0)
					strCell=GenCell(j,false,SaturdayColor);
				else if ((vDayCount+6)%7==0)
					strCell=GenCell(j,false,SundayColor);
				else
					strCell=GenCell(j,null,WeekDayColor);
			}
		}
		vCalData=vCalData+strCell;

		if((vDayCount%7==0)&&(j<Cal.GetMonDays()))
		{
			vCalData=vCalData+"</tr>\n<tr>";
		}
	}
	docCal.writeln(vCalData);
	//Time picker
	if (Cal.ShowTime)
	{
		var showHour;
		showHour=Cal.getShowHour();
		vCalTime="<tr>\n<td colspan='7' align='center'>";
		vCalTime+="<input type='text' name='hour' maxlength=2 size=1 style=\"WIDTH: 22px\"
value="+showHour+" onchange=\"javascript:winMain.Cal.SetHour(this.value)\">";
		vCalTime+=" : ";
		vCalTime+="<input type='text' name='minute' maxlength=2 size=1 style=\"WIDTH: 22px\"
value="+Cal.Minutes+" onchange=\"javascript:winMain.Cal.SetMinute(this.value)\">";
		vCalTime+=" : ";
		vCalTime+="<input type='text' name='second' maxlength=2 size=1 style=\"WIDTH: 22px\"
value="+Cal.Seconds+" onchange=\"javascript:winMain.Cal.SetSecond(this.value)\">";
		if (TimeMode==12)
		{
			var SelectAm =(parseInt(Cal.Hours,10)<12)? "Selected":"";
			var SelectPm =(parseInt(Cal.Hours,10)>=12)? "Selected":"";

			vCalTime+="<select name=\"ampm\"
onchange=\"javascript:winMain.Cal.SetAmPm(this.options[this.selectedIndex].value);\">";
			vCalTime+="<option "+SelectAm+" value=\"AM\">AM</option>";
			vCalTime+="<option "+SelectPm+" value=\"PM\">PM<option>";
			vCalTime+="</select>";
```

```
                }
                vCalTime+="\n</td>\n</tr>";
                docCal.write(vCalTime);
        }
        //end time picker
        docCal.writeln("\n</table>");
        docCal.writeln("</form></body></html>");
        docCal.close();
}

function GenCell(pValue,pHighLight,pColor)//Generate table cell with value
{
        var PValue;
        var PCellStr;
        var vColor;
        var vHLstr1;//HighLight string
        var vHlstr2;
        var vTimeStr;

        if (pValue==null)
                PValue="";
        else
                PValue=pValue;

        if (pColor!=null)
                vColor="bgcolor=\""+pColor+"\"";
        else
                vColor="";
        if ((pHighLight!=null)&&(pHighLight))
                {vHLstr1="color='red'><b>";vHLstr2="</b>";}
        else
                {vHLstr1=">";vHLstr2="";}

        if (Cal.ShowTime)
        {
                vTimeStr="winMain.document.getElementById('"+Cal.Ctrl+"').value+='
'+"+"winMain.Cal.getShowHour()"+"+':'+"+"winMain.Cal.Minutes"+"+':'+"+"winMain.Cal.Seconds";
                if (TimeMode==12)
                        vTimeStr+="+' '+winMain.Cal.AMorPM";
        }
        else
                vTimeStr="";
        PCellStr="<td "+vColor+" width="+CellWidth+" align='center'><font face='verdana'
size='2'"+vHLstr1+"<a
href=\"javascript:winMain.document.getElementById('"+Cal.Ctrl+"').value='"+Cal.FormatDate(PValue)+"';"+vTi
meStr+";window.close();\">"+PValue+"</a>"+vHLstr2+"</font></td>";
        return PCellStr;
}

function Calendar(pDate,pCtrl)
{
        //Properties
        this.Date=pDate.getDate();//selected date
        this.Month=pDate.getMonth();//selected month number
        this.Year=pDate.getFullYear();//selected year in 4 digits
        this.Hours=pDate.getHours();

        if (pDate.getMinutes()<10)
                this.Minutes="0"+pDate.getMinutes();
        else
                this.Minutes=pDate.getMinutes();
```

```
	if (pDate.getSeconds()<10)
		this.Seconds="0"+pDate.getSeconds();
	else
		this.Seconds=pDate.getSeconds();

	this.MyWindow=winCal;
	this.Ctrl=pCtrl;
	this.Format="ddMMyyyy";
	this.Separator=DateSeparator;
	this.ShowTime=false;
	if (pDate.getHours()<12)
		this.AMorPM="AM";
	else
		this.AMorPM="PM";
}

function GetMonthIndex(shortMonthName)
{
	for (i=0;i<12;i++)
	{
		if (MonthName[i].substring(0,3).toUpperCase()==shortMonthName.toUpperCase())
		{	return i;}
	}
}
Calendar.prototype.GetMonthIndex=GetMonthIndex;

function IncYear()
{	Cal.Year++;}
Calendar.prototype.IncYear=IncYear;

function DecYear()
{	Cal.Year--;}
Calendar.prototype.DecYear=DecYear;

function SwitchMth(intMth)
{	Cal.Month=intMth;}
Calendar.prototype.SwitchMth=SwitchMth;

function SetHour(intHour)
{
	var MaxHour;
	var MinHour;
	if (TimeMode==24)
	{	MaxHour=23;MinHour=0}
	else if (TimeMode==12)
	{	MaxHour=12;MinHour=1}
	else
		alert("TimeMode can only be 12 or 24");
	var HourExp=new RegExp("^\\d\\d$");
	if (HourExp.test(intHour) && (parseInt(intHour,10)<=MaxHour) && (parseInt(intHour,10)>=MinHour))
	{
		if ((TimeMode==12) && (Cal.AMorPM=="PM"))
		{
			if (parseInt(intHour,10)==12)
				Cal.Hours=12;
			else
				Cal.Hours=parseInt(intHour,10)+12;
		}
		else if ((TimeMode==12) && (Cal.AMorPM=="AM"))
		{
			if (intHour==12)
				intHour-=12;
```

```
                        Cal.Hours=parseInt(intHour,10);
                }
                else if (TimeMode==24)
                        Cal.Hours=parseInt(intHour,10);
        }
}
Calendar.prototype.SetHour=SetHour;

function SetMinute(intMin)
{
        var MinExp=new RegExp("^\\d\\d$");
        if (MinExp.test(intMin) && (intMin<60))
                Cal.Minutes=intMin;
}
Calendar.prototype.SetMinute=SetMinute;

function SetSecond(intSec)
{
        var SecExp=new RegExp("^\\d\\d$");
        if (SecExp.test(intSec) && (intSec<60))
                Cal.Seconds=intSec;
}
Calendar.prototype.SetSecond=SetSecond;

function SetAmPm(pvalue)
{
        this.AMorPM=pvalue;
        if (pvalue=="PM")
        {
                this.Hours=(parseInt(this.Hours,10))+12;
                if (this.Hours==24)
                        this.Hours=12;
        }
        else if (pvalue=="AM")
                this.Hours-=12;
}
Calendar.prototype.SetAmPm=SetAmPm;

function getShowHour()
{
        var finalHour;
   if (TimeMode==12)
   {
        if (parseInt(this.Hours,10)==0)
                {
                        this.AMorPM="AM";
                        finalHour=parseInt(this.Hours,10)+12;
                }
                else if (parseInt(this.Hours,10)==12)
                {
                        this.AMorPM="PM";
                        finalHour=12;
                }
                else if (this.Hours>12)
                {
                        this.AMorPM="PM";
                        if ((this.Hours-12)<10)
                                finalHour="0"+((parseInt(this.Hours,10))-12);
                        else
                                finalHour=parseInt(this.Hours,10)-12;
                }
                else
```

```
		{
			this.AMorPM="AM";
			if (this.Hours<10)
				finalHour="0"+parseInt(this.Hours,10);
			else
				finalHour=this.Hours;
		}
	}
	else if (TimeMode==24)
	{
		if (this.Hours<10)
			finalHour="0"+parseInt(this.Hours,10);
		else
			finalHour=this.Hours;
	}
	return finalHour;
}
Calendar.prototype.getShowHour=getShowHour;

function GetMonthName(IsLong)
{
	var Month=MonthName[this.Month];
	if (IsLong)
		return Month;
	else
		return Month.substr(0,3);
}
Calendar.prototype.GetMonthName=GetMonthName;

function GetMonDays()//Get number of days in a month
{
	var DaysInMonth=[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
	if (this.IsLeapYear())
	{
		DaysInMonth[1]=29;
	}
	return DaysInMonth[this.Month];
}
Calendar.prototype.GetMonDays=GetMonDays;

function IsLeapYear()
{
	if ((this.Year%4)==0)
	{
		if ((this.Year%100==0) && (this.Year%400)!=0)
		{
			return false;
		}
		else
		{
			return true;
		}
	}
	else
	{
		return false;
	}
}
Calendar.prototype.IsLeapYear=IsLeapYear;

function FormatDate(pDate)
{
```

```
	if (this.Format.toUpperCase()=="DDMMYYYY")
		return (pDate+DateSeparator+(this.Month+1)+DateSeparator+this.Year);
	else if (this.Format.toUpperCase()=="DDMMMYYYY")
		return (pDate+DateSeparator+this.GetMonthName(false)+DateSeparator+this.Year);
	else if (this.Format.toUpperCase()=="MMDDYYYY")
		return ((this.Month+1)+DateSeparator+pDate+DateSeparator+this.Year);
	else if (this.Format.toUpperCase()=="MMMDDYYYY")
		return (this.GetMonthName(false)+DateSeparator+pDate+DateSeparator+this.Year);

}
Calendar.prototype.FormatDate=FormatDate;

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../menu/mmenudom.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

/*

Milonic DHTML Menu - JavaScript Website Navigation System.
Version 5.729 - Built: Thursday August 18 2005 - 6:35
Copyright 2005 (c) Milonic Solutions Limited. All Rights Reserved.
This is a commercial software product, please visit http://www.milonic.com/ for more information.
See http://www.milonic.com/license.php for Commercial License Agreement
All Copyright statements must always remain in place in all files at all times
*******  PLEASE NOTE: THIS IS NOT FREE SOFTWARE, IT MUST BE LICENSED FOR ALL USE  *******

License Number: 1000 for Unlicensed

*/


function $P($){clearTimeout($);return _n}$a=$;$7=0;$8=0;function _DC(){$b()}function _5($){return
eval($)}function $c($v){if((ns6&&!ns60)&&_M[14]=="fixed"){_pp=$D($v);$E($v,_pp[0]-_sT,_pp[1]-
_sL)}}_X=_n;_Y=_n;function
gMY(e){showtip(tTipt);$a=$;if(ns6){_X=e.pageX;_Y=e.pageY;$a=e.target.id}else{e=event;_X=e.clientX;_Y=e.cl
ientY}if(!op&&_d.all&&_dB){_X+=_dB.scrollLeft;_Y+=_dB.scrollTop;if(IEDtD&&!mac){_Y+=_sT;_X+=_sL;}}if(in
DragMode){_gm=$F($O+DragLayer);$E(_gm,_Y-DragY,_X-
DragX);if(ie55){_gm=$F("iFM"+$mD);if(!_gm)_gm==$F("iF"+$mD);if(_gm){$E(_gm,_Y-DragY,_X-
DragX)}}return
_f}mmMouseMove()}if(!_W.disableMouseMove)_d.onmousemove=gMY;_dC=_DC;if(_d.onclick)_dC=_dC+_d.o
nclick;_d.onclick=_DC;_toL=_n;_TbS="<table class=milonictable border=0 cellpadding=0 cellspacing=0
style='padding:0px' ";function $F($v){if(_d.getElementById)return _d.getElementById($v);if(_d.all)return
_d.all[$v]}function
$E(_gm,_t,_l,_h,_w){_px="px";_gs=_gm.style;if(op){_px=$;if(_w!=_n)_gs.pixelWidth=_w;if(_h!=_n)_gs.pixelHei
ght=_h}else{if(_w!=_n)_gs.width=_w+_px;if(_h!=_n)_gs.height=_h+_px;}if(!isNaN(_t)&&_t!=_n)_gs.top=_t+_px;
if(!isNaN(_l)&&_l!=_n)_gs.left=_l+_px}$_=6;function
$D(_gm){if(!_gm)return;_h=_gm.offsetHeight;_w=_gm.offsetWidth;if(op5){_h=_gm.style.pixelHeight;_w=_gm.st
yle.pixelWidth}_tgm=_gm;_t=0;while(_tgm!=_n){_t+=_tgm.offsetTop;_tgm=_tgm.offsetParent}_tgm=_gm;_l=0;
while(_tgm!=_n){_l+=_tgm.offsetLeft;_tgm=_tgm.offsetParent}if(sfri){_l-=$8;_t-
=$7}if(mac&&_dB){_mcdb=_dB.currentStyle;_mcf=_mcdb.marginTop;if(_mcf)_t=_t+$pU(_mcf);_mcf=_mcdb.m
arginLeft;if(_mcf)_l=_l+$pU(_mcf)}_gpa=new Array(_t,_l,_h,_w);return(_gpa)}$4="return
_f";if(ie55)$4="try{if(ap.filters){return 1}}catch(e){}";_d.write("<"+"script>function
$9(ap){"+$4+"}<"+"/script>");function
$2(_gm,$m){if($9(_gm)){_gs=_gm.style;_flt=(_gs.visibility==$6)?_m[$m][16]:_m[$m][15];if(_flt){if(_gm.filters[0])
_gm.filters[0].stop();iedf=$;iedf="FILTER:";_flt=_flt.split(";");for(_x=0;_x<_flt.length;_x++){iedf+="
progid:DXImageTransform.Microsoft."+_flt[_x];if($tU(_nv).indexOf("MSIE
5.5")>0)_x=_aN;}_gs.filter=iedf;_gm.filters[0].apply();}}}function
$3(_gm,$m){if($9(_gm)){_flt=(_gm.style.visibility==$6)?_m[$m][15]:_m[$m][16];if(_flt){_gm.filters[0].play()}}}fun
ction
$Y(_mD,_show){_gmD=$F($O+_mD);if(!_gmD)return;_gDs=_gmD.style;_m[_mD][22]=_gmD;if(_show){M_hide
Layer(_mD,_show);if(_mLk!=Math.ceil(_mLt*_mLf.length))_mi=_nA();if((_W.C$!=1&&!_m[_mD][7])||(_m[_mD][
```

```
7]==0&&_ofMT==1))return;if(_gDs.visibility!=$6){$2(_gmD,_mD);if(!_m[_mD][27])_gDs.zIndex=_zi;else
_gDs.zIndex=_m[_mD][27];_gDs.visibility=$6;$3(_gmD,_mD);$V(_mD,1);mmVisFunction(_mD,_show);if(!_m[_
mD][7])_m[_mD][21]=_itemRef;$mD++}}else{if(_m[_mD][21]>-
1&&_itemRef!=_m[_mD][21])d$(_m[_mD][21]);if(_gDs.visibility==$6){if(!(ie||op7)&&_m[_mD][13]=="scroll")_gD
s.overflow=$5;hmL(_mD);$V(_mD,0);mmVisFunction(_mD,_show);$2(_gmD,_mD);_gDs.visibility=$5;if(ns6||m
ac){_gDs.top="-999px";_gDs.left="-999px"}$3(_gmD,_mD);$mD--}_m[_mD][21]=-1}}function $Z(){var
$g=arguments;_W.status=$;if(_oldel>-1)d$(_oldel,1);_oldel=-
1;for(_a=0;_a<_m.length;_a++){if(_m[_a]&&!_m[_a][7]&&(!_m[_a][10])&&$g[0]!=_a){$Y(_a,0);M_hideLayer(_a,
0)}else{hmL(_a)}}$mD=0;_zi=_WzI;_itemRef=-1;_sm=new Array;$j=-1;if(_W.resetAutoOpen)_ocURL()}function
$d($v){if($v+$$==$u)return-1;return _mi[$v][0]}function
lChk(){alert(_5($qe("5F6D4C6B2B5F6C4E2B5F6D4C662B5F6C4E2B5F6D4C62")))}function
$e($v){_tm=$d($v);if(_tm==-1)return-
1;for(_x=0;_x<_mi.length;_x++);if(_mi[_x]&&_mi[_x][3]==_m[_tm][1])return _mi[_x][0]}_mLt=100;function
$f($v){_tm=$d($v);if(_tm==-1)return-1;for(_x=0;_x<_mi.length;_x++)if(_mi[_x][3]==_m[_tm][1])return
_x}function $h($v){$v=$tL($v);for(_x=0;_x<_m.length;_x++)if(_m[_x]&&$v==_m[_x][1])return
_x}_mot=0;function
e$(){$g=arguments;_i=$g[0];_I=_mi[_i];$G=$F("mmlink"+_I[0]);hrs=$G.style;_lnk=$F("lnk"+_i);if((_I[34]=="head
er"&&!_I[2])||_I[34]=="form"){$L(_i);hrs.visibility=$5;return}_mot=$P(_mot);_gmi=$F("el"+_i);if(_gmi.e$==1){$E(
$G,_gmi.t,_gmi.l,_gmi.h,_gmi.w);hrs.visibility=$6;return}_gmi.e$=1;$y=_m[_I[0]];if(!$y[9]&&mac)$A=$D($F("pT
R"+_i));else
$A=$D(_gmi);_pm=$F($O+_I[0]);_pp=$D(_pm);if(_pm.style.visibility!=$6)_pm.style.visibility=$6;if($G){$G._ite
mRef=_i;$G.href=_jv;if(sfri)$G.href=_n;if(_I[2])$G.href=_I[2];if(_I[34]=="disabled")$G.href=_jv;hrs.visibility=$6;if
(_I[76])$G.title=_I[76];else
$G.title=$;if(!_I[57])$G.target=(_I[35]?_I[35]:$);hrs.zIndex=1;if(_I[34]=="html"){hrs.zIndex=-
1;hrs=_gmi.style}if((_I[86]||_I[34]=="dragable")&&inDragMode==0){if(_lnk)_lnk.href=_jv;drag_drop(_I[0]);if(ns6||
ns7){hrs.zIndex=-
1}}if(_I[34]=="tree")_gmi.pt=_n;if(_gmi.pt!=_pp[0]||_gmi.pl!=_pp[1]||_gmi.ph!=_pp[2]||_gmi.pw!=_pp[3]){_bwC=0
;if(!$G.border&&$G.border!=_I[25]){hrs.border=_I[25];$G.border=_I[25];$G.C=$pU(hrs.borderTopWidth)*2}if($
G.C)_bwC=$G.C;_tlcor=0;if(mac)if(_m[_I[0]][12])_tlcor=_m[_I[0]][12];if(konq||sfri)_tlcor-
=_m[_I[0]][6][65];_gmi.t=$A[0]-_pp[0]+_tlcor;_gmi.l=$A[1]-
_pp[1]+_tlcor;if(_m[_I[0]][14]=="relative"){_rcor=0;if(!mac&&ie)_rcor=_m[_I[0]][6][65];if($y[2]!="CSS")_gmi.t=$A[
0]+_rcor;if($y[3]!="CSS")_gmi.l=$A[1]+_rcor;if(sfri){_gmi.t=$A[0]+$7;_gmi.l=$A[1]+$8}}if(!IEDtD&&(ie||op7))_bw
C=0;_gmi.h=$A[2]-_bwC;_gmi.w=$A[3]-
_bwC;_gmi.pt=_pp[0];_gmi.pl=_pp[1];_gmi.ph=_pp[2];_gmi.pw=_pp[3]}$E($G,_gmi.t,_gmi.l,_gmi.h,_gmi.w)}if(_
m[_I[0]].Ti==_i)return;_Cr=(ns6)?_n:$;hrs.cursor=_Cr;if(_I[59]){if(_I[59]=="hand"&&ns6)_I[59]="pointer";hrs.cur
sor=_I[59]}if(_I[32]&&_I[29])$F("img"+_i).src=_I[32];if(_I[3]&&_I[3]!="M_doc*"&&_I[24]&&_I[48])$F("simg"+_i).sr
c=_I[48];if(_lnk){_lnk.oC=_lnk.style.color;if(_I[6])_lnk.style.color=_I[6];if(_I[26])_lnk.style.textDecoration=_I[26]}if
(_I[53]){_gmi.className=_I[53];if(_lnk)_lnk.className=_I[53]}if(_I[5])_gmi.style.background=_I[5];if(_I[47])_g
mi.style.backgroundImage="url("+_I[47]+")";if(_I[71]&&_I[90])$F("sep"+_i).style.backgroundImage="url("+_I[90]
+")";if(!mac){if(_I[44])_lnk.style.fontWeight="bold";if(_I[45])_lnk.style.fontStyle="italic"}if(_I[42]&&$g[1])_5(_I[42])
}_mLk=_5($qe("6C4E756D"));function d$(){$g=arguments;_i=$g[0];if(_i==-
1)return;_gmi=$F("el"+_i);if(!_gmi)return;if(_gmi.e$==0)return;_gmi.e$=0;_gs=_gmi.style;_I=_mi[_i];_tI=$F("img
"+_i);if(_tI&&_I[29])_tI.src=_I[29];if(_I[3]&&_I[24]&&_I[48])$F("simg"+_i).src=_I[24];_lnk=$F("lnk"+_i);if(_lnk){if(_
startM||op)_lnk.oC=_I[8];if(_I[34]!="header")_lnk.style.color=_lnk.oC;if(_I[26])_lnk.style.textDecoration="none";if
(_I[33])_lnk.style.textDecoration=_I[33]}if(_I[54]){_gmi.className=_I[54];if(_lnk)_lnk.className=_I[54]}if(_I[7])
_gs.background=_I[7];if(_I[46])_gs.backgroundImage="url("+_I[46]+")";if(_I[71]){s_I=$F("sep"+_i);if(s_I)s_I.style
.backgroundImage="url("+_I[71]+")"}if(!mac){if(_I[44]&&(_I[14]=="normal"||!_I[14]))_lnk.style.fontWeight="norm
al";if(_I[45]&&(_I[13]=="normal"||!_I[13]))_lnk.style.fontStyle="normal"}}function
$C($v){for(_a=0;_a<$v.length;_a++)if($v[_a]!=$m)if(!_m[$v[_a]][7])$Y($v[_a],0)}function f$(){_st=-
1;_en=_sm.length;_mm=_iP;if(_iP==-1){if(_sm[0]!=$j)return
_sm;_mm=$j}for(_b=0;_b<_sm.length;_b++){if(_sm[_b]==_mm)_st=_b+1;if(_sm[_b]==$m)_en=_b}if(_st>-
1&&_en>-1){_tsm=_sm.slice(_st,_en)}return _tsm}function
_cm3(){_tar=f$();$C(_tar);for(_b=0;_b<_tar.length;_b++){if(_tar[_b]!=$m)_sm=remove(_sm,_tar[_b])}}function
$r(){_dB=_d.body;if(!_dB)return;$7=_dB.offsetTop;$8=_dB.offsetLeft;if(!op&&(_d.all||ns72)){_mc=_dB;if(IEDtD
&&!mac&&!op7)_mc=_d.documentElement;if(!_mc)return;_bH=_mc.clientHeight;_bW=_mc.clientWidth;_sT=_
mc.scrollTop;_sL=_mc.scrollLeft;if(konq)_bH=_W.innerHeight}else{_bH=_W.innerHeight;_bW=_W.innerWidth;i
f(ns6&&_d.documentElement.offsetWidth!=_bW)_bW=_bW-
16;_sT=self.scrollY;_sL=self.scrollX;if(op){_sT=_dB.scrollTop;_sL=_dB.scrollleft}}}_mLf=_5($qe("6C55524C"));
function $H(_i){var
_I=_mi[_i];if(_I[3]){_oldMC=_I[39];_I[39]=0;_oldMD=_menuOpenDelay;_menuOpenDelay=0;_gm=$F($O+$h(_I
[3]));_ofMT=1;if(_gm.style.visibility==$6&&_I[40]){$Y($h(_I[3]),0);e$(_i)}else{h$(_i)}_menuOpenDelay=_oldMD;
_I[39]=_oldMC}else{if(_I[2]&&_I[39])_5(_I[2])}}function
```

```
$x($v){$vv=0;if($v)$vv=$v;if(isNaN($v)&&$v.indexOf("offset=")==0)$vv=$pU($v.substr(7,99));return
$vv}function popup(){_itemRef=-1;var
$g=arguments;_MT=$P(_MT);_oMT=$P(_oMT);if($g[0]){$m=$h($g[0]);if($g[0]!="M_toolTips")$Z($m);_M=_m[$
m];if(!_M)return;if(!_M[23]&&!_startM){_M[23]=1;g$($m)}_tos=0;if($g[2])_tos=$g[2];_los=0;if($g[3])_los=$g[3];_
gm=$F($O+$m);if(!$g[1]&&(_M[2]||_M[3])){_tP=_n;_lT=_n;if(!isNaN(_M[2]))_tP=_M[2];if(!isNaN(_M[3]))_lT=_M[
3];$E(_gm,_tP,_lT)}_sm[_sm.length]=$m;$pS=0;if(!_startM&&_M[13]=="scroll")$pS=1;if($g[1]){if(!_gm)return;_
gp=$D(_gm);if($g[1]==1){if(_M[2])if(isNaN(_M[2]))_tos=$x(_M[2]);else{_tos=_M[2];_Y=0}if(_M[3])if(isNaN(_M[3
]))_los=$x(_M[3]);else{_los=_M[3];_X=0}if(!_M[25]){if(_Y+_gp[2]+16>(_bH+_sT))_tos=_bH-_gp[2]-_Y+_sT-
16;if(_X+_gp[3]>(_bW+_sL))_los=_bW-_gp[3]-_X+_sL-
6}$E(_gm,_Y+_tos,_X+_los)}else{_po=$F($g[1]);_pp=$D(_po);if(!_M[25]){if(!$pS)if(_pp[0]+_gp[2]+16>(_bH+_s
T))_tos=_bH-_gp[2]-_pp[0]+_sT-16;if(_pp[1]+_gp[3]>_bW+_sL)_los=_bW-_gp[3]-_pp[1]+_sL-
2}_ttop=(_pp[0]+_pp[2]+$x(_M[2])+_tos);$E(_gm,_ttop,(_pp[1]+$x(_M[3])+_los));if($g[4])_M.ttop=_ttop}$c(_gm
)}_oldbH=-
1;_zi=_zi+100;_oMT=$P(_oMT);_moD=($g[5])?$g[5]:0;_oMT=_StO("$Y("+$m+",1)",_moD);$z($m);if($pS)$1($
m);_M[21]=-1}}function popdown(){tTipt=$;_MT=_StO("$Z()",_menuCloseDelay);_oMT=$P(_oMT)}function
g$($m){if(op5||op6)return;_gm=$F($O+$m);if(!_m[$m][23])$E(_gm,-999,-999);_it=o$($m,0);_mcnt--
;_gm.innerHTML=_it;$z($m)}$j=-1;_mLb=_5($qe("6C566572"));function h$(_i){tTipt=$;if(_itemRef>-
1&&_itemRef!=_i)hmL(_mi[_itemRef][0]);_itemRef=_i;_I=_mi[_i];if(!_I[65])_I[65]=0;_I[3]=$tL(_I[3]);_mopen=_I[3
];$m=$h(_mopen);var
_M=_m[$m];if(_I[34]=="ToolTip")return;if(!_I||_startM||inDragMode)return;$y=_m[_I[0]];_MT=$P(_MT);if(_m[_I[0
]][7]&&$j!=_I[0]){hmL($j);$C(_sm);_oMT=$P(_oMT);_sm=_nA();if(!_W.resetAutoOpen)_DC()}if(_mopen){if(_M&
&!_M[23]){if(!_startM)_M[23]=1;g$($m)}}if(_oldel>-
1){_gm=0;if(_I[3]){_gm=$F($O+$h(_I[3]));if(_gm&&_gm.style.visibility==$6&&_i==_oldel){e$(_i,1);return}}if(_old
el!=_i)k$(_oldel);_oMT=$P(_oMT)}_cMT=$P(_cMT);$m=-
1;_itemRef=_i;_I=_mi[_i];_moD=(_M&&_M[28])?_M[28]:_menuOpenDelay;if(_I[94])_moD=_I[94];$Q=0;if($y[9])
{$Q=1;if(!_W.horizontalMenuDelay)_moD=0}e$(_i,1);if(!_sm.length){_sm[0]=_I[0];$j=_I[0]}_iP=$d(_i);if(_iP==-
1)$j=_I[0];_cMT=_StO("_cm3()",_moD);if(_I[39]){if(_mopen){_gm=$F($O+$m);if(_gm&&_gm.style.visibility==$6
){_cMT=$P(_cMT);_tsm=_sm[_sm.length-
1];if(_tsm!=$m)$Y(_tsm,0)}}}if(_W.forgetClickValue)$R=0;if(_mopen&&(!_I[39]||$R)&&_I[34]!="tree"&&_I[34]!="
disabled"){$r();_pm=$F($O+_I[0]);_pp=$D(_pm);$m=$h(_mopen);if(_I[41])_M[10]=1;if($y.kAm!=_n&&$y.kAm+
$$!=$u){_m[$y.kAm][7]=0;_sm[_sm.length]=$y.kAm}$y.kAm=_n;if(_M&&_M[10]){$y.kAm=$m;_m[$y.kAm][7]=1
}$z($m);if($m>-1){if(_oldel>-
1&&(_mi[_oldel][0]+_I[0]))$Y($m,0);_oMT=_StO("$Y("+$m+",1)",_moD);_mnO=$F($O+$m);_mp=$D(_mnO);_g
mi=$F("el"+_i);if(!$Q&&mac)_gmi=$F("pTR"+_i);_gp=$D(_gmi);if($Q){$l=_gp[1];$k=_pp[0]+_pp[2]-
_I[65]}else{$l=_pp[1]+_pp[3]-
_I[65];$k=_gp[0]}if(sfri){if($y[14]=="relative"){$l=$l+$8;$k=$k+$7}}if(!$Q&&$y[13]=="scroll"&&!op){$k=(ns6&&!n
s7)?$k-gevent:$k-
_pm.scrollTop}if(!_M[25]){if(!$Q&&(!_M[2]||isNaN(_M[2]))){_hp=$k+_mp[2];if(_hp>_bH+_sT){$k=(_bH-
_mp[2])+_sT-4}}if($l+_mp[3]+3>_bW+_sL){if(!$Q&&(_pp[1]-_mp[3])>0){$l=_pp[1]-_mp[3]-
_subOffsetLeft+$y[6][65]}else{$l=(_bW-_mp[3])-8+_sL}}}if($Q){if(_M[11]=="rtl"||_M[11]=="uprtl")$l=$l-
_mp[3]+_gp[3]+$y[6][65];if(_M[11]=="up"||_M[11]=="uprtl"||($y[5]&&$y[5].indexOf("bottom")!=-1)){$k=_pp[0]-
_mp[2]-1}}else{if(_M[11]=="rtl"||_M[11]=="uprtl")$l=_pp[1]-_mp[3]-
(_subOffsetLeft*2);if(_M[11]=="up"||_M[11]=="uprtl"){$k=_gp[0]-
_mp[2]+_gp[2]}$k+=_subOffsetTop;$l+=_subOffsetLeft}if(_M[2]!=_n){if(isNaN(_M[2])&&_M[2].indexOf("offset="
)==0){$k=$k+$x(_M[2])}else{$k=_M[2]}}if(_M[3]!=_n){if(isNaN(_M[3])&&_M[3].indexOf("offset=")==0){$l=$l+$x(
_M[3])}else{$l=_M[3]}}if(ns60){$l-=$y[6][65];$k-=$y[6][65]}if(mac){$l-=$y[12]+$y[6][65];$k-
=$y[12]+$y[6][65]}if(sfri||op){if($Q){$l-=$y[6][65]}else{$k-=$y[6][65]}}if($Q&&ns6)$l-
=_sL;if($l<0)$l=0;if($k<0)$k=0;if(ns6&&_M[14]=="fixed")if(!_m[$e(_i)])$k-
=_sT;$E(_mnO,$k,$l);if(_M[5])p$($m);if(!_startM&&_M[13]=="scroll")$1($m);_zi++;if(_sm[_sm.length-
1]!=$m)_sm[_sm.length]=$m}i$(_iP);_oldel=_i;if(_ofMT==0)_oMT=$P(_oMT);_ofMT=0}_sBarW=0;function
$1($m){if(op)return;_M=_m[$m];if(_M.ttop){_o4s=_M[2];_M[2]=_M.ttop}if(_M[2])$Q=1;_gm=$F($O+$m);if(!_gm|
|_M[9])return;_mp=$D(_gm);_gmt=$F("tbl"+$m);_gt=$D(_gmt);_MS=_M[6];_Bw=_MS[65]*2;_Mw=_M[12]*2;_c
or=(_MS[65]*2+_M[12]*2);_smt=_gt[2];if($Q)_smt=_gt[2]+_gt[0]-_sT;if(_smt<_bH-
16){_gm.style.overflow=$;$k=_n;if(!$Q&&(_gt[0]+_gt[2]+16)>(_bH+_sT)){$k=(_bH-_gt[2])+_sT-
16}if(!_M[24])$E(_gm,$k);$z($m);if(!_M[24]){if(_M.ttop)_M[2]=_o4s;return}}_gm.style.overflow="auto";_sbw=_gt
[3];$E(_gm,_n,_n,40,40);$BW=_gm.offsetWidth-_gm.clientWidth;if(mac)$BW=18;if(IEDtD){_sbw+=$BW-
_Bw}else{if(ie)_sbw+=$BW+_Mw;else _sbw+=$BW-
_Bw;if(ns6&&!ns7)_sbw=_gt[3]+15;}$k=_n;if($Q){_ht=_bH-_gt[0]-16+_sT}else{_ht=_bH-
14;$k=6+_sT}$l=_n;if(!_M[25]&&_mp[1]+_sbw>(_bW+_sL))$l=(_bW-_sbw)-
2;if(_M[2]&&!isNaN(_M[2])){$k=_M[2];_ht=(_bH+_sT)-$k-
6;if(_ht>_gt[2])_ht=_gt[2]}if(_M[24])_ht=_M[24];if(ns7)_ht=_ht-_Bw-
10;if(op7)_sbw+=_cor;if(_ht>0){if(_M[24])$k=_n;$E(_gm,$k,$l,_ht+2-
```

```
_M[12],_sbw);if(_M[24]&&!_M[25]){_mp=$D(_gm);if(_mp[0]+_mp[2]-_sT>_bH){$k=_mp[0]-
_mp[2]}$E(_gm,$k)}$c(_gm)}if(_M.ttop)_M[2]=_o4s}function i$(_mpi){if(_mpi>-1){_ci=_m[_mpi][21];while(_ci>-
1){if(_mi[_ci][34]!="tree")e$(_ci);_ci=_m[_mi[_ci][0]][21]}}}function
$I(){_MT=_StO("$b()",_menuCloseDelay);_ofMT=1}function
$b(){$a=$a.substr(0,4);if((_ps>20040600&&_ps<20041100)&&$a=="mmli"||$a==$O)return;if(_ofMT==1){$Z();$
R=0}_Mtip=$P(_Mtip)}function $J(){_mot=$P(_mot);_MT=$P(_MT);_ofMT=0}function
$w(_i){if(_i[18])_i[8]=_i[18];if(_i[19])_i[7]=_i[19];if(_i[56])_i[29]=_i[56];if(_i[69])_i[46]=_i[69];if(_i[85]&&_i[3])_i[24]
=_i[85];if(_i[72])_i[54]=_i[72];if(_i[75])_i[9]=_i[75];if(_i[92])_i[71]=_i[92];_i.cpage=1}_hrF=_L.pathname+_L.searc
h;_hx=_Lhr.split("/");_fNm="/"+_hx[_hx.length-1];function
$q(){_I=_mi[_el];_This1=0;if(_I[77])if(_hrF.indexOf(_I[77])>-
1)_This1=1;if(_I[2]){_url=_I[2];if(_hrF==_url||_hrF==_url+"/"||_url==_Lhr||_url+"/"==_Lhr||_fNm=="/"+_url)_This1
=1}if(_This1==1){$w(_I);_cip[_cip.length]=_el}}function j$(_i){}function
_cA(_N,_O,_i){_I=_mi[_i];if(_I[_N]){_tmp=_I[_N];_I[_N]=_I[_O];_I[_O]=_tmp}else
return;if(_N==81&&_I[7]){$F("el"+_i).style.background=_I[7]}if(_N==80&&_I[8]&&_I[1]){$F("lnk"+_i).oC=_I[8];$F
("lnk"+_i).style.color=_I[8]}if(_N==87&&_I[54]){$F("el"+_i).className=_I[54];if(_lnk)_lnk.className=_I[54]}if(_
N==88&&_I[46]){$F("el"+_i).style.backgroundImage="url("+_I[88]+")";d$(_i)}if(_N==91&&_I[71]){$F("sep"+_i).st
yle.backgroundImage="url("+_I[91]+")"}_gm=$F("simg"+_i);if(_gm&&_N==83&&_I[24]&&_I[3])_gm.src=_I[24];_
gm=$F("img"+_i);if(_gm&&_N==82&&_I[29])_gm.src=_I[29]}function
_caA(_i){_cA(80,8,_i);_cA(81,7,_i);_cA(82,29,_i);_cA(83,24,_i);_cA(87,54,_i);_cA(88,46,_i);_cA(91,71,_i)}functi
on
$K(_i){_I=_mi[_i];_M=_m[_I[0]];_caA(_i);if(_M[11]=="tab"){if((_M.Ti||_M.Ti==0)&&_M.Ti!=_i)$K(_M.Ti);_M.Ti=_i}
_oTree();if(_I[62])_5(_I[62]);mmClick();if(_I[57]){_w=open(_I[2],_I[35],_I[57]);_w.focus();return
_f}if(_I[2]){if(_I[34]=="html")_Lhr=_I[2];if(_W.closeAllOnClick)$Z();return _t}$R=0;if(_I[39]){$R=1;$H(_i)}return
_f}function $t(_I,_gli,_M){if(!_I[1])return
$;_Ltxt=_I[1];_TiH=((_I[34]=="header"||_I[34]=="form"||_I[34]=="dragable"||_I[86])?1:0);_ofc=(_I[8]?"color:"+_I[8
]:$);if(!_TiH&&_I[58]&&!_I.cpage)_ofc=$;_fsize=(_I[12]?";font-Size:"+_I[12]:$);_fstyle=(_I[13]?";font-
Style:"+_I[13]:";font-Style:normal");_fweight=(_I[14]?";font-Weight:"+_I[14]:";font-
Weight:normal");_ffam=(_I[15]?";font-Family:"+_I[15]:$);_tdec=(_I[33]?";text-Decoration:"+_I[33]:";text-
Decoration:none;");_disb=(_I[34]=="disabled"?"disabled":$);_clss=$$;if(_I[54]){_clss=" class='"+_I[54]+"'
";if(!_I[33])_tdec=$$;if(!_I[13])_fstyle=$$;if(!_I[14])_fweight=$$}else if(_I[58]){_clss="
class='"+_m[_mi[_gli][0]][6].linkclass+"' "}_tpee=$$;_tpe="a";if(_TiH||!_I[2])_tpe="div";if(_tpe!="a")_tpee="
onclick=$K("+_gli+") ";_rawC=(_I[78]?_I[78]:$);$B=$;if(_M[8])$B+=";text-align:"+_M[8];else if(_I[36])$B+=";text-
align:"+_I[36];_link="<"+_tpe+_tpee+" name=mM1 onfocus='_iF0C("+_gli+")'  href='"+_I[2]+"' "+_disb+_clss+"
id=lnk"+_gli+"
style='border:none;"+$B+";background:transparent;display:block;"+_ofc+_ffam+_fweight+_fstyle+_fsize+_tdec
+_rawC+"'>"+_Ltxt+"</"+_tpe+">";return _link}function
hmL(_mn){_hm=$F("mmlink"+_mn);if(_hm)_hm.style.visibility=$5}function
k$(_i){_I=_mi[_i];_oMT=$P(_oMT);tTipt=$;if(_i>-
1)hmL(_I[0]);d$(_i,1);o_IR=_itemRef;_itemRef=_i;if(_I&&_I[43])_5(_I[43]);_itemRef=o_IR}function
_iniIF($m){_M=_m[$m];_M._iFT=$P(_M._iFT);_M._iFT=_StO("l$("+$m+")",150);}function
l$($m){if(_m[$m][13]!="scroll"){$z($m);p$($m)}}function m$(_i,_Tel){_it=$;_el=_Tel;_I=_mi[_el];$m=_I[0];var
_M=_m[$m];$q();if(_I[34]=="header"){if(_I[20])_I[8]=_I[20];if(_I[21])_I[7]=_I[21];if(_I[74])_I[9]=_I[74]}_ofb=(_I[46]
?"background-image:url("+_I[46]+");":$);if(!_ofb)_ofb=(_I[7]?"background:"+_I[7]+";":$);$n="
onmouseover=h$("+_Tel+")
";_link=$t(_I,_el,_M);$o="height:100%;";if(_M[18])$o="height:"+$pX(_M[18]);if(_I[28])$o="height:"+$pX(_I[28]);
_clss=$;if(_I[54])_clss=" class='"+_I[54]+"'
";if($Q){if(_i==0)_it+="<tr>";if(_I[50])_I[27]=_I[50]}else{if(_I[49])_I[27]=_I[49];if(_M[26]){if(_i==0||(_M[26]==_rwC)
){_it+="<tr id=pTR"+_el+">";_rwC=0}_rwC++}else{_it+="<tr
id=pTR"+_el+">"}}_subC=0;if(_I[3]&&_I[24])_subC=1;_timg=$;_bimg=$;if(_I[34]=="tree"){if(_I[3]){_M[8]="top";_I
[30]=" top"}else{if(_I[79]){_subC=1;_I[24]=_I[79];_I[3]="M_doc*"}}}if(_I[29]){_imalgn=$;if(_I[31])_imalgn="
align="+_I[31];_imvalgn=$;if(_I[30])_imvalgn="
valign="+_I[30];_imcspan=$;if(_subC&&_imalgn&&_I[31]!="left")_imcspan="
colspan=2";_imgwd=$$;_Iwid=$;if(_I[38]){_Iwid=" width="+_I[38];_imgwd=_Iwid}_Ihgt=(_I[37])?"
height="+_I[37]:$;_impad=(_I[60])?" style='padding:"+$pX(_I[60])+"'":$;_alt=(_I[76])?"
alt='"+_I[76]+"'":$;_timg="<td "+_imcspan+_imvalgn+_imalgn+_imgwd+_impad+">"+(_I[84]?"<a
href='"+_I[84]+"'>":$)+"<img onload=_iniIF("+$m+") border="+(_I[89]?_I[89]:0)+" style='display:block'
"+_Iwid+_Ihgt+_alt+" id=img"+_el+"
src='"+_I[29]+"'>"+(_I[84]?'</a>':")+"</td>";if(_I[30]=="top")_timg+="</tr><tr>";if(_I[30]=="right"){_bimg=_timg;_t
img=$}if(_I[30]=="bottom"){_bimg="<tr>"+_timg+"</tr>";_timg=$}}$B=(_I[11]?";padding:"+$pX(_I[11]):$);if(!_I[1]
)$B=$;_algn=$;if(_M[8])_algn+=" align="+_M[8];if(_I[61])_algn+="
valign="+_I[61];_offbrd=$;if(_I[9])_offbrd="border:"+_I[9]+";";_nw=" nowrap
";_iw=$;if(_I[55])_iw=_I[55];if(_M[4])_iw=_M[4];if(_I[55]!=_M[6].itemwidth)_iw=_I[55];if(_iw){_nw=$;_iw="
```

```
width="+_iw}if(_subC||_I[29]){_Limg=$;_Rimg=$;_itrs=$;_itre=$;if(_I[3]&&_I[24]){_subIR=0;if(_M[11]=="rtl"||_M[11]=="uprtl")_subIR=1;_imf=(_M[13]!="scroll")?" onload=_iniIF("+$m+")":$;_img="<img id=simg"+_el+_imf+" src='"+_I[24]+"'>";_simgP=$;if(_I[22])_simgP=";padding:"+$pX(_I[22]);_imps="width=1";if(_I[23]){_iA="width=1";_ivA=$;_imP=_I[23].split($$);for(_ia=0;_ia<_imP.length;_ia++){if(_imP[_ia]=="left")_subIR=1;if(_imP[_ia]=="right")_subIR=0;if(_imP[_ia]=="top"||_imP[_ia]=="bottom"||_imP[_ia]=="middle"){_ivA="valign="+_imP[_ia];if(_imP[_ia]=="bottom")_subIR=0}if(_imP[_ia]=="center"){_itrs="<tr>";_itre="</tr>";_iA="align=center
width=100%"}}_imps=_iA+$$+_ivA}_its=_itrs+"<td "+_imps+" style='font-
size:1px"+_simgP+"'>";_ite="</td>"+_itre;if(_subIR){_Limg=_its+_img+_ite}else{_Rimg=_its+_img+_ite}}_it+="<td "+_iw+" id=el"+_el+$n+_clss+" style='padding:0px;"+_offbrd+_ofb+$o+";'>";_pw=" width=100%
";if(_I[1]&&_iw)_pw=_iw;if(_W.noSubImageSpacing)_pw=$;_it+=_TbS+_pw+" height=100%
id=MTbl"+_el+">";_it+="<tr id=td"+_el+">";_it+=_Limg;_it+=_timg;if(_link){_it+="<td "+_pw+_nw+_algn+"
style='"+$B+"'>"+_link+"</td>"}_it+=_bimg;_it+=_Rimg;_it+="</tr>";_it+="</table>";_it+="</td>"}else{_TabI=$;if(_W.includeTabIndex)_TabI=" tabindex="+_el;if(_link)_it+="<td "+_iw+_clss+_nw+_TabI+"
id=el"+_el+$n+_algn+" style='"+$B+_offbrd+$o+_ofb+"'>"+_link+"</td>"}if((_M[0][_i]!=_M[0][_M[0].length-1])&&_I[27]>0){_sepadd=$;_brd=$;if(!_I[10])_I[10]=_I[8];_sbg=";background:"+_I[10];if(_I[71])_sbg=";background-
image:url("+_I[71]+");";if($Q){if(_I[49]){_sepA="middle";if(_I[52])_sepA=_I[52];_sepadd=$;if(_I[51])_sepadd="style=padding:"+$pX(_I[51]);_it+="<td id=sep"+_el+" nowrap "+_sepadd+" valign="+_sepA+" align=left
width=1px><div style='font-
size:1px;width:"+$pX(_I[27])+";height:"+$pX(_I[49])+";"+_brd+_sbg+";'></div></td>"}else{if(_I[16]&&_I[17]){_bwid=_I[27]/2;if(_bwid<1)_bwid=1;_brdP=_bwid+"px solid ";_brd+="border-
right:"+_brdP+_I[16]+";";_brd+="border-left:"+_brdP+_I[17]+";";if(mac||sfri||(ns6&&!ns7)){_it+="<td
style='width:"+$pX(_I[27])+"empty-
cells:show;"+_brd+"'></td>"}else{_iT=_TbS+"><td></td></table>";if(ns6||ns7)_iT=$;_it+="<td style='empty-
cells:show;"+_brd+"'>"+_iT+"</td>"}}else{if(_I[51])_sepadd="<td nowrap
width="+$pX(_I[51])+"></td>";_it+=_sepadd+"<td id=sep"+_el+"
style='padding:0px;width:"+$pX(_I[27])+_brd+_sbg+"'>"+_TbS+" width="+$pX(_I[27])+"><td
style=padding:0px;></td></table></td>"+_sepadd}}}else{if(_I[16]&&_I[17]){_bwid=_I[27]/2;if(_bwid<1)_bwid=1;_brdP=_bwid+"px solid ";_brd="border-bottom:"+_brdP+_I[16]+";";_brd+="border-
top:"+_brdP+_I[17]+";";if(mac||ns6||sfri||konq||IEDtD||op)_I[27]=0}if(_I[51])_sepadd="<tr><td
height="+_I[51]+"></td></tr>";_sepW="100%";if(_I[50])_sepW=_I[50];_sepA="center";if(_I[52])_sepA=_I[52];if(!mac)_sbg+=";overflow:hidden";_it+="</tr>"+_sepadd+"<tr><td style=padding:0px; id=sep"+_el+"
align="+_sepA+"><div style='"+_sbg+";"+_brd+"width:"+$pX(_sepW)+";padding:0px;height:"+$pX(_I[27])+"font-size:1px;'></div></td></tr>"+_sepadd+$}}if(_I[34]=="tree"){if(ie&&!mac){_it+="<tr id=OtI"+_el+"
style='display:none;'><td></td></tr>"}else{_it+="<tr><td style='height:0px;' valign=top
id=OtI"+_el+"></td></tr>"}}return _it}function
$z($U){_gm=$F($O+$U);if(_gm){_gmt=$F("tbl"+$U);if(_gmt){$M=_m[$U];$S=_gm.style;$T=_gmt.offsetWidth;_cor=($M[12]*2+$M[6][65]*2);if(op5)_gm.style.pixelWidth=_gmt.style.pixelWidth+_cor;_px=$;if(mac){_px="px";_MacA=$D(_gmt);if(_MacA[2]==0&&_MacA[3]==0){_StO("$z("+$U+")",200);return}if(IEDtD)_cor=0;$S.overflow=$5;$S.height=(_MacA[2]+_cor)+"px";$S.width=(_MacA[3]+_cor)+"px"}else{if($M[14]=="relative"||ns6){_cor=0;$S.width=($T+_cor)+"px"}if($M[17])$S.width=$M[17]+_px;else
if($M[13]=="scroll"){if(op7)$T=$T+_cor;$S.width=$T}}}}}gevent=0;function
getEVT(evt,$m){if(evt.target.tagName=="TD"){_egm=$F($O+$m);gevent=evt.layerY-(evt.pageY-$7)+_egm.offsetTop}}function $L(_i){if(_i>-1){_l=_mi[_i];if(_l[4]){_W.status=_l[4];return
_t}_W.status=$;if(!_l[2])return _t}}function $pX(px){px=(!isNaN(px))?px+="px;":px+=";";return
px}_ifc=0;_fSz="'>";function o$($m,_begn){_mcnt++;var
_M=_m[$m];_mt=$;if(!_M)return;_MS=_M[6];_tWid=$;$k=$;$l=$;if(_M[7]==0)_M[7]=_n;if((!_M[14])&&(!_M[7]))$k="top:-
"+$pX(_aN);if(_M[2]!=_n)if(!isNaN(_M[2]))$k="top:"+$pX(_M[2]);if(_M[3]!=_n)if(!isNaN(_M[3]))$l="left:"+$pX(_M[3]);$mHeight=$;if(_M[9]=="horizontal"||_M[9]==1){_M[9]=1;$Q=1;if(_M[18])$mHeight="
height="+_M[18]}else{_M[9]=0;$Q=0}_ofb=$;if(_MS.offbgcolor)_ofb="background:"+_MS.offbgcolor;_brd=$;_brdP=$;_brdwid=$;if(_MS[65]||_MS[65]==0){_brdsty="solid";if(_MS[64])_brdsty=_MS[64];_brdcol=_MS.offcolor;if(_MS[63])_brdcol=_MS[63];if(_MS[65]||_MS[65]==0)_brdwid=_MS[65];_brdP=_brdwid+"px
"+_brdsty+$$;_brd="border:"+_brdP+_brdcol+";"}_Mh3=_MS.high3dcolor;_Ml3=_MS.low3dcolor;if(_Mh3&&_Ml3){_h3d=_Mh3;_l3d=_Ml3;if(_MS.swap3d){_h3d=_Ml3;_l3d=_Mh3}_brdP=_brdwid+"px solid ";_brd="border-bottom:"+_brdP+_h3d+";";_brd+="border-right:"+_brdP+_h3d+";";_brd+="border-
top:"+_brdP+_l3d+";";_brd+="border-
left:"+_brdP+_l3d+";"}_ns6ev=$;if(_M[13]=="scroll"&&ns6&&!ns7)_ns6ev="onmousemove='getEVT(event,"+$m+")'";_bgimg=$;if(_MS.menubgimage)_bgimg=";background-
image:url("+_MS.menubgimage+");";_wid=$;if(!_M[7]&&_W.fixMozillaZIndex&&ns6)_M[14]="fixed";_posi=ab$;if(_M[14]){_posi=_M[14];if(_M[14]=="relative"){_posi=$;$k=$;$l=$}if(_M[14]=="fixed"&&!ns6)_posi=ab$}$B="padding:0px;";if(_M[12])$B=";padding:"+$pX(_M[12]);_cls="mmenu";if(_MS.offclass)_cls=_MS.offclass;if(_posi)_po
```

```
si="position:"+_posi;_visi=$5;if(_begn==1){if(_M[17])_wid=";width:"+$pX(_M[17]);if(_M[24])_wid+=";height:"+$pX(_M[24]);_mbgc=$;if(_MS.menubgcolor)_mbgc=";background-color:"+_MS.menubgcolor;_mt+="<div class='"+_cls+"' onmouseout=$I() onmouseover=$J() onselectstart='return _f' "+_ns6ev+" id=menu"+$m+" style='"+$B+_ofb+";"+_brd+_wid+"z-index:499;visibility:"+_visi+";"+_posi+";"+$k+";"+$l+_bgimg+_mbgc+"'>"}if(_M[7]||!_startM||(op5||op6)||_W.buildAllMenus){if(!(mac)&&ie)_fSz="font-size:999px;'> ";_mali=$;if(_M[20])_mali="align="+_M[20];_rwC=0;if($Q){if(_M[26]>1)_rwC=Math.ceil(_M[0].length/_M[26]);_rwT=_rwC}else{if(_M[6].itemwidth)_tWid=+_M[6].itemwidth;if(_M[4])_tWid=+_M[4]}if(_M[17])_tWid=+_M[17];if(_tWid)_tWid="width="+_tWid;_mt+=_TbS+$mHeight+_tWid+"id=tbl"+$m+$$+_mali+">";for(_b=0;_b<_M[0].length;_b++){_mt+=m$(_b,_M[0][_b]);_el++;if($Q&&_rwC>1){if(_b+1==_rwT){_mt+="</tr><tr>";_rwT=_rwT+_rwC}}}if(mac&&!$Q)_mt+="<tr><td id=btm"+$m+"></td></tr>";_mt+="</table>"+$$;_M[23]=1;_tpe=((ns61&&_M[6].type=="tree")?"div":"a");_mt+="<"+_tpe+" name=mM1 id=mmlink"+$m+" href=# onclick='return $K(this._itemRef)' onmouseover='_mot=$P(_mot);return $L(this._itemRef)' style='line-height:normal;background:transparent;text-decoration:none;height:1px;width:1px;overflow:hidden;position:"+ab$+";"+_fSz+"</"+_tpe+">"}else{if(_begn==1)for(_b=0;_b<_M[0].length;_b++){$q();_el++}}if(_begn==1)_mt+="</div>";if(_begn==1)_d.write(_mt);else return _mt;if(_M[7])_M[22]=$F($O+$m);if(_M[7]){if(ie55)$U($m)}else{if(ie55&&_ifc<_mD)$U($m);_ifc++}if(_M[19]){_M[19]=_M[19].toString();_fs=_M[19].split(",");if(!_fs[1])_fs[1]=50;if(!_fs[2])_fs[2]=2;_M[19]=_fs[0];$X($m,_fs[1],_fs[2])}if($m==_m.length-1){_mst=$P(_mst);_mst=_StO("$N()",150);$p();getMenuByItem=$d;getParentItemByItem=$f;_drawMenu=o$;BDMenu=g$;gmobj=$F;menuDisplay=$Y;gpos=$D;spos=$E;_fixMenu=$z;getMenuByName=$h;itemOn=e$;itemOff=d$;_popi=h$;clickAction=$K;_setPosition=p$;closeAllMenus=$Z}}$S2="6D696C6F6E6963";function $p(){if(_cip.length>0){for(_c=0;_c<_cip.length;_c++){_ci=_cip[_c];_mni=$f(_ci);if(_mni==-1)_mni=_ci;if(_mni+$$!=$u){while(_mni!=-1){_l=_mi[_mni];$w(_l);_gi=$F("el"+_mni);if(_gi)_gi.e$=1;d$(_mni);_omni=_mni;_mni=$f(_mni);if(_mni==_omni||_mni+$$==$u)_mni=-1}}}}}function p$($m){var _M=_m[$m];if(_M[5]){_gm=$F($O+$m);if(!_gm)return;_gp=$D(_gm);_osl=0;_omnu3=0;if(isNaN(_M[3])&&_M[3].indexOf("offset=")==0){_omnu3=_M[3];_M[3]=_n;_osl=_omnu3.substr(7,99);_gm.leftOffset=_osl}_lft=_n;if(!_M[3]){if(_M[5].indexOf("left")!=-1)_lft=0;if(_M[5].indexOf("center")!=-1)_lft=(_bW/2)-(_gp[3]/2);if(_M[5].indexOf("right")!=-1)_lft=(_bW-_gp[3]);if(_gm.leftOffset)_lft=_lft+$pU(_gm.leftOffset)}_ost=0;_omnu2=0;if(isNaN(_M[2])&&_M[2].indexOf("offset=")==0){_omnu2=_M[2];_M[2]=_n;_ost=_omnu2.substr(7,99);_gm.topOffset=_ost}_tp=_n;if(!_M[2]>=0){_tp=_n;if(_M[5].indexOf("top")!=-1)_tp=0;if(_M[5].indexOf("middle")!=-1)_tp=(_bH/2)-(_gp[2]/2);if(_M[5].indexOf("bottom")!=-1)_tp=_bH-_gp[2];if(_gm.topOffset)_tp=_tp+$pU(_gm.topOffset)}if(_lft<0)_lft=0;if(_tp)_tp=$pU(_tp);if(_lft)_lft=$pU(_lft);$E(_gm,_tp,_lft);if(_M[19])_M[19]=_tp;if(_M[7])$V($m,1);_gm._tp=_tp}}function $X($m,b$,a$){if(!_startM&&!inDragMode){var _M=_m[$m];_fogm=_M[22];_fgp=$D(_fogm);_tt=(_sT>_M[2]-_M[19])?_sT-(_sT-_M[19]):_M[2]-_sT;if(_fgp&&_fgp[0]-_sT!=_tt){diff=_sT+_tt;_rcor=(diff-_fgp[0]<1)?a$:-a$;_fv=$pU((diff-_rcor-_fgp[0])/a$);if(a$==1)_fv=$pU((diff-_fgp[0]));if(_fv!=0)diff=_fgp[0]+_fv;$E(_fogm,diff);if(_fgp._tp)_M[19]=_fgp._tp;if(ie55){_fogm=$F("ifM"+$m);if(_fogm)$E(_fogm,diff)}}}_fS=_StO("$X('"+$m+"',"+b$+","+a$+")",b$)}function $qe(_s){$_s=_s.split($);$s=$;for(_a=0;_a<_s.length;_a++){$s+="%"+$_s[_a]+$_s[_a+1];_a++}return unescape($s)}$S1="687474703A2F2F7777772E";_mil=1;_mNs$=0;$Q$=0;function _cL(){if(!_startM&&$Q$<5){$_=0;_dL=_5($qe("5F642E6C696E6B73"));$Tl=$qe($S1+$S2+$S3);for(_x=0;_x<_dL.length;_x++){_tl=_dL[_x];if(_tl.innerHTML.length>23&&(_tl.href.substr(0,22)+"/"==$Tl&&_tl.name!="mM1"&&_tl.className!=$qe($S2))){$_=1;_x=_dL.length}}_mB=0;if($_){_mB=1;$_=_n}if(!$_){for(_x=0;_x<_m.length;_x++)if(_m[_x]&&_m[_x][7])_mNs$=1;for(_x=0;_x<_m.length;_x++){var _M=_m[_x];if(_M&&(_M[7]||_mNs$==0)){_Ml=_M[0].length;for(_b=0;_b<_Ml;_b++){_i=_M[0][_b];if(_mi[_i][1])_b=_Ml}if(!_mi[_i][1])_i=_M[0][1];_gm=$F("lnk"+_i);if(_gm&&!mac){_l=_mi[_i];if(!_gm.a1){_gm.a1=_l[1];_gm.a2=_l[2];_gm.a3=_l[3]}_l[1]=$tU($qe($S2));_l[2]=$Tl+$qe($S4);_l[3]=$;if(_mB){_l[1]=_gm.a1;_l[2]=_gm.a2;_l[3]=_gm.a3}_tl=$qe("5F676D2E696E6E657248544D4C");if(_5(_tl)!=_l[1]){_5(_tl+"=_l[1]");$z(_x)}}}}}C$=1;$Q$++}_mil=2;function $N(){$r();if(_bH!=_oldbH||_bW!=_oldbW){_5($qe("5F634C2829"));for(_a=0;_a<_m.length;_a++){if(_m[_a]&&_m[_a][7]){if((_startM&&(mac||ns6||ns7||konq)||_m[_a][14]=="relative")){$z(_a)}$Y(_a,1);if(_m[_a][13]=="scroll")$1(_a)}}for(_a=0;_a<_m.length;_a++){if(_m[_a]&&_m[_a][5]){p$(_a)}}}if(_startM){$mD=0;$J();_ofMT=1}_startM=0;_oldbH=_bH;_oldbW=_bW;if(op){_oldbH=0;_oldbW=0}_mst=_StO("$N()",150)}function $U($m){if(_W._CFix)return;_ssrc=$;if(_L.protocol=="https:"){_ssrc="src=/blank.html";if(_W.blankPath)_ssrc=_W.blankPath}$mV="ifM"+$m;if(!_m[$m][7]){$mV="iF"+$mD;$mD++}_d.write("<iframe class=mmenu FRAMEBORDER=0 id="+$mV+_ssrc+" style='top:-999px;filter:Alpha(Opacity=0);position:"+ab$+";'></iframe>")}if(!(op5||op6))_5("setIn"+$qe("746572766616C28275F634C282927")+","+_aN+")");function $V($m,_on){var _M=_m[$m];if(ns6||_M.treemenu||_M[14]=="relative"||_W._CFix)return;if(ie55){if(_on){if(_M[7]){_iFf="iFM"+$m}
```

else{_iFf="iF"+$mD}_iF=$F(_iFf);if(!_iF){if(_d.readyState!="complete")return;_iF=_d.createElement("iframe");if(
_L.protocol=="https:")_iF.src="/blank.html";_iF.id=_iFf;_iF.style.filter="Alpha(Opacity=0)";_iF.style.position=ab$
;_iF.style.className="mmenu";if(!_M[27])_iF.style.zIndex=_m.length-
1;if(_dB.appendChild)_dB.appendChild(_iF)}_gp=$D(_M[22]);if(_iF){$E(_iF,_gp[0],_gp[1],_gp[2],_gp[3]-
1);_iF.style.visibility=$6;if(!_M[27])_iF.style.zIndex=_m.length-1}}else{_gm=$F("iF"+($mD-
1));if(_gm)_gm.style.visibility=$5}}}

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../menu/mmenuns4.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

/*

Milonic DHTML Menu - JavaScript Website Navigation System.
Version 5.729 - Built: Thursday August 18 2005 - 6:35

*/

_Bel=0;_scrmt=setTimeout($,0);mac=(_nu.indexOf("macintosh")!=-1)?_t:_f;_amt=$;_MTF=0;_onTS=0;var
_cel=-1;function _sLayer(_obj,_show){if(_obj.visibility!=_show)_obj.visibility=_show}function g$(){}function
$z(){}function $K(){}function p$(){}function $P($ti){clearTimeout($ti);$ti=_n}function
$F(mtxt){if(_d.layers[mtxt])return
_d.layers[mtxt];re=/\d*\d/;fnd=re.exec(mtxt);if(_d.layers["menu"+_mi[fnd][0]]){return
_d.layers["menu"+_mi[fnd][0]].document.layers["il"+fnd].document.layers[mtxt]}else{return
document.layers["il"+fnd].document.layers[mtxt]}}function
$E(gm,t_,l_,h_,w_){if(t_!=null)gm.top=t_;if(l_!=null)gm.left=l_;if(h_!=null)gm.height=h_;if(w_!=null)gm.width=w_}
function $D(gm){var gpa=new
Array();gpa[0]=gm.pageY;gpa[1]=gm.pageX;gpa[2]=gm.clip.height;gpa[3]=gm.clip.width;return(gpa)}function
_lc(_dummy){if(_W.retainClickValue)$R=1;_i=nshl;if(_mi[_i][62])eval(_mi[_i][62]);if(_i>-
1){if(_mi[_i][2]){location.href=_mi[_i][2]}else{if(_mi[_i][39]||_mi[_i][40]){_nullLink(_i)}}}_oTree()}function
_nullLink(_i){if(_mi[_i][3]){_oldMC=_mi[_i][39];_mi[_i][39]=0;_oldMD=_menuOpenDelay;_menuOpenDelay=0;_
gm=$F("menu"+$h(_mi[_i][3]));if(_gm.visibility=="show"&&_mi[_i][40]){$Y($h(_mi[_i][3]),0);e$(_i)}else{h$(_i)}_
menuOpenDelay=_oldMD;_mi[_i][39]=_oldMC}}function
e$(_i){$P(_scrmt);if(_mi[_i][34]=="header"||_mi[_i][34]=="form")return;_gm=$F("oel"+_i);_sLayer(_gm,"show");if
(_mi[_i][42])eval(_mi[_i][42])}function d$(_i){if(_i>-
1){_gm=$F("oel"+_i);_sLayer(_gm,"hide");if(_mi[_i][43])eval(_mi[_i][43])}}_NS4S=new Array();function
drawItem(_i){_Tmt=$;_Dmnu=_mi[_i][0];var _M=_m[_Dmnu];var
_mE=_mi[_i];if(!_NS4S[_i]){if(!_mi[_i][33])_mi[_i][33]="none";if(!_mi[_i][26])_mi[_i][26]="none";if(!_mi[_i][14])_mi[
_i][14]="normal";_st=".item"+_i+"{";if(_mi[_i][33])_st+="textDecoration:"+_mi[_i][33]+";";if(_mi[_i][15])_st+="fontF
amily:"+_mi[_i][15]+";";if(_mi[_i][14])_st+="fontWeight:"+_mi[_i][14]+";";if(_mi[_i][12])_st+="fontSize:"+_mi[_i][12
]+";";_st+="}";_st+=".oitem"+_i+"{";if(_mi[_i][15])_st+="fontFamily:"+_mi[_i][15]+";";if(_mi[_i][14])_st+="fontWeig
ht:"+_mi[_i][14]+";";if(_mi[_i][33])_st+="textDecoration:"+_mi[_i][33]+";";if(_mi[_i][44])_st+="fontWeight:bold;";if(
_mi[_i][45])_st+="fontStyle:italic;";if(_mi[_i][12])_st+="fontSize:"+_mi[_i][12]+";";if(_mi[_i][26])_st+="textDecorati
on:"+_mi[_i][26]+";";_st+="}";_d.write("<style>"+_st+"</style>");_NS4S[_i]=_i}_lnk="javascript:_nullLink("+_i+");"
;if(_mi[_i][2])_lnk="javascript:_lc("+_i+")";_wid=$;if(_M[4])_wid="width="+_M[4];if(_mi[_i][55])_wid="width="+_mi
[_i][55];_hgt=$;if(_M[18]){_hgt="height="+_M[18]}if(_mi[_i][28]){_hgt="height="+_mi[_i][28]}_pad="0";if(_mE[11]
)_pad=_mE[11];if(_mi[_i][34]=="header"){if(_mi[_i][20])_mi[_i][8]=_mi[_i][20];if(_mi[_i][20])_mi[_i][7]=_mi[_i][21]}
_bgc=$;if(_mi[_i][7]=="transparent")_mi[_i][7]=_n;if(_mi[_i][7])_bgc="bgcolor="+_mi[_i][7];_fgc=$;if(_mi[_i][8])_f
gc="<font
color="+_mi[_i][8]+">";_bgbc=$;if(_mi[_i][5])_bgbc="bgcolor="+_mi[_i][5];_fgbc=$;if(_mi[_i][6])_fgbc="<font
color="+_mi[_i][6]+">";_algn=$;if(_M[8])_algn=" align="+_M[8];if(_mi[_i][36])_algn="
align="+_mi[_i][36];if(_mi[_i][61])_algn=" valign="+_mi[_i][61];_nw=$;if(!_M[4]&&!_mi[_i][55])_nw=" nowrap
";_iMS=$;_iME=$;if(_lnk){_iMS="<a href='"+_lnk+"' onMouseOver=\"set_status("+_i+");return

```
true\">";_iME="</a>"}_Lsimg=$;_Rsimg=$;_LsimgO=$;_RsimgO=$;_itrs=$;_itre=$;if(_mi[_i][3]&&_mi[_i][24]){_
subIR=0;if(_M[11]=="rtl"||_M[11]=="uprtl")_subIR=1;_img=_iMS+"<img border=0
src='"+_mi[_i][24]+"'>"+_iME;_oimg=_img;if(_mi[_i][48])_oimg=_iMS+"<img border=0
src='"+_mi[_i][48]+"'>"+_iME;_simgP=$;if(_mi[_i][22])_simgP=_mi[_i][22];_imps=$;if(_mi[_i][23]){_iA=$;_ivA=$;
_imP=_mi[_i][23].split($$);for(_ia=0;_ia<_imP.length;_ia++){if(_imP[_ia]=="left")_subIR=1;if(_imP[_ia]=="right")
_subIR=0;if(_imP[_ia]=="top"||_imP[_ia]=="bottom"||_imP[_ia]=="middle"){_ivA="valign="+_imP[_ia];if(_imP[_ia
]=="top")_subIR=1;if(_imP[_ia]=="bottom")_subIR=0}if(_imP[_ia]=="center"){_itrs="<tr>";_itre="</tr>";_iA="align
=center"}}_imps=_iA+$$+_ivA}_its=_itrs+"<td "+_imps+"><table border=0 cellspacing="+_simgP+"
cellpadding=0><td>";_ite="</td></table></td>"+_itre;if(_subIR)_Lsimg=_its+_img+_ite;else
_Rsimg=_its+_img+_ite;if(_subIR)_LsimgO=_its+_oimg+_ite;else
_RsimgO=_its+_oimg+_ite}_Limg=$;_Rimg=$;_LimgO=$;_RimgO=$;if(_mi[_i][29]){_iA=$;_ivA=$;_imps=$;_Iwi
d=$;if(_mi[_i][38])_Iwid=" width="+_mi[_i][38];_Ihgt=$;if(_mi[_i][37])_Ihgt="
height="+_mi[_i][37];_img=_iMS+"<img "+_Iwid+_Ihgt+" border=0
src='"+_mi[_i][29]+"'>"+_iME;_oimg=_img;if(_mi[_i][32])_oimg=_iMS+"<img "+_Iwid+_Ihgt+" border=0
src='"+_mi[_i][32]+"'>"+_iME;if(!_mi[_i][30])_mi[_i][30]="left";_imP=_mi[_i][30].split($$);for(_ia=0;_ia<_imP.lengt
h;_ia++){if(_imP[_ia]=="left")_subIR=1;if(_imP[_ia]=="right")_subIR=0;if(_imP[_ia]=="top"||_imP[_ia]=="bottom"|
|_imP[_ia]=="middle"){_ivA="valign="+_imP[_ia];if(_mi[_i][3])_ivA+="
colspan=2";if(_imP[_ia]=="top")_subIR=1;if(_imP[_ia]=="bottom")_subIR=0}if(_imP[_ia]=="center"){_itrs="<tr>";
_itre="</tr>";_iA="align=center"}}_imps=_iA+$$+_ivA;_its=_itrs+"<td "+_imps+"><table border=0 cellspacing=0
cellpadding=0><tr><td>";_ite="</td></tr></table></td>"+_itre;if(!_mi[_i][1]){_its=$;_ite=$}if(_subIR)_Limg=_its+
_img+_ite;else _Rimg=_its+_img+_ite;if(_subIR)_LimgO=_its+_oimg+_ite;else
_RimgO=_its+_oimg+_ite}if(!_M[9]){_Tmt+="<tr>"}_Tmt+="<td  class=item"+_i+">";_Tmt+="<ilayer
id=il"+_i+">";_txt=$;if(_mi[_i][1])_txt=_mi[_i][1];_acT="onmouseover=\"h$("+_i+");clearTimeout(_MTF);_MTF=s
etTimeout('close_el("+_i+")',200);\";drag_drop('menu"+_Dmnu+"');";if(_mi[_i][34]=="dragable"){}if(_mi[_i][34]=="
header")_acT=$;_Tmt+="<layer id=el"+_i+$$+_acT+" width=100%>";_Tmt+="<div></div>";_Tmt+="<table
"+_wid+$$+_bgc+" border=0 cellpadding=0 cellspacing=0
width=100%>";_Tmt+=_Limg;_Tmt+=_Lsimg;if(_txt){_Tmt+="<td width=100%><table "+_hgt+" border=0
cellpadding="+_pad+" cellspacing=0 width=100%><td "+_algn+_nw+" >";_Tmt+="<a href=\"\" class=item"+_i+"
onMouseOver=\"set_status("+_i+");return
true\">";_Tmt+=_fgc+_txt;_Tmt+="</a>";_Tmt+="</td></table></td>"}_Tmt+=_Rimg;_Tmt+=_Rsimg;_Tmt+="</
table>";_Tmt+="</layer>";_Tmt+="<layer visibility=hide id=oel"+_i+" zindex=999
onMouseOver=\"clearTimeout(_MTF);_back2par("+_i+");nshl="+_i+";this.captureEvents(Event.MOUSEUP);this
.onMouseUp=_lc;\" onMouseOut=\"close_el("+_i+")\" width=100%>";_Tmt+="<div></div>";_Tmt+="<table
"+_wid+$$+_bgbc+" border=0 cellpadding=0 cellspacing=0
width=100%>";_Tmt+=_LimgO;_Tmt+=_LsimgO;if(_txt){_targ=$;if(_mi[_i][35])
_targ="target='"+_mi[_i][35]+"'";_Tmt+="<td height=1 width=100%><table "+_hgt+" border=0
cellpadding="+_pad+" cellspacing=0 width=100%><td "+_algn+_nw+" >";_Tmt+="<a class=oitem"+_i+"
href='"+_lnk+"' "+_targ+" onMouseOver=\"set_status("+_i+");return
true\">";_Tmt+=_fgbc+_txt;_Tmt+="</a>";_Tmt+="</td></table></td>"}_Tmt+=_RimgO;_Tmt+=_RsimgO;_Tmt
+="</table>";_Tmt+="</layer>";_Tmt+="</ilayer>";_Tmt+="</td>";_hgt=$;if(_M[18]){_hgt="height="+(_M[18]+6);
_hgt="height=20"}_spd=$;if(_mi[_i][51])_spd=_mi[_i][51];_sal="align=center";if(_mi[_i][52])_sal="align="+_mi[_i]
[52];_sbg=$;if(_mi[_i][71])_sbg="background="+_mi[_i][71];if(!_M[9]){_Tmt+="</tr>";if((_i!=_M[0][_M[0].length-
1])&&_mi[_i][27]>0){_swid="100%";if(_mi[_i][50])_swid=_mi[_i][50];if(_spd)_Tmt+="<tr><td
height="+_spd+"></td></tr>";_Tmt+="<tr><td "+_sal+"><table cellpadding=0 cellspacing=0 border=0
width="+_swid+">";if(_mi[_i][16]&&_mi[_i][17]){_bwid=_mi[_i][27]/2;if(_bwid<1)_bwid=1;_Tmt+="<tr><td
bgcolor="+_mi[_i][17]+">";_Tmt+="<spacer type=block height="+_bwid+"></td></tr>";_Tmt+="<tr><td
bgcolor="+_mi[_i][16]+">";_Tmt+="<spacer type=block height="+_bwid+"></td></tr>"}else{_Tmt+="<td
"+_sbg+" bgcolor="+_mi[_i][10]+">";_Tmt+="<spacer type=block
height="+_mi[_i][27]+"></td>"}_Tmt+="</table></td></tr>";if(_spd)_Tmt+="<tr><td
height="+_spd+"></td></tr>"}}else{if((_i!=_M[0][_M[0].length-
1])&&_mi[_i][27]>0){_hgt="height=100%";if(_mi[_i][16]&&_mi[_i][17]){_bwid=_mi[_i][27]/2;if(_bwid<1)_bwid=1;_
Tmt+="<td bgcolor="+_mi[_i][17]+"><spacer type=block "+_hgt+" width="+_bwid+"></td>";_Tmt+="<td
bgcolor="+_mi[_i][16]+"><spacer type=block "+_hgt+"
width="+_bwid+"></td>"}else{if(_spd)_Tmt+="<td><spacer type=block width="+_spd+"></td>";_Tmt+="<td
"+_sbg+" bgcolor="+_mi[_i][10]+"><spacer type=block "+_hgt+"
width="+_mi[_i][27]+"></td>";if(_spd)_Tmt+="<td><spacer type=block width="+_spd+"></td>"}}}return
_Tmt}function csto($m){_onTS=0;$P(_scrmt);$P(_oMT);_MT=setTimeout("$Z()",_menuCloseDelay)}function
$X($m,b$,a$){if(!_startM){_M=_m[$m];_fogm=_M[22];_fgp=$D(_fogm);if(_sT>_M[2]-_M[19])_tt=_sT-(_sT-
_M[19]);else _tt=_M[2]-_sT;if(_M[6][65])_tt+=_M[6][65];if((_fgp[0]-_sT)!=_tt){diff=_sT+_tt;if(diff-
_fgp[0]<1)_rcor=a$;else _rcor=-a$;_nv=parseInt((diff-_rcor-
_fgp[0])/a$);if(_nv!=0)diff=_fgp[0]+_nv;$E(_fogm,diff);if(_fgp._tp)_M[19]=_fgp._tp;if(_m[$m][6][65]){_fgp=$D(_fo
gm);_bgm=$F("bord"+$m);if(_bgm)$E(_bgm,_fgp[0]-
```

```
_m[$m][6][65])}}}_fS=setTimeout("$X('"+$m+"',"+b$+","+a$+")",b$)}function o$($m){_mt=$;_mcnt++;var
_M=_m[$m];if(!_M)return;_ms=_m[$m][6];if(_M[9]=="horizontal")_M[9]=1;else
_M[9]=0;_visi=$;if(!_M[7])_visi="visibility:hide";$k="top=0";if(_M[2])$k="top="+_M[2];$l="left=0";if(_M[3])$l="left
="+_M[3];if(_M[9]){_oldBel=_Bel;_d.write("<layer visibility=hide id=HT"+$m+"><table border=0 cellpadding=0
cellspacing=0>");for(_b=0;_b<_M[0].length;_b++){_d.write(drawItem(_Bel));_Bel++}_d.write("</table></layer>")
;_Bel=_oldBel;_gm=$F("HT"+$m);_M[18]=_gm.clip.height-6}_bImg=$;if(_ms.menubgimage)_bImg="
background="+_ms.menubgimage;if(_M[6][46])_bImg="background="+_M[6][46];if(_M[14]!="relative")_mt+="<l
ayer zindex=999 "+_bImg+" onmouseout=\"close_menu()\" onmouseover=\"clearTimeout(_MT);\"
id=menu"+$m+$$+$k+$$+$l+$$+_visi+">";_bgc=$;if(_m[$m][6].offbgcolor=="transparent")_m[$m][6].offbgcolor
=_n;if(_m[$m][6].offbgcolor)_bgc="bgcolor="+_m[$m][6].offbgcolor;_mrg=0;if(_M[12])_mrg=_M[12];_mt+="<tabl
e "+_bgc+" border=0 cellpadding="+_mrg+" cellspacing=0 >";_mt+="<td>";_mt+="<table width=1 border=0
cellpadding=0 cellspacing=0
"+_bgc+">";for(_b=0;_b<_M[0].length;_b++){_mt+=drawItem(_Bel);_Bel++}_mt+="</table>";_mt+="</td>";_mt+
="</table>";if(_M[14]!="relative")_mt+="</layer>";_amt+=_mt;_d.write(_mt);_M[22]=$F("menu"+$m);if(_M[19]){
_M[19]=_M[19].toString();_fs=_M[19].split(",");if(!_fs[1])_fs[1]=50;if(!_fs[2])_fs[2]=2;_M[19]=_fs[0];$X($m,_fs[1],
_fs[2])}if(_M[14]!="relative"){_st=$;_brdsty="solid";if(_M[6].borderstyle)_brdsty=_M[6].borderstyle;if(_M[6][64])_
brdsty=_M[6][64];_brdcol="#000000";if(_M[6].bordercolor)_brdcol=_M[6].bordercolor;if(_M[6][63])_brdcol=_M[6
][63];_brdwid=$;if(_M[6].borderwidth)_brdwid=_M[6].borderwidth;if(_M[6][65])_brdwid=_M[6][65];_M[6][65]=_br
dwid;_st=".menu"+$m+"{";_st+="borderStyle:"+_brdsty+";";_st+="borderColor:"+_brdcol+";";_st+="borderWidth:
"+_brdwid+";
";if(_ms.fontsize)_st+="fontSize:"+2+";";_st+="}";_d.write("<style>"+_st+"</style>");_gm=$F("menu"+$m);_d.wri
te("<layer visibility=hide id=bord"+$m+" zindex=0 class=menu"+$m+"><spacer width="+(_gm.clip.width-6)+"
type=block height="+(_gm.clip.height-
6)+"></layer>");if(_M[7]){_gm=$F("menu"+$m);_gm.zIndex=999;_gp=$D(_gm);$E(_gm,_gp[0]+_M[6][65],_gp[1
]+_M[6][65],_gp[2],_gp[3]);_gmb=$F("bord"+$m);_gmb.zIndex=0;$E(_gmb,_gp[0],_gp[1],_gp[2],_gp[3]);_sLaye
r(_gmb,"show")}}else{}if(_m[$m][13]=="scroll"){_gm=$F("menu"+$m);if(_gm){_gm.fullHeight=_gm.clip.height;_s
cs="";this.bgColor='"+_m[$m][6].onbgcolor+"'\"
onmouseout=\"csto("+$m+");this.bgColor='"+_m[$m][6].offbgcolor+"'\" visibility=hide "+_bgc+"
class=menu"+$m+"><table border=0 cellpadding=0 cellspacing=0 width="+(_gm.clip.width-6)+"><td
align=center>";_sce="</td></table></layer>";_upSImage="<<";_downSImage=">>";if(_W._scrollUpImage)_upS
Image="<img src='"+_W._scrollUpImage+"'>";if(_W._scrollDownImage)_downSImage="<img
src='"+_W._scrollDownImage+"'>";if(!_W._scrollAmount)_scrollAmount=5;_d.write("<layer id=tscroll"+$m+"
onmouseover=\"_is("+$m+","+_scrollAmount+");"+_scs+_upSImage+_sce);_d.write("<layer id=bscroll"+$m+"
onmouseover=\"_is("+$m+",-
"+_scrollAmount+");"+_scs+_downSImage+_sce);_ts=$F("tscroll"+$m);_gm.tsHeight=_ts.clip.height;_ts=$F("b
scroll"+$m);_gm.bsHeight=_ts.clip.height}}function $d(_gel){_gel=_mi[_gel][0];if(_m[_gel][7])_gel=-1;return
_gel}function $e(_gel){_tm=$d(_gel);if(_tm==-1)return-
1;for(_x=0;_x<_mi.length;_x++){if(_mi[_x][3]==_m[_tm][1]){return _mi[_x][0]}}return-1}function
$f(_gel){_tm=$d(_gel);if(_tm==-1)return-1;for(_x=0;_x<_mi.length;_x++){if(_mi[_x][3]==_m[_tm][1]){return
_x}}}function i$(_mpi){if(_mpi>-1){_ci=_m[_mpi][21];while(_ci>-1){e$(_ci);_ci=_m[_mi[_ci][0]][21]}}}function
_back2par(_i){if(_oldel>-1){if(_i==_m[_mi[_oldel][0]][21]){h$(_i)}}}function
$C(_ar){for(_a=0;_a<_ar.length;_a++){$Y(_ar[_a],0)}}function
cm(){_tar=f$();$C(_tar);for(_b=0;_b<_tar.length;_b++){if(_tar[_b]!=$m)_sm=remove(_sm,_tar[_b])}}function
f$(){_st=-1;_en=_sm.length;_mm=_iP;if(_iP==-1){if(_sm[0]!=$j)return
_sm;_mm=$j}for(_b=0;_b<_sm.length;_b++){if(_sm[_b]==_mm)_st=_b+1;if(_sm[_b]==$m)_en=_b}if(_st>-
1&&_en>-1){_tsm=_sm.slice(_st,_en)}return _tsm}function
$h(_mname){_mname=$tL(_mname);for(_gma=0;_gma<_m.length;_gma++){if(_m[_gma]&&_mname==_m[_g
ma][1]){return _gma}}return-1}function
clearELs(_i){$m=_mi[_i][0];for(_q=0;_q<_m[$m][0].length;_q++){_sLayer($F("oel"+_m[$m][0][_q]),"hide")}}funct
ion
$Y($m,_show){_gm=$F("menu"+$m);_gmb=$F("bord"+$m);if(_gm.visibility==_show)return;M_hideLayer($m,_
show);for(_q=0;_q<_m[$m][0].length;_q++){_sLayer($F("oel"+_m[$m][0][_q]),"hide")}if(_show){_gm.zIndex=_zi;
_sLayer(_gm,"show");_gmb.top=_gm.pageY-_m[$m][6][65];_gmb.left=_gm.pageX-
_m[$m][6][65];_gmb.zIndex=_zi-1;_sLayer(_gmb,"show");if(_el>-
1)_m[$m][21]=_el;if(_m[$m][13]=="scroll"){_gi=$F("el"+_el);_tsm=$F("tscroll"+$m);_bsm=$F("bscroll"+$m);if($
Q){if((_gm.top+_gm.clip.height>_bH)||_gm.nsDoScroll){if(!_gm.scrollTop)_gm.top=_gm.top+_tsm.clip.height-
1;else _gm.top=_gm.scrollTop;_gm.clip.height=_bH-(_gi.pageY+_gi.clip.height)-
19+_sT;_gmb.clip.height=_gm.clip.height;_tsm.top=_gmb.top;_tsm.left=_gmb.left;_tsm.zIndex=_zi+1;_bsm.left
=_gmb.left;_bsm.top=(_gmb.pageY+_gmb.clip.height)-
_tsm.clip.height+_gm.tsHeight;_tsm.zIndex=_zi+1;_sLayer(_tsm,"show");_bsm.zIndex=_zi+1;_sLayer(_bsm,"s
how");_gm.nsDoScroll=1}}else{if((_gm.clip.height>_bH)||_gm.nsDoScroll){_cor=_tsm.clip.height;if(_gmb.top<_
cor-2){_gmb.top=2;_gm.clip.height=_bH-
```

```
_cor;_gmb.clip.height=_gm.clip.height;_sScrTop=0;if(_gm.mmScrollTop)_sScrTop=_gm.mmScrollTop;_gm.top=_cor-1+_m[$m][12]-
_sScrTop;_tsm.top=2;_tsm.left=_gmb.left;_tsm.zIndex=_zi+1;_bsm.left=_gmb.left;_bsm.top=(_gmb.pageY+_gmb.clip.height)-
_tsm.clip.height+_gm.tsHeight;_tsm.zIndex=_zi+1;_sLayer(_tsm,"show");_bsm.zIndex=_zi+1;_sLayer(_bsm,"show");_gm.nsDoScroll=1}}}}else{if(!(_m[$m][7])){_sLayer(_gm,"hide");_sLayer(_gmb,"hide");if(_m[$m][13]=="scroll"){_tsm=$F("tscroll"+$m);_sLayer(_tsm,"hide");_tsm=$F("bscroll"+$m);_sLayer(_tsm,"hide")}}}}function
forceCloseAllMenus(){if(_cel>-
1){_cmo=$F("menu"+_mi[_cel][0]);if(!_cmo)_cmo=$F("oel"+_cel);for(_a=0;_a<_m.length;_a++){if(!_m[_a][7]&&!_m[_a][10])$Y(_a,0)}_zi=999;_el=-1}}function $Z(){if(_cel>-
1){_cmo=$F("menu"+_mi[_cel][0]);if(!_cmo)_cmo=$F("oel"+_cel);if(!_onTS&&_cmo&&(MouseX>(_cmo.pageX+_cmo.clip.width)||MouseY>(_cmo.pageY+_cmo.clip.height)||MouseX<_cmo.pageX||MouseY<_cmo.pageY)){$R=0;for(_ca=0;_ca<_m.length;_ca++){if(!_m[_ca][7]&&!_m[_ca][10])$Y(_ca,0);if(_m[_ca][21]>-
1){d$(_m[_ca][21]);_m[_ca][21]=-1}}_zi=999;_el=-1}}}function close_menu(){if(_el==-
1)_MT=setTimeout("$Z()",_menuCloseDelay)}function
close_el(_i){if(_mi[_i][43])eval(_mi[_i][43]);clearELs(_i);_W.status=$;$P(_oMT);_MT=setTimeout("$Z()",_menuCloseDelay);_el=-1;_oldel=_i}function $e(_gel){_gel=_mi[_gel][0];if(_m[_gel][7])_gel=-1;return _gel}function
getParentsByItem(_gmi){}function lc(_i){if(_mi[_i]=="disabled")return;location.href=_mi[_i][2]}function
_is($m,_SCRam){_onTS=1;_cel=_m[$m][0][0];$P(_MT);$P(_scrmt);_doScroll($m,_SCRam);if(!_W._scrollDelay)_scrollDelay=10;_scrmt=setTimeout("_is("+$m+","+_SCRam+")",_scrollDelay)}function
_doScroll($m,_SCRam){gm=$F("menu"+$m);if(_SCRam<0&&((gm.clip.top+gm.clip.height)>gm.fullHeight+gm.tsHeight+_SCRam))return;if(_SCRam>0&&gm.clip.top<_SCRam)return;gm.top=gm.top+_SCRam;gm.scrollTop=gm.top;gm.mmScrollTop=gm.clip.top-_SCRam;gm.clip.top=gm.clip.top-
_SCRam;gm.clip.height=gm.clip.height-_SCRam}function
set_status(_i){if(_mi[_i][4]!=null){status=_mi[_i][4]}else{if(_mi[_i][2])status=_mi[_i][2];else status=$}}function
$x(_ofs){_ofsv=0;if(isNaN(_ofs)&&_ofs.indexOf("offset=")==0){_ofsv=parseInt(_ofs.substr(7,99))}return
_ofsv}function popup(){_arg=arguments;$P(_MT);$P(_oMT);if(_arg[0]!="M_toolTips")if(_cel>-
1)forceCloseAllMenus();if(_arg[0]){if(_arg[0]!="M_toolTips"){_sm=new
Array;$Z()}_ofMT=0;$m=$h(_arg[0]);if(!_m[$m])return;_cel=_m[$m][0][0];_tos=0;if(_arg[2])_tos=_arg[2];_los=0;if(_arg[3])_los=_arg[3];_sm[_sm.length]=$m;if(_arg[1]){_gm=$F("menu"+$m);_gp=$D(_gm);if(_arg[1]==1){if(MouseY+_gp[2]>(_bH)+_sT)_tos=-(MouseY+_gp[2]-_bH)+_sT;if(MouseX+_gp[3]>(_bW)+_sL)_los=-(MouseX+_gp[3]-
_bW)+_sL;if(_m[$m][2]){if(isNaN(_m[$m][2]))_tos=$x(_m[$m][2]);else{_tos=_m[$m][2];MouseY=0}}if(_m[$m][3]){if(isNaN(_m[$m][3]))_los=$x(_m[$m][3]);else{_los=_m[$m][3];MouseX=0}}if(ns6&&!ns60){_los-=_sL;_tos-=_sT}$E(_gm,MouseY+_tos,MouseX+_los)}else{for(_a=0;_a<_d.images.length;_a++){if(_d.images[_a].name==_arg[1])_po=_d.images[_a]}$E(_gm,_po.y+_po.height+$x(_m[$m][2]),_po.x+$x(_m[$m][3]))}}$Y($m,1);_m[$m][21]=-1}}function
Opopup(_mn,_mp){$P(_MT);$Z();if(_mn){$m=$h(_mn);_sm[_sm.length]=$m;$Y($m,1);_m[$m][21]=-1}}function
popdown(){_MT=setTimeout("$Z()",_menuCloseDelay)}function
h$(_i){if(mac)_menuOpenDelay=0;_cel=_i;$P(_MT);$P(_cMT);$P(_oMT);if(_mi[_i][34]=="disabled")return;clearELs(_i);if(_oldel>-1)clearELs(_oldel);$m=-
1;_el=_i;_itemRef=_i;_mopen=$tL(_mi[_i][3]);$Q=0;if(_m[_mi[_i][0]][9])$Q=1;e$(_i);if(!_sm.length){_sm[_sm.length]=_mi[_i][0];$j=_mi[_i][0]}_iP=$e(_el);if(_iP==-
1)$j=_mi[_i][0];set_status(_el);_cMT=setTimeout("cm()",_menuOpenDelay);if(_mopen&&(!_mi[_el][39]||$R)&&_mi[_el][34]!="tree"){_gel=$F("el"+_i);_gp=$D(_gel);$m=$h(_mopen);if(_mi[_i][41])_m[$m][10]=1;if($m>-
1){_gp=$D(_gel);_mnO=$F("menu"+$m);_mp=$D(_mnO);if($Q){$k=_gp[0]+_gp[2]+1;$l=_gp[1];if(_m[$m][11]=="rtl"||_m[$m][11]=="uprtl"){$l=$l-(_mp[3]-_gp[3])-
_mi[_i][27]}if(_m[_mi[_i][0]][5]=="bottom"||_m[$m][11]=="up"||_m[$m][11]=="uprtl"){$k=(_gp[0]-
_mp[2])}}else{$k=_gp[0]+_subOffsetTop;$l=_gp[1]+_gp[3]+_subOffsetLeft;if(_m[$m][11]=="rtl"||_m[$m][11]=="uprtl"){$l=_gp[1]-_mp[3]-
_subOffsetLeft}}if($l<0)$l=0;if($k<0)$k=0;if(_m[$m][2]){if(isNaN(_m[$m][2])&&_m[$m][2].indexOf("offset=")==0){_os=_m[$m][2].substr(7,99);$k=$k+parseInt(_os)}else{$k=_m[$m][2]}}if(_m[$m][3]){if(isNaN(_m[$m][3])&&_m[$m][3].indexOf("offset=")==0){_os=_m[$m][3].substr(7,99);$l=$l+parseInt(_os)}else{$l=_m[$m][3]}}if($l+_mp[3]>_bW+_sL){if(!$Q&&(_gp[1]-_mp[3])>0){$l=_gp[1]-_mp[3]-_subOffsetLeft}else{$l=(_bW-_mp[3])-
2}}if(!$Q&&$k+_mp[2]>_bH+_sT){$k=(_bH-_mp[2])-2}if(!$Q){$k=$k-_m[$m][6][65]}else{$k--;$l--
}$E(_mnO,$k+_m[$m][6][65],$l+_m[$m][6][65]);if(_m[$m][5])p$($m);_zi++;_mnb=$F("bord"+$m);_oMT=setTimeout("$Y("+$m+",1)",_menuOpenDelay);if(_sm[_sm.length-1]!=$m)_sm[_sm.length]=$m}i$(_iP)}function
p$($m){if(_m[$m][5]){_gm=$F("menu"+$m);_gp=$D(_gm);_osl=0;_omnu3=0;if(isNaN(_m[$m][3])&&_m[$m][3].indexOf("offset=")==0){_omnu3=_m[$m][3];_m[$m][3]=_n;_osl=_omnu3.substr(7,99);_gm.leftOffset=_osl}_lft=_n;if(!_m[$m][3]){if(_m[$m][5].indexOf("left")!=-1)_lft=0;if(_m[$m][5].indexOf("center")!=-1)_lft=(_bW/2)-(_gp[3]/2);if(_m[$m][5].indexOf("right")!=-1)_lft=_bW-
_gp[3];if(_gm.leftOffset)_lft=_lft+parseInt(_gm.leftOffset)}_ost=0;_omnu2=0;if(isNaN(_m[$m][2])&&_m[$m][2].in
```

```
dexOf("offset=")==0){_omnu2=_m[$m][2];_m[$m][2]=_n;_ost=_omnu2.substr(7,99);_gm.topOffset=_ost}_tp=_n
;if(!_m[$m][2]>=0){_tp=_n;if(_m[$m][5].indexOf("top")!=-1)_tp=0;if(_m[$m][5].indexOf("middle")!=-
1)_tp=(_bH/2)-(_gp[2]/2);if(_m[$m][5].indexOf("bottom")!=-1)_tp=_bH-
_gp[2];if(_gm.topOffset)_tp=_tp+parseInt(_gm.topOffset)}if(_lft<0)_lft=0;$E(_gm,_tp,_lft);if(_m[$m][19])_m[$m][
19]=_tp;if(_tp)_tp=_tp-_m[$m][6][65];if(_lft)_lft=_lft-
_m[$m][6][65];_sb=$F("bord"+$m);$E(_sb,_tp,_lft);_gm._tp=_tp}}getMenuByItem=$d;getParentItemByItem=$f;
_drawMenu=o$;BDMenu=g$;gmobj=$F;menuDisplay=$Y;gpos=$D;spos=$E;_fixMenu=$z;getMenuByName=$
h;itemOn=e$;itemOff=d$;_popi=h$;clickAction=$K;_setPosition=p$;closeAllMenus=$Z;function
$N(){_bW=self.innerWidth-16;_bH=self.innerHeight-
17;_sT=self.pageYOffset;if(_startM==1){for(_a=0;_a<_m.length;_a++){if(_m[_a]&&_m[_a][5]){p$(_a)}}}_startM=
0;_oldbH=_bH;_oldbW=_bW}setInterval("$N()",200);function
getMouseXY(e){MouseX=e.pageX;MouseY=e.pageY;mmMouseMove()}_d.captureEvents(Event.MOUSEMOV
E);_d.onmousemove=getMouseXY;
```

-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

../menu/milonic_src.js
-o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o--o-

```
/*

Milonic DHTML Menu - JavaScript Website Navigation System.
Version 5.729 - Built: Thursday August 18 2005 - 6:35
Copyright 2005 (c) Milonic Solutions Limited. All Rights Reserved.
This is a commercial software product, please visit http://www.milonic.com/ for more information.
See http://www.milonic.com/license.php for Commercial License Agreement
All Copyright statements must always remain in place in all files at all times
*******  PLEASE NOTE: THIS IS NOT FREE SOFTWARE, IT MUST BE LICENSED FOR ALL USE  *******

License Number: 1000 for Unlicensed

*/


_mD=2;_d=document;_dB=_d.body;_n=navigator;_L=location;_nv=$tL(_n.appVersion);_nu=$tL(_n.userAgent)
;_ps=parseInt(_n.productSub);_f=false;_t=true;_n=null;_W=window;$$="
";$="";_wp=_W.createPopup;ie=(_d.all)?_t:_f;ie4=(!_d.getElementById&&ie)?_t:_f;ie5=(!ie4&&ie&&!_wp)?_t:_f;i
e55=(!ie4&&ie&&_wp)?_t:_f;ns6=(_nu.indexOf("gecko")!=-1)?_t:_f;konq=(_nu.indexOf("konqueror")!=-
1)?_t:_f;sfri=(_nu.indexOf("safari")!=-
1)?_t:_f;if(konq||sfri){_ps=0;ns6=0}ns4=(_d.layers)?_t:_f;ns61=(_ps>=20010726)?_t:_f;ns7=(_ps>=20020823)?
_t:_f;ns72=(_ps>=20040804)?_t:_f;op=(_W.opera)?_t:_f;if(op||konq)ie=0;op5=(_nu.indexOf("opera 5")!=-
1)?_t:_f;op6=(_nu.indexOf("opera 6")!=-1||_nu.indexOf("opera/6")!=-1)?_t:_f;op7=(_nu.indexOf("opera 8")!=-
1||_nu.indexOf("opera 7")!=-1||_nu.indexOf("opera/7")!=-1)?_t:_f;mac=(_nv.indexOf("mac")!=-
1)?_t:_f;if(ns6||ns4||op||sfri)mac=_f;ns60=_f;if(ns6&&!ns61)ns60=_t;if(op7)op=_f;IEDtD=0;if(!op&&((_d.all||ns7)
&&_d.compatMode=="CSS1Compat")||(mac&&_d.doctype&&_d.doctype.name.indexOf(".dtd")!=-
1))IEDtD=1;_jv="javascript:void(0)";_lN="\n";_rstC=inDragMode=_d.dne=lcl=$R=$mD=_mcnt=_sL=_sT=_ofMT
=_oldbW=_bW=_oldbH=_bl=_el=_st=_en=_cKA=0;_startM=_c=1;_oldel=_itemRef=_mn=-
1;_zi=_aN=_bH=999;if(op)ie55=_f;ab$="absolute";$O="menu";$5="hidden";_d.write("<style>.milonic{width:1px;
visibility:hidden;position:absolute}</style>");function _nA(){return new Array()}function _StO(_fnc,_ms){return
setTimeout(_fnc,_ms)}tTipt=$;_m=_nA();_mi=_nA();_sm=_nA();_tsm=_nA();_cip=_nA();$S3="2E636F6D2F";$
S4="646D2E706870";_MT=_StO($,0);_oMT=_StO($,0);_cMT=_StO($,0);_mst=_StO($,0);_Mtip=_StO($,0);$u=
"undefined
";lNum=1000;lURL="Unlicensed";lVer="5.729";_Lhr=_L.href;$6="visible";if(op5){$5=$tU($5);$6=$tU($6)}functio
n M_hideLayer(){}function _oTree(){}function mmMouseMove(){}function _cL(){}function _ocURL(){}function
mmClick(){}function autoOT(){}function _X_(){}function _iF0C(){}function showtip(){}function
mmVisFunction(){}function remove(_ar,_dta){var
_tar=_nA();for(_a=0;_a<_ar.length;_a++){if(_ar[_a]!=_dta){_tar[_tar.length]=_ar[_a]}}return _tar}function
copyOf(_w){for(_cO in _w){this[_cO]=_w[_cO]}}function $tL($v){if($v)return $v.toLowerCase()}function
$tU($v){if($v)return $v.toUpperCase()}function $pU($v){if($v)return parseInt($v)}function
drawMenus(){_startM=1;_oldbH=0;_oldbW=0;for(_y=_mcnt;_y<_m.length;_y++){o$(_y,1)}}_$S={menu:0,text:1,
url:2,showmenu:3,status:4,onbgcolor:5,oncolor:6,offbgcolor:7,offcolor:8,offborder:9,separatorcolor:10,padding:
11,fontsize:12,fontstyle:13,fontweight:14,fontfamily:15,high3dcolor:16,low3dcolor:17,pagecolor:18,pagebgcolor
```

:19,headercolor:20,headerbgcolor:21,subimagepadding:22,subimageposition:23,subimage:24,onborder:25,ondecoration:26,separatorsize:27,itemheight:28,image:29,imageposition:30,imagealign:31,overimage:32,decoration:33,type:34,target:35,align:36,imageheight:37,imagewidth:38,openonclick:39,closeonclick:40,keepalive:41,onfunction:42,offfunction:43,onbold:44,onitalic:45,bgimage:46,overbgimage:47,onsubimage:48,separatorheight:49,separatorwidth:50,separatorpadding:51,separatoralign:52,onclass:53,offclass:54,itemwidth:55,pageimage:56,targetfeatures:57,visitedcolor:58,pointer:59,imagepadding:60,valign:61,clickfunction:62,bordercolor:63,borderstyle:64,borderwidth:65,overfilter:66,outfilter:67,margin:68,pagebgimage:69,swap3d:70,separatorimage:71,pageclass:72,menubgimage:73,headerborder:74,pageborder:75,title:76,pagematch:77,rawcss:78,fileimage:79,clickcolor:80,clickbgcolor:81,clickimage:82,clicksubimage:83,imageurl:84,pagesubimage:85,dragable:86,clickclass:87,clickbgimage:88,imageborderwidth:89,overseparatorimage:90,clickseparatorimage:91,pageseparatorimage:92,menubgcolor:93,opendelay:94};function mm_style(){for($i in _$S)this[$i]=_n;this.built=0}_$M={items:0,name:1,top:2,left:3,itemwidth:4,screenposition:5,style:6,alwaysvisible:7,align:8,orientation:9,keepalive:10,openstyle:11,margin:12,overflow:13,position:14,overfilter:15,outfilter:16,menuwidth:17,itemheight:18,followscroll:19,menualign:20,mm_callItem:21,mm_obj_ref:22,mm_built:23,menuheight:24,ignorecollision:25,divides:26,zindex:27,opendelay:28};function menuname(name){for($i in _$M)this[$i]=_n;this.name=$tL(name);_c=1;_mn++;this.menunumber=_mn}function _incItem(_it){_mi[_bl]=_nA();for($i in _x[6])if(_x[6][$i])_mi[_bl][_$S[$i]]=_x[6][$i];_mi[_bl][0]=_mn;_it=_it.split(";");for(_a=0;_a<_it.length;_a++){_sp=_it[_a].indexOf("`");if(_sp!=-1){_tI=_it[_a];if(_sp==_it[_a].lastIndexOf("`")){for(_b=_a;_b<_it.length;_b++){if(_it[_b+1]){_tI+=";"+_it[_b+1];_a++;if(_it[_b+1].indexOf("`")!=-1)_b=_it.length}}}_it[_a]=_tI.replace(/`/g,$)}_sp=_it[_a].indexOf("=");if(_sp==-1){if(_it[_a])_si=_si+";"+_it[_a]}else{_si=_it[_a].slice(_sp+1);_w=_it[_a].slice(0,_sp);if(_w=="showmenu")_si=$tL(_si)}if(_it[_a]){_mi[_bl][_$S[_w]]=_si}}_m[_mn][0][_c-2]=_bl;_c++;_bl++}_c=0;function ami(txt){_t=this;if(_c==1){_c++;_m[_mn]=_nA();_x=_m[_mn];for($i in _t)_x[_$M[$i]]=_t[$i];_x[21]=-1;_x[0]=_nA();if(!_x[12])_x[12]=0;_MS=_m[_mn][6];_MN=_m[_mn];if(_MN[15]==_n)_MN[15]=_MS.overfilter;if(_MN[16]==_n)_MN[16]=_MS.outfilter;_MS[65]=(_MS.borderwidth)?$pU(_MS.borderwidth):0;_MS[64]=_MS.borderstyle;_MS[63]=_MS.bordercolor;if(_W.ignoreCollisions){_MN[25]=1}if(!_MS.built){_WzI=_zi;if(_W.menuZIndex){_WzI=_W.menuZIndex;_zi=_WzI}lcl++;_vC=_MS.visitedcolor;if(_vC){_oC=_MS.offcolor;if(!_oC)_oC="#000000";if(!_vC)_vC="#ff0000";_Lcl="<style>.linkclass"+lcl+":link{color:"+_oC+"}.linkclass"+lcl+":visited{color:"+_vC+"}</style>";_d.write(_Lcl);_MS.linkclass="linkclass"+lcl}_MS.built=1}}_incItem(txt)}menuname.prototype.aI=ami;

Bibliografía.

Libros:

1. Applegate, L., McFarlan, W. y McKenney, J., Corporate Information Systems Management: Text and Cases, McGraw-Hill, 5a. Edición, 1999.m

2. Witten, J.L. y L.D. Bentley, Systems Analysis and Design Methods, Irwin McGraw-Hill, Boston, 1999.

3. V. Pérez y J. Pino, Curso de Computación e Informática, Sistema de Administración y Sistemas de Información Administrativos, Volumen 3 y 4, Editorial Universitaria S.A., 4a. edición 1986.

4. O. Barros, Reingeniería de Procesos de Negocios: un Planteamiento Metodológico, Dolmen, 1994.

5. Kendall y Kendall, Análisis y Diseño de Sistemas Editorial Prentice-Hall, Primera edición.

6. Apuntes y presentaciones del diplomado elaborados por Pedro Gabriel Ramírez y Paola Medina Mora Barrera

Material de Internet:

1. Cómo obtener el control de sus costos de viajes y entretenimiento
http://www.microsoft.com/spain/medianaempresa/businessvalue/travelcosts.mspx

2. Manejo y control de los gastos de viaje y representación.
http://www10.americanexpress.com/sif/cda/page/0,1641,13120,00.asp

3. Gastos en viajes de empresas se duplicarán hasta llegar 900 billones
http://www.eleconomista.es/empresas-finanzas/noticias/91460/10/06/Gastos-viajes-empresas-se-duplicaran-hasta-llegar-900-billones.html

4. El gasto anual por concepto de viajes y representación, que se paga con tarjetas corporativas, aumentó más de 20% de 2005 a 2006.
http://www.noticiascadadia.com/noticias/articulo/el-gasto-anual-por-concepto-de-viajes-y-representacion-que-se-paga-con-tarjetas-corporativas-aume.html

5. Wikipedia (www.wikipedia.org):

Artículos:

Data Transformation Serveices - DTS(p. 19):
http://en.wikipedia.org/wiki/Data_Transformation_Services

Database Administrator - DBA (p. 19):
http://en.wikipedia.org/wiki/Database_administrator

Buisness to business - B2B (p. 133):
http://es.wikipedia.org/wiki/B2B

Normalización de bases de datos (p. 76):
http://en.wikipedia.org/wiki/Database_normalization