

53
2ej



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

"SISTEMA INTEGRAL DE ABASTECIMIENTOS BAJO EL AMBIENTE SYBASE"

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION

P R E S E N T A N :

DORA MA. MIJANGOS ORDOÑEZ
LIGIA FERNANDEZ GAMA
GABRIEL DE JESUS COSSIO NOEGGERATH

DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA



MEXICO, D. F.

1993

TESIS CON
FALLA DE ORIGEN



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

OBJETIVOS i

INTRODUCCION.....iii

CAPITULO I. FUNDAMENTOS TEORICOS

I.1 CONCEPTOS BASICOS DEL SISTEMA OPERATIVO UNIX ... 1

I.2 SISTEMAS MULTIUSUARIO 20

I.2.1 DEFINICION 21

I.2.2 CARACTERISTICAS GENERALES..... 23

I.2.3 VENTAJAS Y DESVENTAJAS 35

I.3 INTRODUCCION A LAS BASES DE DATOS..... 37

I.3.1 DEFINICION 43

I.3.2 ELEMENTOS..... 49

I.3.3 ARQUITECTURAS 55

I.3.4 ABSTRACCION DE DATOS..... 59

I.3.5 BASES DE DATOS RELACIONALES 60

I.3.6 BASES DE DATOS EN AMBIENTE MULTIUSUARIO..... 72

I.3.7 LENGUAJE ESTANDAR AUXILIAR EN EL DESARROLLO
DE LOS SISTEMAS DE BASES DE DATOS SQL..... 87

I.4	DISEÑO DE BASES DE DATOS	105
I.4.1	ETAPAS DEL PROCESO DE DISEÑO	108
I.4.2	MODELO ENTIDAD-RELACION	111
I.4.3	ANÁLISIS DE DATOS.....	114
I.4.4	DISEÑO DEL ESQUEMA LÓGICO.....	123
I.4.5	CONVERSIÓN DEL ESQUEMA LÓGICO A UNA BASE DE DATOS FÍSICA.....	124
I.5	SISTEMAS MANEJADORES DE BASES DE DATOS	126
I.5.1	ORACLE	133
I.5.2	INGRES	140
I.5.3	SYBASE.....	145

CAPÍTULO II. ANÁLISIS DE LA PROBLEMÁTICA DE LA SUBDIRECCIÓN DE ABASTECIMIENTOS DEL GRUPO PEÑÓLES

II.1	ANTECEDENTES.....	152
II.2	DETECCIÓN DE LA PROBLEMÁTICA	166
II.2.1	PUNTOS CRÍTICOS.....	166
II.2.2	CONSECUENCIAS.....	168

II.3 IDENTIFICACION Y ANALISIS DE LOS REQUERIMIENTOS DEL AREA.....	173
II.4 ANALISIS DE LA INFORMACION	189
II.5 ALTERNATIVAS DE SOLUCION.....	200

CAPITULO III. DISEÑO E IMPLEMENTACION DEL SISTEMA DE ABASTECIMIENTOS.

III.1 ELECCION DEL MANEJADOR DE BASES DE DATOS.....	210
III.1.1 CRITERIOS DE EVALUACION	211
III.1.1.1 ESTRUCTURA Y ARQUITECTURA.....	212
III.1.1.2 FUNCIONALIDAD.....	217
III.1.1.3 FUNCIONALIDAD DISTRIBUIDA.....	222
III.1.1.4 PERFORMANCE.....	227
III.1.1.5 CPU	228
III.1.1.6 MEMORIA.....	232
III.1.1.7 ENTRADAS/SALIDAS DISCO	234
III.1.1.8 AMBIENTE DE DESARROLLO	236
III.1.1.9 UTILERIAS	238
III.1.2 ELECCION DE LA HERRAMIENTA	240

III.2 APLICACIONES DE DESARROLLO EN SYBASE	251
III.2.1 ISQL.....	251
III.2.2 DATA WORKBENCH.....	256
III.2.3 REPORT WORKBENCH	258
III.2.4 BULLK COPY	259
III.2.5 APT WORKBENCH	260
III.3 DISEÑO DEL SISTEMA	265
III.3.1 MODULOS PRINCIPALES	265
III.3.2 DIAGRAMA DE FLUJO DE DATOS	274
III.3.3 DIAGRAMA ENTIDAD-RELACION.....	280
III.3.4 DICCIONARIO DE DATOS.....	285
III.3.5 NIVELES DE ACCESO A USUARIOS.....	285
III.4 DESARROLLO DEL SISTEMA.....	287
III.4.1 PROGRAMACION	287
III.4.2 PRUEBAS Y AJUSTES	297
III.5 IMPLANTACION Y EVALUACION.....	300
III.5.1 CAPACITACION.....	300
III.5.2 EVALUACION DE RESULTADOS.....	303

CONCLUSIONES..... 312

BIBLIOGRAFIA.....315

GLOSARIO.....317

APENDICE A. DICCIONARIO DE DATOS

APENDICE B. GRANTS, PROCEDIMIENTOS - ALMACENADOS Y TRIGGERS

APENDICE C. PROGRAMACION

APENDICE D. DEFINICION DE FORMAS DEL SISTEMA

APENDICE E. DOCUMENTACION RECOPIADA

APENDICE F. MANUAL DE USUARIO

OBJETIVOS

- Apoyar mediante la implementación de procesos automatizados el logro de las metas y el incremento de la productividad de la Subdirección de Abastecimiento del Grupo Peñoles.
- Desarrollo de un sistema para el control de las adquisiciones de materias primas, suministros, equipo y refacciones en las diferentes unidades y oficinas regionales que tiene el Grupo Peñoles distribuidas en todo el país.
- El sistema tiene como metas específicas:
 - Asegurar la disponibilidad de material para la producción
 - Tener mejores bases para el flujo de efectivo
 - Reducir el ciclo de las órdenes de pago
 - Mejorar la calidad en la selección de los proveedores
- Será desarrollado bajo el enfoque de Bases de Datos Relacionales en una arquitectura Cliente-Servidor.
- Deberá tener una interface de usuario amigable, para evitar que los usuarios tengan que realizar operaciones desconocidas para ellos.

INTRODUCCION

En la actualidad, para muchas organizaciones, los sistemas de información basados en computadora son el corazón de las actividades cotidianas y objeto de gran consideración en la toma de decisiones.

Las empresas consideran con mucho cuidado las capacidades de sus sistemas de información cuando deciden ingresar o no en nuevos mercados o cuando planean la respuesta que darán a la competencia. Sin ayuda de la automatización, las dependencias tendrían que hacer un alto ante el volumen de trabajo que abrumaría a sus administradores y empleados. Por otra parte, la capacidad de comunicación de datos determina tanto el lugar como el momento en que fluirá la información.

El desarrollo de sistemas de información involucra tanto a los analistas de sistemas como a todos aquellos que harán uso de las aplicaciones que se desarrollen, es decir los usuarios finales. El análisis y diseño de sistemas de información incluye diversas partes de las organizaciones y no están limitados al dominio de los especialistas en computadoras.

De esta forma, nuestro trabajo de tesis está basado en el desarrollo de un sistema que tiene una aplicación real.

El sistema al que hacemos referencia es el **Sistema Integral de Abastecimiento**, el cual se encargará de automatizar el proceso administrativo de la Subdirección de Abastecimiento del Grupo Peñoles.

En cuanto al trabajo, podemos decir que este consta de tres capítulos principales distribuidos de la siguiente forma:

Capítulo I.- Trata todo lo referente a los fundamentos teóricos que están directamente relacionados con el desarrollo de nuestro sistema, tales como: los Sistemas Multiusuario, el Sistema Operativo UNIX , el ambiente de las Bases de Datos y los principales Sistemas Manejadores de Bases de Datos Relacionales (RDBMS) existentes hoy en día en el mercado.

Capítulo II.- Se da una breve introducción sobre lo que es el Grupo Peñoles y donde está ubicada la Subdirección de Abastecimiento dentro de la empresa. Se analiza la problemática actual de dicha Subdirección y se plantea la solución al analizar sus funciones y sus requerimientos de información.

Capítulo III.- Se diseñan los diferentes módulos del sistema (Modelo conceptual) y se hace la elección de la herramienta a utilizar para su desarrollo. Una vez realizado esto, se expone brevemente cómo se llevó a cabo la programación y la implementación física junto con todas las actividades relacionadas.

CAPITULO I

FUNDAMENTOS TEORICOS

I.1 CONCEPTOS BASICOS DEL SISTEMA OPERATIVO UNIX

¿Qué es un sistema operativo?

Los programas para computadora pueden dividirse en dos grandes tipos:

-Programas-sistema los cuales manejan la computadora y realizan las actividades cotidianas de manipulación del equipo físico. El equipo físico lo constituye el CPU, la memoria primaria, los dispositivos externos de entrada y de salida y los dispositivos de almacenamiento.

-Programas de aplicación los cuales resuelven problemas de los usuarios.

El programa-sistema más importante es el sistema operativo. Este ofrece una plataforma para ejecutar los programas de aplicación, además de controlar y administrar todos los recursos de la máquina. Un sistema operativo es un programa que interactúa entre el usuario y el equipo físico (hardware).

Se puede decir así, que un sistema operativo es un controlador y administrador de los recursos de la máquina. Este tipo de programa representa una interfaz entre la máquina y los usuarios.

El sistema operativo lo podemos estudiar por sus funciones y por sus componentes. Es decir, por lo que hace y por lo que es. Este realiza ciertas tareas para el usuario y ciertas tareas para la administración de los recursos de la máquina.

El sistema operativo controla y coordina el uso del hardware para que los usuarios y los programas de aplicación utilicen fácil y eficazmente los recursos del mismo. También es el enlace entre las diferentes partes de un sistema informático. El sistema operativo ofrece ciertos servicios, entre los que tenemos:

- Ejecución de programas.
- Operaciones de E/S.
- Manipulación de sistemas de archivos.
- Detección de errores.
- Distribución de recursos.
- Contabilidad sobre la utilización de recursos.
- Mecanismos de protección de la información.

Un sistema operativo tiene cuatro componentes principales:

- Manejador de procesos.
- Manejadores de dispositivos de E/S.
- Manejador de memoria.
- Sistema de archivos.

Entre las principales funciones y componentes de un sistema operativo tenemos:

- El interpretador de mandatos. En UNIX se le denomina "Shell".
- Las utilerías. Por ejemplo los editores, compiladores, filtros, correo, etc..
- Los controladores de recursos. Los administradores del tiempo compartido, los manejadores de dispositivos de E/S, los manejadores de memoria primaria y secundaria.

Desde un punto de vista descendente, el sistema operativo es una interfaz entre el hardware y los usuarios. Desde un punto de vista ascendente, el sistema operativo es un manejador y administrador de recursos que controla en forma eficaz la distribución de los procesadores, de las memorias y de los dispositivos de E/S, entre todos los programas que soliciten los servicios. UNIX es un sistema operativo que ofrece todos estos servicios.

Generalidades del Sistema Operativo.

UNIX es el sistema operativo que en la actualidad ha sido más implantado en diferentes tipos de máquinas, desde las personales PC hasta las supercomputadoras, como la CRAY. UNIX es el sistema más utilizado para la comunicación entre máquinas, ya que permite realizar redes de computadoras en sistemas multiusuarios en forma abierta. Es el ejemplo de la tecnología de Sistemas Abiertos.

Actualmente, se requiere que la mayoría de los sistemas informáticos se encuentren conectados en red, por esta razón el aspecto conectividad y comunicaciones del sistema UNIX, es de vital importancia para la industria y la empresa. Este ayuda a

evitar las islas de información, ofreciendo de esta manera un factor importante de productividad como es el tener un flujo acelerado de información entre los equipos de producción.

UNIX es un sistema multiusuario. Esto significa que pueden estar utilizando al mismo tiempo el sistema varios usuarios, cada uno teniendo a su disposición todos los recursos del sistema. UNIX es un sistema multiproceso, lo que permite que cada usuario pueda ejecutar varios procesos al mismo tiempo.

UNIX es un ejemplo de la corriente de Sistemas Abiertos con las características de escalabilidad de aplicaciones y conectividad entre diversos sistemas en redes de cómputo distribuidas.

La popularidad y éxito de UNIX se debe a varios factores. Estos incluyen aplicaciones portables a un gran número de máquinas, su fácil adaptación y sencillez de uso, el amplio espectro de tareas que puede realizar, su naturaleza multiusuario y multiproceso.

El sistema operativo está escrito en lenguaje C, en lugar de ensamblador como la mayoría de los sistemas; esto le permite una fácil lectura y una completa adaptación a las diferentes computadoras. Como otros sistemas, consta de dos partes : el núcleo y los programas del sistema.

El CPU está organizado de forma que favorece los procesos interactivos, de tal suerte que para la planificación de procesos se usan algoritmos basados en prioridades. Los

archivos se organizan en directorios estructurados en árbol (sistema jerárquico de procesos y archivos) y se manejan mediante bloques de datos.

Está formado por una serie de elementos que pueden representarse en forma de capas concéntricas donde, en primer lugar, alrededor del hardware de la máquina se encuentra el núcleo (Kernel), que interactúa directamente con el hardware, aislando a éste de los usuarios, además de adaptar el resto del sistema operativo a la máquina, debido a la característica de este de ser un sistema portable (figura I.1.1).

En una segunda capa se encuentran los comandos, que no son otra cosa que la interfase entre los programas de aplicación y el núcleo del sistema operativo. La última de las capas contiene los programas de aplicación.

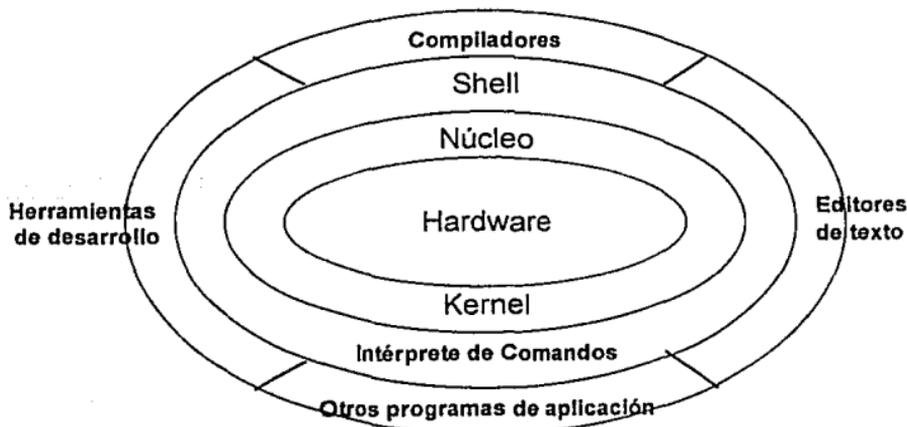


Figura I.1.1. Capas concéntricas de UNIX

Historia del sistema UNIX

Entre los años 1965 y 1970 surgió el proyecto MULTICS, el cual pretendía construir un sistema operativo para computadoras grandes., MULTICS surge de MULTiplexed Information and Computing System. Este fue un gran proyecto donde participaron, entre otras empresas: General Electric, ATT, MIT, etc...Sin embargo, este proyecto no continuó desarrollándose como se había planeado originalmente.

En 1968, Ken Thompson, antiguo miembro del grupo MULTICS, y Dennis Ritchie (Bell Laboratories), pensaron construir un sistema de archivos, para uso interno dentro del laboratorio. Originalmente, Thompson y Ritchie no tenían la idea de construir un sistema operativo. Sin embargo, posteriormente al sistema de archivos se le añadió el concepto de proceso. El sistema construido por Thompson y Ritchie tuvo bastante aceptación entre los miembros del laboratorio, y se decidió ampliarlo para desarrollar un sistema operativo. A este grupo se unió Brian Kernighan, quien en 1970 bautiza al sistema como UNIX, en recuerdo a MULTICS.

Entre las fechas mas importantes en la primera fase del desarrollo del sistema tenemos:

- 1971, se publica la primera edición del manual.
- 1972, se tiene la noción de pipe o entubamiento entre los mandatos del sistema y se edita también la segunda edición del manual.

En los años setenta, en los Estados Unidos de América existía una ley que impedía a los laboratorios Bell del grupo AT&T comercializar programas de computadora. Por este motivo el sistema UNIX se le regalaba a las Universidades y Centros de

este motivo el sistema UNIX se le regalaba a las Universidades y Centros de Investigación. Entre las Universidades que más contribuyeron al desarrollo del sistema tenemos la Universidad de Berkeley en California. En esta Universidad surgió el sistema UNIX que se conoce como la versión BSD (Berkeley Software Distribution). Este sistema tiene características importantes para conectividad y desarrollo de aplicaciones.

Una de las primeras compañías que ofreció equipo con dispositivos Ethernet fue DEC para las máquinas VAX. La Universidad de Berkeley en los años ochentas principia un gran proyecto de desarrollo de aplicaciones para redes de computadoras con el sistema UNIX corriendo en máquinas VAX. En la versión BSD se soportan los protocolos para redes.

La versión BSD fue la primera versión UNIX que soportó la memoria virtual. Actualmente muchos sistemas UNIX están basados en la versión BSD. Todo esto nos inclina a decir que el sistema BSD ha marcado un papel importante en la historia de UNIX. Además de la Universidad de Berkeley tenemos la Free Software Foundation, esta es una asociación de informáticos dedicada a la producción de software de la mas alta calidad en el mercado para una distribución gratuita, donde particularmente realizan software para el mundo UNIX.

La conectividad del mundo UNIX

En el mundo de la conectividad de los sistemas informáticos, UNIX tiene un papel primordial. Soporta casi todos los protocolos de redes y ofrece una serie de utilerías para la conexión de máquinas de otros sistemas operativos.

UNIX ahora ofrece implantaciones de los protocolos TCP/IP (Transmission Control Protocol / Internet Protocol). Además se tienen mandatos para login remoto y transferencia de archivos. Se tienen mandatos para transferencia de archivos FTP (File Transfer Protocol), para ejecución remota del Shell y copiar archivos en remoto.

La comunicación entre máquinas UNIX se puede establecer por diferentes canales. Se cuenta con el mecanismo de NFS (Network File System) de Sun Microsystems. Este mecanismo facilita la comunicación de datos entre usuarios. Con este mecanismo, el cual no solo se puede implantar entre máquinas UNIX, un usuario puede acceder al sistema de archivos de otra máquina pensando que todas las operaciones se realizarán en su sistema de archivos local. Es una herramienta muy poderosa.

Se cuenta con el mecanismo de AFS (Andrew File System) el cual generaliza en cierto sentido el concepto de NFS. Sin embargo, AFS actualmente no se encuentra implantado en todo sistema UNIX.

El Kernel de UNIX

La idea fundamental en UNIX es mantener el Kernel lo mas pequeño y simple; ofrecer los servicios más solicitados por medio de módulos reemplazables, en vez de tener todos estos dentro del Kernel. En las palabras de K. Thompson,

- ...el Kernel es la única parte del código UNIX que no puede ser sustituida por un usuario a su libre gusto. Por esta razón, el Kernel debe de realizar las menores decisiones posibles. Esto no significa ofrecer al usuario un millón de opciones para efectuar una misma cosa. En vez de esto, se debe de permitir solamente una forma de hacer una sola cosa, pero esta forma debe de ser el mínimo común divisor de todas las opciones posibles, ...

El Kernel es el núcleo del sistema , tiene un doble papel; simula actividad concurrente para los procesos y también proporciona la interfase con el hardware, recibiendo todas las señales provenientes del mundo externo. El núcleo mantiene una lista de procesos actuales , lleva un control de aquellos que están libres para continuar y comparte el procesador entre ellos de acuerdo a una estrategia escogida. El núcleo también administra la asignación de almacenamiento y es responsable de controlar cómo se comparten los recursos. Esto significa que todas las operaciones que implican procesos son controlados por el núcleo, este representa solamente una pequeña porción del código de todo el sistema operativo, pero se encuentra dentro de los códigos de más amplio uso. Por esta razón, el núcleo suele permanecer en el almacenamiento primario , mientras otras porciones del sistema operativo son transportadas de un lado a otro. entre el almacenamiento primario y el secundario, según las necesidades (figura 1.1.2).

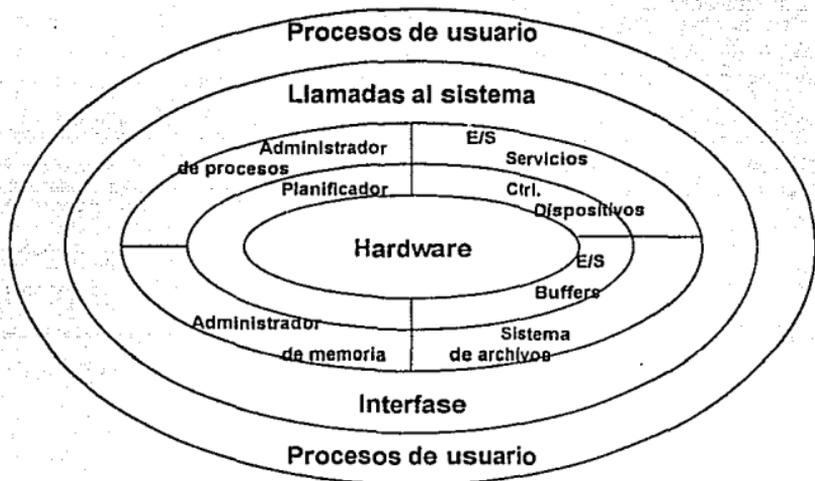


Figura I.1.2. Funciones del Kernel

Lo ideal es solo tener en el Kernel servicios mínimos: creación de procesos, comunicación entre procesos, acceso controlado a dispositivos de E/S, acceso al CPU. Fuera del Kernel tener: manejo de sistema de archivos, manejadores de terminal, protocolos de red, etc...

Sin embargo, UNIX no tiene una estructura ideal ya que dentro del Kernel se encuentra el manejo de terminales, de archivos, etc... En los nuevos sistemas operativos como el Chorus se está logrando esto. El Kernel del sistema es mínimo y todos los servicios están fuera de él.

El Shell de UNIX

La interacción con el sistema operativo generalmente se realiza por medio de un interpretador de mandatos. En el sistema operativo UNIX este interpretador de mandatos recibe el nombre de Shell. UNIX ofrece tres tipos de estos: BOURNE, C-Shell y el K-Shell. Cada usuario puede seleccionar el Shell que mejor se ajuste a sus necesidades por sus características, las cuales se muestran en la siguiente tabla.

Bourne Shell	Es el intérprete de comandos básico
C-Shell	Es el intérprete de comandos creado por Berkeley, un poco más completo que el anterior. Su programación es prácticamente lenguaje C.
Korn Shell	Se basa en los dos anteriores, siendo compatible con el Bourne en un 95%. Añade posibilidades de programación avanzada, facilidades aritméticas y mayor rapidez de ejecución.

Tabla I.1.1. Intérpretes de comandos

El Shell en UNIX actúa en la mayor parte de las ocasiones como un medio por el cual se invocan otros mandatos. El Shell es un interpretador de mandatos por línea; lee las

líneas tecleadas por el usuario y las interpreta como requerimientos para ejecutar otros programas.

Los mandatos en UNIX consisten de una lista de cadenas de caracteres o palabras interpretadas, como el nombre del mandato, seguido por los argumentos, el Shell separa el nombre del mandato y los argumentos en variables diferentes. Después se busca un archivo con el nombre del mandato especificado, si se encuentra, este archivo se lleva a memoria principal y se ejecuta. Cuando termina de ejecutarse este mandato, el Shell termina su propia ejecución e indica que está listo para recibir otro mandato. La indicación de que el Shell se encuentra listo para recibir otro mandato varía según el tipo de Shell. Tenemos que en el Shell de Bourne o en el Shell de Korn, el indicador es el símbolo \$ (dólar), y en el C-Shell el símbolo es % (porcentaje). En el caso del super usuario, está establecido entre los usuarios de UNIX el indicador de # (gato). Estos símbolos se pueden modificar asignando a las variables shell correspondientes el indicador deseado.

Tenemos así que, el Shell es entonces un interpretador de mandatos y un lenguaje de programación.

Administración del sistema UNIX.

En el sistema UNIX existe un usuario que tiene privilegios especiales, diferentes a todos los demás usuarios. Este usuario, llamado superusuario o root, es el encargado de la administración en el sistema, de los respaldos y de los cambios de configuración del mismo.

La administración del sistema es la tarea que consiste en mantener funcionando al mismo, además de verificar que esta tarea se efectúe de la mejor forma. Entre las tareas fundamentales del administrador del sistema tenemos:

- Administrar y dar mantenimiento a los sistemas de archivos.
- Monitorear la seguridad del sistema.
- Realizar los respaldos y restauraciones respectivas.
- Instalar las aplicaciones y reinstalar el sistema operativo.
- Realizar altas y bajas de usuarios al sistema.
- Administrar los procesos activos.
- Monitorear la actividad del sistema.
- Afinar la configuración del Kernel.
- Configurar los dispositivos periféricos.
- Administrar la conexión en la red.
- Verificar el tiempo de CPU de los procesos del sistema y de algún usuario en particular.

Por omisión, el superusuario trabaja siempre en el shell de Bourne, que es el estándar. Se recomienda ampliamente que el superusuario no tenga asignada en la variable de ambiente PATH ningún camino; de esta forma se obligará a teclear el nombre absoluto de todos los mandatos.

El manejo del sistema de archivos es una de las principales tareas del administrador del sistema. Entre las tareas del administrador relativas al sistema de archivos se tiene:

- Monitoreo de la ocupación del espacio de disco por usuarios en general y de un usuario en particular.
- Protección de datos limitando el acceso a archivos a cierto grupo de usuarios.
- Realización de respaldos sistemáticos en la información generada por los usuarios.
- Montaje y desmontaje de los sistemas de archivos.

El desarrollo de programas en UNIX

UNIX tiene muchas utilerías para el desarrollo de programas. Es una plataforma muy utilizada en Ingeniería de Software, fundamentalmente para la fase de implantación. Esta característica es fundamental en UNIX, ya que su desarrollo ha sido por programadores y científicos dedicados al desarrollo de programas. Entre los principales colaboradores se tienen los grupos de cómputo de Berkeley, Stanford, MIT, Free Software Foundation, y muchas otras instituciones más que realizan herramientas y las ponen a disposición de otras personas.

El sistema operativo UNIX proporciona una serie de herramientas o programas de utilidad para el desarrollo de programas. Estas herramientas se muestran en la tabla 1.1.2.

VI	Es un editor de pantalla que tiene como misión principal la creación de archivos fuente.
CC	Es el compilador del lenguaje C.
LINT	Es un analizador sintáctico de lenguaje C más estricto que el compilador .
AR	Es un programa para mantener librerías o bibliotecas de programas objeto o cualquier otro tipo de archivos.
MAKE	Es un programa de utilidad de UNIX que acepta especificaciones de las dependencias existentes entre módulos de un programa y establecer mecanismos para mantener las versiones, trasladando al resto de los módulos los cambios que se realicen en uno de ellos.
SCCS	(Sistema de Control de Código Fuente) : Son un conjunto de utilidades para el desarrollo de aplicaciones con muchos módulos que se desarrollan en distintos equipos.
AUK	Es el intérprete de un lenguaje de programación cuyo fin es tabular, dar formato y preprocesar archivos para crear un analizador léxico que basa su análisis en ciertas reglas de lenguaje definidas por el usuario.
YACC	Es un analizador semántico cuya misión es transformar un archivo con instrucciones de un lenguaje especificado, que cumpla con las reglas semánticas dadas, en un módulo compatible en C.

Tabla I.1.2. Herramientas de desarrollo

UNIX y los estándares

UNIX es un sistema operativo que se ha implantado en diversos tipos de máquinas de varios constructores. Cada uno de estos constructores ofrece una versión diferente de UNIX. A pesar de tener el mismo origen todas estas versiones de UNIX, cada una es diferente. Así tenemos que un programa realizado para un tipo de UNIX no se ejecutará necesariamente en otro UNIX.

El sistema operativo UNIX ha sido modificado de diferentes formas por diferentes proveedores e instituciones. Ahora UNIX no es un sistema operativo que pueda ser utilizado indistintamente en una variedad de arquitecturas. Dado este gran número de versiones y diferentes tipos de máquinas, es necesario establecer estándares que permitan la comunicación entre todos los sistemas UNIX y aún más, entre todas las arquitecturas y ambientes de cómputo. La idea es establecer las bases para lograr que los productos de diferentes proveedores puedan trabajar juntos y sin tener que realizar modificación alguna.

En un ambiente de sistemas abiertos, los sistemas o Software de diferentes proveedores son intercambiables, es decir, son portables y se comunican entre sí. La portabilidad se refiere a la capacidad de utilizar el sistema operativo y las aplicaciones de software en una variedad de arquitecturas de diferentes proveedores.

La comunicación entre sí se refiere a la capacidad de intercambio de información entre diferentes arquitecturas. A este concepto se le conoce también como interconectividad o redes de computadoras.

UNIX no es completamente un sistema operativo abierto. Sin embargo, este sistema es una buena plataforma para la construcción de este ambiente de computación donde sea posible intercambiar código entre diferentes arquitecturas.

Ventajas de UNIX

El mundo de los sistemas operativos es muy vasto. Cada fabricante tiene para sus máquinas, el sistema operativo que explota las capacidades de la arquitectura lo mejor posible. Sin embargo, en este mundo de un sin fin de fabricantes, pero que requiere una conectividad de máquinas y también de aplicaciones, es necesario tener un esperanto de los sistemas operativos.

Otro factor importante a tomar en consideración es el hecho de tener en cada gran fabricante al menos una máquina que opere con el sistema operativo UNIX. Esta característica nos permite lograr una conectividad de un tipo de arquitectura con un sistema operativo 'A' con otra arquitectura con un sistema operativo 'B'.

La historia nos indica que UNIX no surgió como un proyecto de construcción de un sistema operativo comercial. Fue la iniciativa de un conjunto de investigadores y gente del mundo de la computación que crearon el proyecto, desarrollaron y mejoraron para sus propias necesidades, por esta razón, UNIX es el sistema operativo por excelencia del mundo de los desarrolladores de software.

Las ventajas que se obtienen con UNIX son:

- Mayor independencia para elegir proveedores de equipo, servicios y soluciones.
- Mayor independencia para elegir el mejor equipo de cómputo.
- Mejor nivel de integración de sistemas locales y remotos en redes.
- Mayor seguridad en el cómputo corporativo.
- Amplia capacidad de procesamiento remoto.
- Amplia capacidad de compartición de recursos físicos y lógicos.
- Aplicaciones portables de una plataforma a otra.
- Facilidad y menor costo para el desarrollo de aplicaciones por medio de API (Application Programming Interface).
- Paquetes de software y periféricos comerciales más abundantes y económicos.
- Menores costos y frecuencia de capacitación del personal.
- Facilidad de desarrollo de aplicaciones en un ambiente de red.

El sistema operativo UNIX es el ejemplo de la tendencia hacia una tecnología de sistemas abiertos con los cuales se piensa lograr un alto grado de compatibilidad con estándares de la industria, facilidad de comunicación entre procesos, transferencia de datos, redes de computadoras, estaciones de trabajo y aplicaciones portables de una plataforma a otra.

USL (UNIX System Laboratories), es el organismo dependiente de AT&T, el cual cuenta con alrededor de 350 investigadores. Estos están en colaboración estrecha con Bell Laboratories. El organismo USL propone la utilización de open look como la interfaz usuario gráfica y desarrolla gran parte del sistema UNIX System V Release IV, el sistema de transacciones TUXEDO entre otros proyectos.

Se tiene también para UNIX, X- Window que es un sistema de comunicación con la máquina que consiste de un sistema de ventanas en red que permite la construcción de redes de sistemas heterogéneos. El sistema X debe pensarse como un servicio de red que utiliza la tecnología cliente- servidor. Este servicio es parte de una interface de usuario. Hasta hace poco tiempo, la interface usuario del sistema UNIX, consistía sólo de una serie de líneas de mandato. Ahora, tenemos una interface usuario gráfica. Esta viene a sustituir el estilo de interacción con el sistema basado en líneas de mandato, por otro estilo basado en menús de opciones, iconos, y en la selección y manipulación de objetos.

I.2 SISTEMAS MULTIUSUARIO

Los sistemas *Multiusuario*, también conocidos como sistemas de *Tiempo-Compartido* (timesharing-system) aparecieron por primera vez a principios de los años 60's. En sus inicios estuvieron muy relacionados con las actividades del ejército y los sistemas de reservación de vuelos de las líneas aéreas.

La cuestión que dió origen a este tipo de sistemas fue la de que si una organización puede beneficiarse con el procesamiento computarizado para una o más aplicaciones pero no puede justificar el costo u otros requerimientos de una computadora separada para este uso solamente, la respuesta está en la compartición del uso de la computadora. De esta forma las aproximaciones para la compartición del uso del sistema fueron las siguientes:

- Tiempo Compartido
- Procesos Batch Remotos
- Servicio de Procesamiento Central
- Uso de Bloques de Tiempo

De las aproximaciones anteriores se hace referencia sólo a la primera, *Tiempo-Compartido*, debido a que es el concepto que está estrechamente relacionado a nuestro tema de tesis.

I.2.1 Definición

El término *tiempo-compartido* denota usualmente a un *Sistema Multiusuario* en donde todos los usuarios hacen uso de un sólo sistema de cómputo simultáneamente, teniendo cada uno de ellos la capacidad de definir y correr sus propios programas en la que consideran como su propia computadora de propósito general, operando independientemente sin estar enterados de lo que están haciendo los demás, siempre con la condición de que el sistema deberá responder con toda seguridad a sus requerimientos lo suficientemente rápido para estar a gusto con la interacción hombre-máquina. Los usuarios también podrán extraer o almacenar información desde o hacia la gran memoria conectada al sistema. Además el usuario podrá tener conectada una impresora de bajo costo a su terminal o podrá direccionar sus programas o salida de datos a una impresora de alta velocidad conectada directamente a la computadora central. Comparados con los "Sistemas Batch" y los "Sistemas de Multiprogramación" más comunes, los sistemas de tiempo-compartido han sido una mejor idea y una forma diferente de interactuar de las personas con las computadoras. Puesto que representan una manera más íntima para trabajar, actividades tales como la solución a problemas de diseño de ingeniería y todo tipo de programas de construcción y depuración son muy convenientes de desarrollar. Por otra parte una mejora importante en el ambiente creativo no sólo resultó en una economía de los recursos humanos, sino que a la vez la productividad aumentó debido al mejoramiento de la creatividad humana.

Los sistemas multiusuario son populares en las universidades y para ciertos tipos de aplicaciones científicas y de ingeniería, como es el caso del control de la maquinaria y de los procesos de fabricación aprovechando que la velocidad de las computadoras es mucho mayor que la de las máquinas, permitiendo de esta forma checar y ajustar

parámetros clave como son: presión, temperaturas, velocidades de motores, etc., en cuestión de segundos. En cuanto a la aplicación en las empresas, el servicio de los sistemas de tiempo compartido se enfoca básicamente en las tareas de rutina tales como la preparación de facturas, actualización e inventario de registros, control contable, etc., las personas desde su terminal pueden preestablecer procedimientos, y de esta manera los pasos del procesamiento serán estables y no cambiarán día con día; las formas deberán ser preparadas y los controles preestablecidos.

Los usuarios remotos de una computadora de tiempo-compartido están en una posición con respecto a la computadora central, misma que es un poco análoga a la relación que existe entre un subscriptor de un teléfono y la transmisión a la compañía de teléfonos que ofrece las facilidades para conectarse. El usuario del teléfono no necesita comprender la operación del intercambio por teléfono; todo lo que él necesita saber es como opera el suministro del teléfono. El usuario siempre espera que el servicio esté disponible a su demanda. Similarmente, el usuario de un sistema de tiempo-compartido primero deberá estar consciente de lo que pasa en su terminal, considerada por él, como su computadora de propósito general. Sin embargo, la comprensión por parte del usuario de que la respuesta en su terminal mejorará si él comprende cómo la computadora central facilita su accionar. Este conocimiento ayudará a entender mejor los controles necesarios en un sistema de tiempo-compartido.

Científicamente la computación orientada al tiempo-compartido está disponible desde los centros de servicio localizados en las grandes ciudades, y los usuarios desde algún lugar podrán conectarse a estos centros gracias a las facilidades de comunicación. En una típica aproximación, el usuario cuenta con una terminal de video conectada a la

computadora vía módem. El uso individual de este tipo de servicio con el cual podrá definir su propio problema, diseñar su procedimiento computacional, y en ambos usar un programa común de librerías estándar.

El tiempo-compartido de el procesador central del cual el sistema deriva este nombre genérico es sólo una de las muchas diferentes oportunidades para la compartición. Otras incluyen la compartición de la memoria principal, impresoras y otros dispositivos, la compartición de programas del sistema, programas traductores especializados como son los interpretes y los compiladores, librerías de subrutinas, y los programas de usuario. Esta accesibilidad disponible a los tiempos de reacción humana, insignificantes comparándolos con los de un sistema de tiempo-compartido convierte a estos en un inmenso medio de comunicaciones por el cual las personas pueden acceder rápidamente las experiencias de otras.

I.2.2 Características Generales

En cuanto a los rasgos característicos de la implementación de los sistemas de tiempo-compartido, un tema de discusión, el cual fue resuelto de una manera simple y sorpresiva, es que el equipo para un sistema de computo más o menos grande requiere relativamente poco en la forma de dispositivos adicionales o piezas para correr varios sistemas de tiempo-compartido. Estos sistemas se caracterizan por tres aspectos esenciales del CPU: un esquema de interrupción, un reloj de tiempo real, y un mecanismo de protección de la memoria, los cuales ya estaban presentes en las computadoras fabricadas en los años 60's, siendo necesarios para correr

eficientemente los modernos sistemas operativos de multiprogramación en batch como es el caso del poderoso sistema operativo UNIX.

La figura 1.2.1 muestra la configuración típica de un sistema multiusuario. De esta forma un sistema multiusuario estará conformado por cuatro elementos principales: 1) Una computadora central que puede ser un Mainframe o una Minicomputadora; 2) Una Memoria de gran capacidad para almacenar datos en masa; 3) Diferentes terminales de video que son conectadas a la computadora central por medio de cables o líneas telefónicas. La terminal podrá estar en el escritorio del usuario o si es posible hasta en su propia casa y 4) Impresoras locales de bajo costo e impresoras de alta velocidad.

Un arreglo típico de un sistema multiusuario tendrá las siguientes características:

1. Cada usuario tiene uno o más dispositivos de Entrada/Salida conectados a la computadora central a través de líneas de comunicación. El dispositivo de E/S esencial y algunas veces único característico de los ambientes de este tipo es la *terminal*.
2. Cada usuario actúa independientemente de los otros conectados al sistema. Cada quien envía datos e instrucciones desde su terminal hacia la computadora central. La razón a la cual un usuario introduce datos en el sistema es muy lenta comparada con la velocidad a la que la computadora puede procesar dichos datos, lo que hace parecer al usuario como si tuviera el uso exclusivo de la computadora.
3. La computadora central acepta los datos e instrucciones llegadas simultáneamente desde las terminales de los diferentes usuarios y proporciona a cada usuario un pequeño pero frecuente y repetido segmento de tiempo de

computadora, sirviendo a todos los usuarios concurrentemente. Para problemas mayores, la computadora puede enviar la salida requerida inmediatamente.

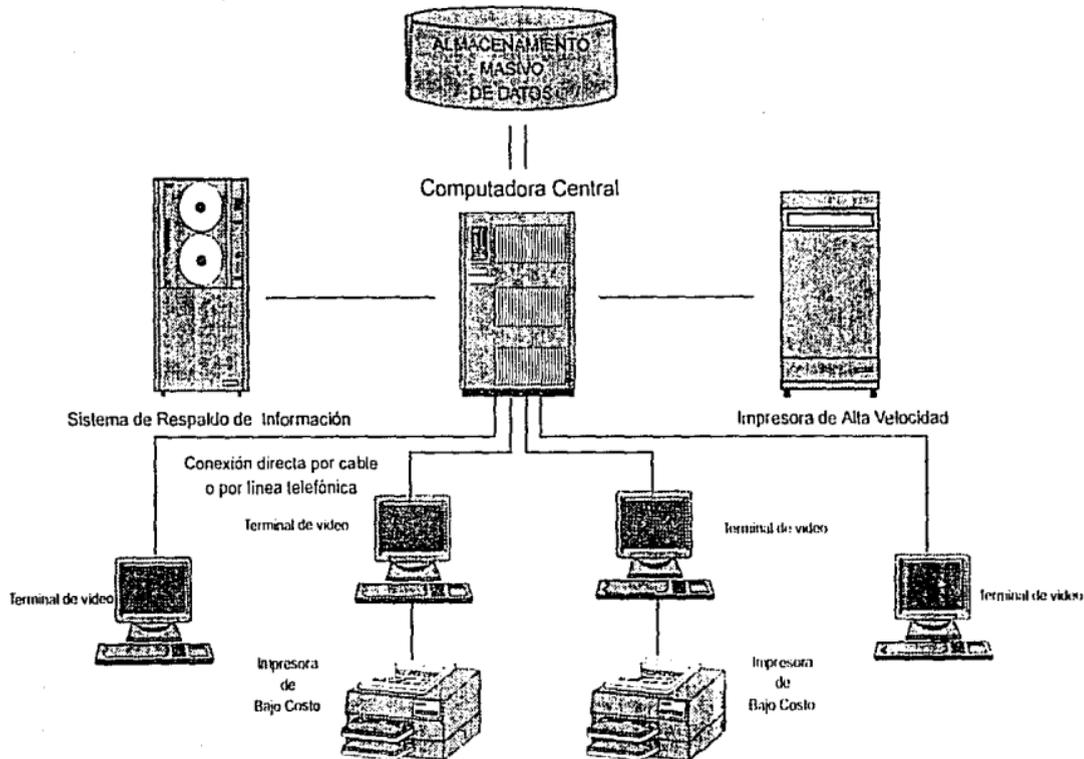


Figura I.2.1 Configuración básica de un sistema multiusuario

4. Los archivos de datos de los usuarios son mantenidos en la computadora central. Las instrucciones del usuario enviadas a la computadora identifican los archivos que serán usados. El sistema está diseñado para prevenir que un usuario haga accesos no autorizados a los archivos de los demás usuarios.
5. Cada usuario tiene su propio conjunto privado de programas para acceder al conjunto de programas de dominio público.
6. Los archivos de datos, archivos de programa, y dispositivos de Entrada/Salida están todos conectados directamente a la computadora, pero el procesamiento puede ser realizado aleatoriamente como transacciones ocurran o requisiciones sean hechas.

El tiempo-compartido puede ser visto como una extensión de la multiprogramación en la cual la computadora trabaja concurrentemente con diferentes programas, pasando de uno hacia otro con el propósito de maximizar el uso del hardware del sistema. En los sistemas de tiempo-compartido, el concepto adicional es la adición de servicio a todos la programas lo bastante frecuente como para que el usuario no se desaliente con el tiempo de respuesta. El acceso directo a la computadora por varios usuarios es la razón de que los sistemas de tiempo-compartido sean llamados también *Sistemas Multiusuario* o *Sistemas de Acceso Múltiple*.

Considerando lo que puede hacer la computadora dentro del ambiente de tiempo-compartido, se pueden mencionar los siguientes puntos:

1. Comunicar en intervalos irregulares con cierto número de dispositivos remotos.
2. Almacenar programas y datos en la memoria secundaria hasta que estos estén listos para ser ejecutados.
3. Decidir el orden en el cual los programas serán ejecutados.
4. Colocar el programa que será corrido próximamente y asociarlo con los datos en la memoria principal.
5. Correr el programa y transmitir los resultados a la terminal apropiada.

En la implementación de este arreglo de compartición existen dos problemas: compartir la Memoria Principal y compartir el Procesador Central. La compartición de la memoria principal es un problema, ya que esta representa un límite en el tamaño del almacenamiento puesto que todos los programas compartiendo el procesador no podrán estar cargados a la vez. Con lo que respecta a la compartición del procesador central, este presenta dificultades por que la computadora deberá intercambiarse entre programas, proporcionando a cada uno una parte de tiempo retardándolos notablemente.

Existen varios esquemas para compartir la memoria principal, los cuales pueden clasificarse en: 1) Paginación y 2) Swapeo de Memoria (Memory Swaping). En el esquema de paginación, ver figura 1.2.2 cada programa es partido en pequeños conjuntos de instrucciones llamados páginas. Estos deberán estar en un rango equitativo de un número pequeño de instrucciones hasta completar el segmento más grande de un programa. De esta forma cuando un programa está a punto de ser corrido, una sola página o un pequeño número de páginas son cargadas en la memoria y son ejecutadas. Para cualquier referencia de esta página con otras páginas en el programa será necesario cargar la página nueva a la memoria, y la página previa si no

va a ser utilizada por un tiempo considerable se descarga de la memoria principal y el espacio es liberado para otra página. Para poder llevar el control de que páginas serán cargadas y cuales descargadas es necesario que esté siempre cargado en la memoria el bloque de control principal que corresponde al programa. Páginas de distintos programas pueden estar en la memoria simultáneamente permitiendo así a la computadora atender a los diferentes programas.

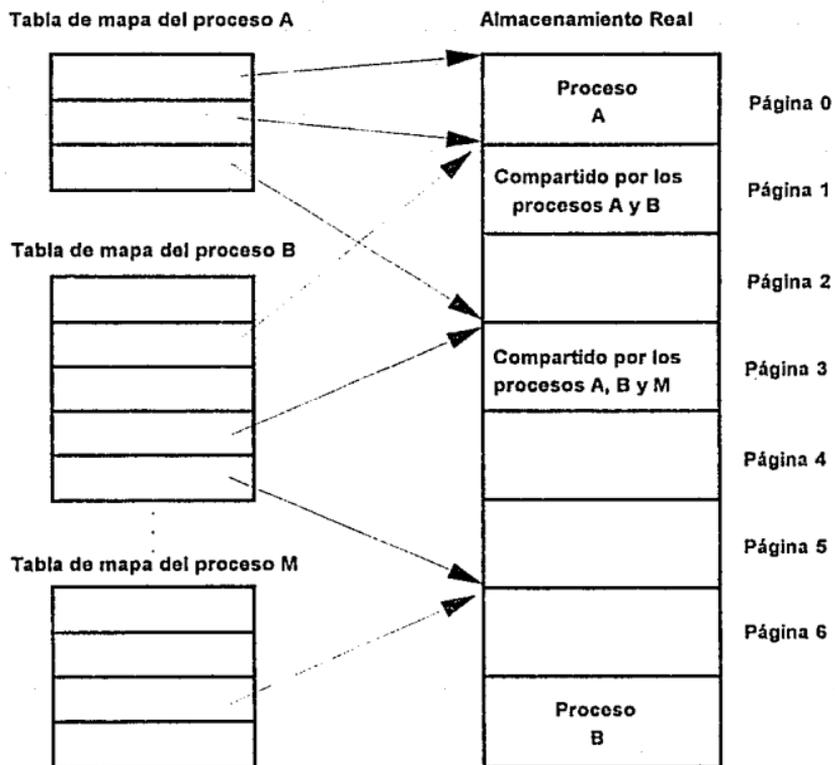
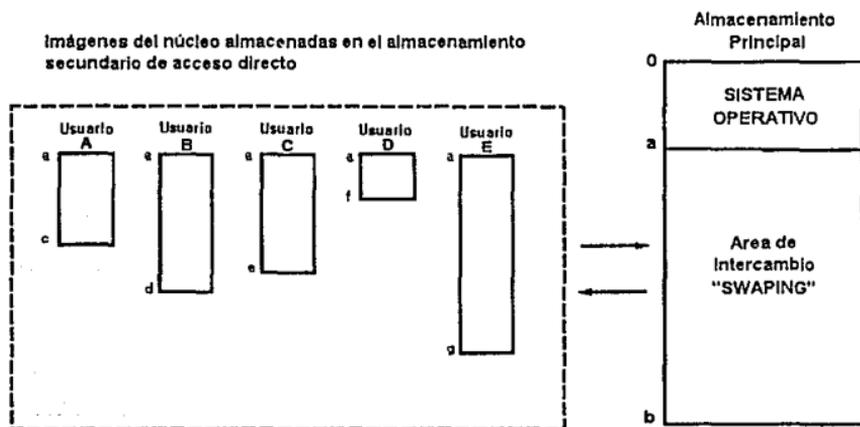


Figura I.2.2 Método de paginación para la compartición de la memoria en un sistema multiusuario

En el esquema de Memory-Swaping, ver figura 1.2.3, un programa entero es movido hacia o desde la memoria principal. Sólo un programa puede estar en la memoria a la vez, pero sólo se le permite un período de residencia en la memoria muy corto antes de rolarlo hacia afuera en favor del siguiente programa. Ambas alternativas requieren un sofisticado programa que supervise la operación y la tasa de transferencia a alta velocidad desde la memoria secundaria.



Sólo hay un usuario a la vez en el almacenamiento principal

Figura 1.2.3 Método de Swaping (Intercambio) para la compartición de la memoria en un Sistema Multiusuario

Si las páginas de varios programas están en la memoria al mismo tiempo, la computadora comparte el tiempo de procesamiento entre estos. Dos métodos pueden

ser utilizados para decidir cuando atender a cada uno. Un método es para atender al final un tiempo repartido muy corto. Cada programa agarra este tiempo repartido si este puede usarlo. Esto es muy usado en procesos científicos donde el tiempo para un problema es desconocido. El otro método es la atención cada vez que el programa sea ejecutado teniendo que esperar por un archivo de referencia de acceso directo o por una operación de Entrada/Salida. Un programa puede entonces ejecutarse y esperar varias veces antes de ser ejecutado por completo. Si el swapping de memoria es utilizado, cada programa recibe una pequeña parte del tiempo de proceso antes de ser movido desde la memoria para ser reemplazado por el siguiente programa. El programa espera su turno, repitiendo el ciclo hasta que este es finalizado.

Con lo que se refiere a los esquemas de control de los sistemas de tiempo-compartido, el servicio lo provee el procesamiento de la computadora y el almacenamiento para los diferentes usuarios., esto deberá ser conocido especialmente para establecer controles y protecciones tales como:

1. Protección contra alteración o destrucción de programas de usuario por otros programas también en la memoria.
2. Control contra accesos no autorizados a los datos de los usuarios o los programas propietarios.
3. Herramientas para la reconstrucción de archivos de datos que fueron destruidos.

Los métodos usados por los servicios de tiempo-compartido incluyen hardware de protección de memoria, identificación de códigos, una jerarquía para la preservación de archivos y controles internos de auditoría.

Si más de un programa está en la memoria principal de la computadora al mismo tiempo con la técnica de paginación, una protección de memoria es usada para prevenir que un programa sea introducido en el espacio de memoria destinado para otro. Esto es llevado a cabo a través de la adición de una llave de memoria para todas las instrucciones de cada programa. Si una instrucción de un programa intenta invadir el dominio de memoria de otro programa, la correspondiente llave de memoria prevendrá que la instrucción sea ejecutada.

Para prevenir accesos no autorizados para los archivos de datos y programas propietarios, un código debe ser asociado con todas las peticiones dentro del sistema, y el programa de supervisión del control comparará este código contra el código para que los archivos y programas sean examinados. Si el código de petición proporcionado a la computadora no es válido para los archivos, la requisición será descartada.

La posibilidad de que los programas y archivos de datos sean destruidos siempre estará latente. Por lo cual los sistemas de tiempo-compartido ofrecen las herramientas necesarias para copiar los datos y los archivos de programa hacia otro medio de almacenamiento secundario como son: las Cintas Magnéticas o los Dispositivos de Almacenamiento de Datos (DDS, Device Data Storage), desde las cuales la reconstrucción puede tomar lugar.

Por último, dentro de los aspectos de selección más importantes cuando se requiere evaluar un sistema-multiusuario podemos considerar los siguientes:

1. Tiempo de Respuesta.
2. Terminales
3. Lenguajes de Programación soportados
4. Método de Carga
5. Aspectos de Control

De esta forma, el usuario de un sistema de tiempo-compartido trabajando con una aplicación comercial también deberá estar preocupado acerca del nivel de asistencia que él puede recibir por parte del fabricante del sistema, la integridad financiera del servicio, si se trata de un negocio, y el control del procesamiento de datos para asegurar la rectitud y la terminación de los procesos.

1.2.3 Ventajas y Desventajas

Al hacer una evaluación de los Sistemas Multiusuario se encontró que ofrecen un número mayor de ventajas que de desventajas. Así, algunos consideran su costo como una desventaja debido a que el sistema estará constituido por diversos elementos como son: la computadora central, las terminales, que pueden ser Terminales Tontas o PC's, líneas de comunicación, impresoras, etc. Sin embargo, aunque el costo es mayor comparado al de otros sistemas, la unidad de tiempo compartido ofrece una nueva dimensión en el procesamiento de los datos, permitiendo el acceso continuo e inmediato a los archivos de la computadora, proporciona respuestas inmediatas a las peticiones realizadas, los programas pueden ser escritos y depurados sin esperar a que se les asigne un tiempo de compilación. De esta forma, un sistema de tiempo-compartido compensa su costo sirviendo a un número razonable de usuarios concurrentemente. Una computadora típica de tamaño mediano puede servir alrededor de 50 usuarios. Un número grande de usuarios es esencial para que sea viable económicamente y también técnicamente factible, tomando en cuenta que cada usuario individualmente es una gran carga para el sistema.

Otra de las ventajas del tiempo-compartido es que los problemas pequeños no justificarán un tropiezo para la computadora central o la disposición de tiempo requerido para arreglar el procesamiento que puede ser ejecutado económicamente en los sistemas de tiempo-compartido. La respuesta inmediata es especialmente atractiva para pequeños problemas de procesamiento los cuales son probables de originarse en intervalos irregulares y en casos donde hay un retardo mientras el problema es enviado y retornado desde el centro de servicio, es probable que signifique un retardo en el trabajo. ¿Pero que pasa si la computadora no está trabajando?, la respuesta a esta

cuestión es la paralización total del trabajo. También por otro lado, surge otro problema potencial de los sistemas de tiempo-compartido simples que es la saturación debido a que el número de usuarios crece constantemente lo que origina que el tiempo que la computadora le asigna a cada usuario vaya disminuyendo. Eventualmente el tiempo de respuesta de la computadora para cada usuario se convierte en una larga espera, por lo que mucha gente se queja al respecto.

Por otra parte, un sistema de tiempo-compartido puede asistir a los usuarios de una pequeña ciudad, y hasta personas situadas a una gran distancia, las cuales desde el centro de servicio pueden hacer uso de la computadora vía una terminal remota.

Así, los sistemas de tiempo-compartido a lo largo de su existencia han ido evolucionado rápidamente, desde la introducción de hardware especializado para tiempo-compartido, nuevos dispositivos de terminales de video y sobre todo nuevas y potentes facilidades de comunicación. De esta forma, las posibilidades inherentes de la compartición de un sistema a gran escala han hecho que los sistemas de tiempo-compartido sean parte fundamental de la estructura organizacional de las grandes instituciones.

I.3 INTRODUCCION A LAS BASES DE DATOS

La expresión Base de Datos comenzó a popularizarse al principio de la década de los 60s. Antes de esa época, en el mundo de la informática se hablaba de archivos y de conjuntos de datos. Como ocurre a menudo, hubo quienes quisieron subir de categoría sus archivos llamándolos bases de datos sin preocuparse de proporcionarles las características de no redundancia, independencia de datos, interconectividad, seguridad y accesibilidad en tiempo real. Estas características comenzaron a diseminarse al mismo tiempo que el empleo de un software más eficaz para la administración de datos. la base de datos puede definirse como una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que se usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados. Dicese que un sistema comprende una colección de datos cuando éstos son totalmente independientes desde el punto de vista estructural (James Martin, Computer Data-Base Organization).

En la figura I.3.1 se puede observar una vista simplificada de la base de datos.

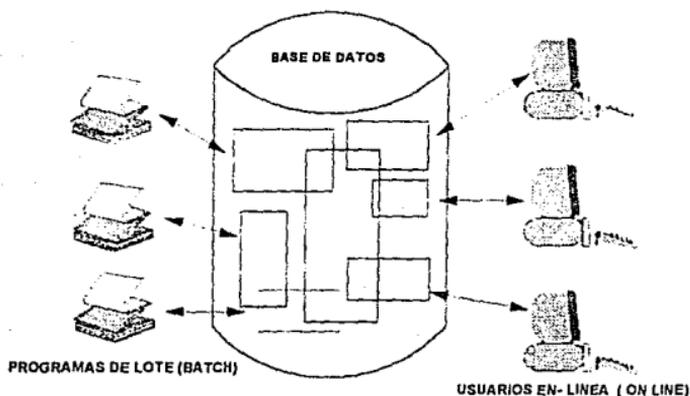


Figura I.3.1 Vista simplificada de la base de datos

Objetivos de la organización de las bases de datos

Las bases de datos pueden organizarse de muchas maneras. Los principios para la selección de técnicas de organización han sido estudiadas ampliamente. La comisión de sistemas CODASYL (Conference on Data Systems Language) ha emitido varios informes sobre el tema, y ha emitido un informe clásico sobre los requerimientos que debe satisfacer un sistema de administración de bases de datos. De los cuales a continuación se presenta un resumen de sus opiniones:

Los datos podrán utilizarse de múltiples maneras. Es decir, diferentes usuarios, perciben de manera distinta los mismos datos, y pueden emplearlos de diversas formas.

Se protegerá la inversión intelectual. No será necesario rehacer los programas y las estructuras lógicas existentes (que representan muchos hombres-año de trabajo) cuando se modifique la base de datos.

Bajo costo. Bajo costo del almacenamiento y el uso de los datos y minimización del costo de los cambios.

Menor proliferación de datos. Las necesidades de las nuevas aplicaciones serán satisfechas con los datos existentes en lugar de crear nuevos archivos, evitándose así la excesiva proliferación de datos que se advierte en las cintotecas actuales.

Desempeño. Los pedidos de datos se atenderán con la rapidez adecuada según el uso que de ellos habrá de hacerse.

Claridad. Los usuarios sabrán que datos se encuentran a su disposición y los comprenderán sin dificultad.

Facilidad de uso. Los usuarios tendrán fácil acceso a los datos. Las complejidades internas son ajenas al usuario, gracias al sistema de administración de la base.

Flexibilidad. Los datos podrán ser utilizados o explorados de manera flexible, con diferentes caminos de acceso.

Rápida atención de interrogantes no previstas. Los pedidos espontáneos de información se atenderán sin necesidad de escribir un programa de aplicación (lo que significa un cuello de botella por la pérdida de tiempo) sino utilizando un lenguaje de alto nivel para averiguación o generación de reportes.

Facilidad para el cambio. La base de datos puede crecer y variar sin interferir con las maneras establecidas de usar los datos.

Precisión y coherencia. Se utilizarán controles de precisión.

El sistema evitará las versiones múltiples de los mismos datos con diferentes estados de actualización.

Seguridad de acceso. Se evitará el acceso no autorizado de datos. Los mismos datos podrán estar sujetos a diferentes restricciones de acceso para distintos usuarios.

Integridad. Los datos estarán protegidos contra fallas, catástrofes, delincuentes, vándalos, incompetentes y personas que intenten falsificarlos.

Disponibilidad. Los datos se hallarán inmediatamente disponibles para los usuarios casi todas las veces que los necesiten.

Independencia física de los datos. El hardware de almacenamiento y las técnicas físicas de almacenamiento podrán ser alteradas sin obligar a la modificación de los programas de aplicación.

Independencia lógica de los datos. Podrán agregarse nuevos datos, o expandirse la estructura lógica general, sin que sea necesario reescribir los programas de aplicación existentes.

Redundancia controlada. Los datos serán almacenados una sola vez, excepto cuando existan razones técnicas o económicas que aconsejen el almacenamiento redundante.

Adecuada rapidez de acceso. Los mecanismos de acceso y los métodos de direccionamiento serán lo suficientemente rápidos, para satisfacer las necesidades de los usos previstos.

Normalización de los datos dentro de un organismo. Se necesita un acuerdo interdepartamental sobre los formatos y las definiciones de datos. La normalización es indispensable porque de otro modo se crearían datos incompatibles.

Diccionario de datos. Se necesita un diccionario de datos que defina las estructuras de la Base de Datos.

Interfaz de alto nivel con los programadores. Los programadores de aplicaciones deben disponer de medios sencillos para pedir datos y estar

aislados de las complejidades internas de organización y direccionamiento de los archivos.

Lenguaje de usuario final. Un lenguaje de consulta de alto nivel o un lenguaje para la generación de reportes permitirán que los usuarios finales se vean libres de tener que escribir un programa de aplicación convencional.

Controles de integridad. Siempre que sea posible, se recurrirá a revisiones de límites y otros controles para asegurar la exactitud de los datos.

Fácil recuperación en caso de falla. Recuperación automática sin pérdida de información.

Afinación. La base de datos debe ser afinable, para mejorar su desempeño sin exigir la reescritura de los programas de aplicación.

Ayudas para el diseño y la supervisión. Ayudas que permitan al diseñador o al administrador de la base datos a predecir y optimizar el desempeño del sistema.

Migración o reorganización automática. Migración de datos u otra reorganización física previstas para la mejora del desempeño.

I.3.1 Definición de una base de datos

¿Qué es una Base de Datos?

Una base de datos es un conjunto de datos de operación almacenados y utilizados por los sistemas de aplicación de una empresa específica.

Los datos de operación no incluyen datos de entrada o de salida, colas de espera de trabajo ni cualquier otra información de índole transitoria.

¿Qué es un sistema de base de datos?

Un sistema de base de datos, no es más que un sistema de mantenimiento de registros basados en computadores, es decir, un sistema cuyo propósito general es registrar y mantener información.

Tal información puede estar relacionada con cualquier cosa que sea significativa para la organización donde el sistema opera.

Este sistema de bases de datos comprende cuatro componentes principales: los datos, el hardware en el que residen, el software (llamado sistema de administración de bases de datos o DBMS) que controla el almacenamiento y recuperación de los datos y los propios usuarios.

En seguida se presenta un análisis breve de cada uno ellos:

DATOS:

Los datos almacenados en el sistema se dividen en una o más bases de datos. Desde el punto de vista didáctico es más conveniente suponer que sólo hay una base de datos, la cual contiene todos los datos almacenados en el sistema.

Una base de datos en general, es tanto integrada como compartida.

Por integrada se entiende que la base de datos puede considerarse como una unificación de varios archivos de datos independientes, donde se elimina parcial o totalmente cualquier redundancia entre los mismos.

Por compartida se entiende que partes individuales de la base de datos pueden compartirse entre varios usuarios distintos, en el sentido de que cada uno de ellos puede tener acceso a la misma parte de la base de datos (y utilizarla con propósitos diferentes).

HARDWARE:

El hardware se compone de los volúmenes de almacenamiento secundarios -discos, tambores, etc.-, donde reside la base de datos, junto con dispositivos asociados como las unidades de control, los canales, etc.

SOFTWARE:

Entre la base de datos física en sí y los usuarios del sistema existe un nivel de software, que a menudo recibe el nombre de sistema de administración de bases de datos o DBMS. Este maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios. Una función general del DBMS, es proteger a los usuarios de la base de datos contra los detalles a nivel de hardware. Es decir, el DBMS ofrece una vista de la base de datos que está por encima del nivel de hardware y apoya las operaciones del usuario que se expresan en términos de esa vista de nivel superior.

USUARIOS:

Se consideran tres clase generales de usuarios:

La primera la representa el programador de aplicaciones, encargado de escribir los programas de aplicación que utilicen bases de datos. Estos programas de aplicación operan con los datos de todas las maneras usuales: recuperan información, crean información nueva, suprimen o cambian información existente, etc.

La segunda es entonces, el usuario final que accesa la base de datos desde una terminal. Un usuario final puede emplear un lenguaje de consulta proporcionado como parte integral del sistema o recurrir a un programa de aplicación escrito por un usuario programador que acepte órdenes desde la terminal y a su vez formule solicitudes al DBMS en nombre del usuario final.

La tercera clase de usuario la representa el administrador de bases de datos o DBA.

ENTIDADES Y ATRIBUTOS

Llamaremos entidades a los objetos sobre las cuales se almacena información. Una entidad puede ser un objeto tangible, por ejemplo un empleado, pero también puede ser algo intangible, como un suceso o un concepto abstracto. Toda entidad tiene propiedades que eventualmente conviene registrar. A menudo, en el procesamiento de datos nos interesan las colecciones de entidades similares y necesitamos registrar información acerca de las mismas propiedades de cada una de ellas. A estas colecciones de objetos similares las llamaremos conjuntos de entidades.

Así como las entidades describen el mundo real, existen términos que se utilizan para describir la información acerca de ellas. Por lo general mantenemos un registro para cada entidad y agrupamos en conjuntos de registros de entidad todos los registros pertinentes a entidades similares. Los registros se refieren a atributos de las entidades y contienen los valores de estos atributos.

Cuando hablamos de información podemos referir a tres diferentes campos, y tendremos, a veces confusamente, a saltar de uno a otro sin advertencia previa. El primero de estos campos es el del mundo real, en el que hay entidades y estas exhiben ciertas propiedades. El segundo es el dominio de las ideas y la información existente en las mentes de las personas y los programadores. Aquí hablamos de los atributos de las entidades y nos referimos a éstos simbólicamente. Asignamos valores

a los atributos. El tercer grupo es el de los datos, en el que usamos caracteres o bits para codificar información.

Este grupo de entidades y atributos se puede ver representada de manera gráfica en la siguiente figura I.3.1.1

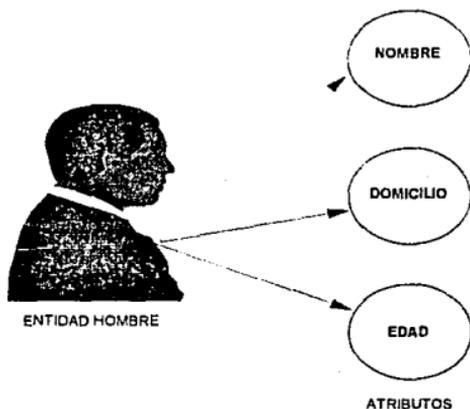


Figura I.3.1.1 Entidades y Atributos

La manera más común de asociar un valor con un dato y de asociar datos con atributos de entidad, consiste en almacenar juntos los datos en una secuencia fija, por ejemplo, en un registro de personal. En este podría usarse una codificación para representar los valores del atributo. Para esta entidad se indicará si el atributo es de longitud fija o de longitud variable. Los nombres de los atributos y las representaciones de valor no se registran en el archivo, aunque se deben encontrar registrados en

alguna parte, por ejemplo, en el diccionario de datos, en el que se listen los nombres y tipos de los diferentes datos de la base.

Podemos determinar entonces un registro de entidad a cada elemento y a los atributos que lo acompañan. A este tipo de distribución bidimensional de datos se le llama a menudo disposición plana.

Por otra parte un identificador de identidad es aquel atributo que es único a esa entidad y que sirve para reconocerlo, y en ocasiones se requiere más de un atributo para identificar un registro.

Una llave es el atributo o conjunto de estos que se utilizan para reconocer un registro. Una llave primaria es la que se utiliza para definir unívocamente un registro, es decir el identificador de entidad formado por uno o más atributos. También se usan llaves que no identifican registros únicos, sino todos aquellos que tienen cierta propiedad, y a estas se les llama llaves secundarias.

I.3.2 Elementos del sistema de bases de datos

Lenguaje de definición de datos

Un esquema de bases de datos se especifica por un conjunto de definiciones que son expresadas por un lenguaje especial llamado "Data Definition Lenguaje" (DDL). El resultado de la compilación del código de DDL es un conjunto de tablas que son almacenados en un archivo especial llamado diccionario de datos.

Un diccionario de datos es un archivo que contiene "metadatos", es decir "datos sobre los datos". Este archivo es consultado antes de que los datos reales sean leídos o modificados en el sistema de bases de datos.

La estructura de almacenamiento y métodos de acceso usados por el sistema de bases de datos son especificados por un conjunto de definiciones en un tipo especial de DDL llamado lenguaje de definición y almacenamiento. El resultado de la compilación de estas definiciones es un conjunto de instrucciones para especificar la implementación de detalles de los esquemas de bases de datos que están ocultos a los usuarios.

Lenguaje de manipulación de datos

Los niveles de abstracción se aplican también a la manipulación de los datos. Por manipulación de datos debemos entender:

- La recuperación de información almacenada en una base de datos.
- La inserción de información nueva en la base de datos.
- El borrado de la información de la base de datos.

A nivel físico debemos definir algoritmos que nos permitan el acceso eficiente a los datos. En un nivel de abstracción mayor, se da más énfasis a la facilidad de uso. El objetivo es proporcionar una interacción humana eficiente con el sistema.

Un sistema de manipulación de datos (DML, Data Manipulation Language) es un lenguaje que permite a los usuarios acceder o manipular datos organizados de acuerdo al modelo de datos apropiados. Existen dos tipos básicos de DML:

- **De procedimientos.** Este requiere que el usuario especifique que datos son necesarios y como obtenerlos.
- **De no procedimientos.** Este requiere que el usuario especifique qué datos son necesarios, sin especificar cómo obtenerlos.

Los DMLs de no procedimientos son generalmente más fáciles de aprender que los procedurales, sin embargo, como el usuario no tiene que señalar como obtener los datos, estos lenguajes pueden generar código que no será tan eficiente como el producido por los lenguajes procedurales. Esta dificultad puede ser remediada usando técnicas de optimización.

Un "query" es una declaración que indica la recuperación de información. La porción del DML que se ocupa de la recuperación de la información se llama lenguaje de

"query". Aunque es técnicamente incorrecto, comúnmente se usan los términos lenguaje de "query" y DML indistintamente.

Manejador de base de datos

Un manejador de Bases de Datos es un módulo del programa que proporciona la interfaz entre los datos de bajo nivel almacenados en la base de datos y en los programas de aplicación y consultas del sistema. El manejador de la base de datos es responsable de las siguientes tareas:

- Interacción con el manejador de archivos.

Los datos son almacenados en disco usando el sistema de archivos que es proporcionado por un sistema operativo convencional. El manejador de la base de datos traduce las instrucciones de DML en comandos de sistema de archivos de bajo nivel. Por lo tanto, el manejador es responsable del almacenamiento, recuperación y actualización de los datos en la base de datos.

- Garantizar la integridad.

El valor de los datos almacenados en la base de datos debe satisfacer ciertos requerimientos de consistencia. Estos deben ser especificados por el administrador de la base de datos y así, el manejador de la base de datos puede checar si las actualizaciones de la base de datos violan estas restricciones. tomando medidas necesarias.

- Garantizar la seguridad.

No todos los usuarios podrán acceder todos los datos, y esto es responsabilidad del manejador.

- Respaldo y recuperación.

Es responsabilidad del manejador detectar las fallas que pudieran afectar los datos y restaurar la base de datos al estado en que se encontraba antes de la ocurrencia de la falla. Esto se lleva a cabo mediante la iniciación de procedimientos de respaldo y recuperación.

- Control de concurrencia.

Cuando varios usuarios accedan la base de datos concurrentemente, la consistencia de los datos puede ser dañada. Es necesario que se controle la interacción entre los usuarios concurrentes y ésta es una de las responsabilidades del manejador.

Estructura general del sistema

Un sistema de base de datos se divide en módulos que se encargan de cada una de las funciones del sistema. Algunas de estas pueden ser proporcionadas por el sistema operativo de la computadora. En la mayoría de los casos, el sistema operativo provee

solamente los servicios básicos y el sistema de bases de datos debe ser construido sobre éste.

Un sistema de base de datos consiste en un número de componentes funcionales, los cuales incluyen:

- **Manejador de archivos**
- **Manejador de la base de datos**
- **Procesador de queries**
- **Precompilador de DML**
- **Compilador de DDL**

Adicionalmente se requieren también varias estructuras de datos como parte de la implementación del sistema físico, que incluyen:

- **Archivo de datos**
- **Diccionario de Datos**
- **Indices**

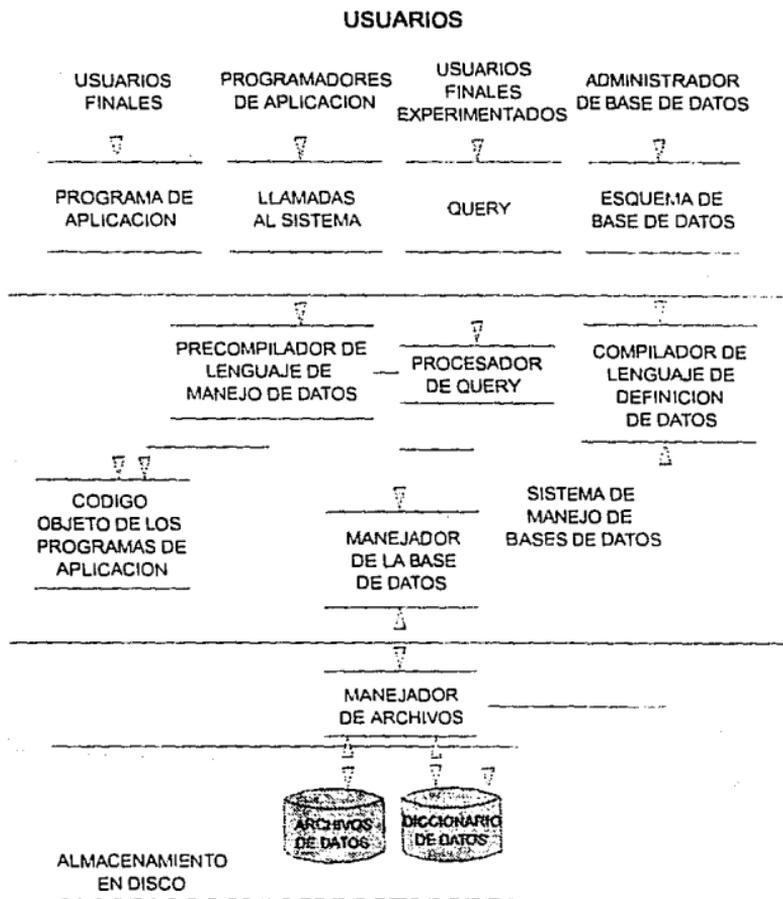


Figura 1.3.2 Estructura general del sistema de base de datos

1.3.3 Arquitectura de un sistema de base de datos

El propósito de describir esta arquitectura es proporcionar un marco de referencia. Este marco de referencia es muy útil para describir conceptos generales sobre bases de datos y para explicar la estructura de los sistemas individuales; pero no se pretende que cada sistema de bases de datos corresponda con fidelidad a este marco de referencia específico, ni se quiere dar a entender que esta arquitectura particular ofrece el único marco de referencia posible. Sin embargo, la arquitectura parece ser pertinente para el análisis de muchos sistemas; además coincide en gran medida con lo propuesto por el Grupo de Estudio de ANSI/SPARC sobre Sistemas de Administración de Bases de Datos.

La arquitectura se divide en tres niveles generales: interno, conceptual y externo. En términos generales, el nivel interno es el más cercano al almacenamiento físico, es decir, el que concierne a la manera como los datos se almacenan en la realidad; el nivel externo es el más cercano a los usuarios, es decir, el que atañe a la manera como cada usuario ve los datos; y el nivel conceptual es un *nivel de mediación* entre los otros dos. Si el nivel externo se relaciona con las vistas de los usuarios individuales, el nivel conceptual puede considerarse como el que define una vista de la comunidad de usuarios. En otras palabras, habrá muchas vistas externas, cada una compuesta por una representación más o menos abstracta de alguna parte de la base de datos, y habrá una sola vista conceptual, compuesta por una representación también abstracta de la base de datos en su totalidad. Así mismo habrá una sola vista interna, que representa la base de datos total tal como está almacenado.

En esta sección se hace una revisión a las principales arquitecturas de bases de datos y se muestra una figura con la cual se describe la arquitectura del sistema de base de datos (Figura I.3.3), que fueron definidas por grupos de expertos para tener un marco de referencia para la descripción de conceptos generales sobre las bases de datos. Después se presentan las características principales de algunos modelos de organización de datos que son: El modelo jerárquico o de árbol y el modelo reticular o de red, el modelo Entidad - Relación y el Modelo semántico, además nos ocuparemos del modelo relacional que será presentado más a detalle en la siguiente sección.

Niveles de arquitectura de una organización de datos

El manejo y organización de la información en una base de datos puede ser vista en 4 niveles de arquitectura que van desde el nivel de Organización Lógica hasta su correspondiente en el nivel de Organización Física, desde el más independiente del equipo hasta el más dependiente.

En el nivel 4, el más alto, es donde se hacen solo las consideraciones lógicas de datos, donde se hace el análisis de necesidades del usuario, de los datos con que cuenta y los que necesita, se especifica el flujo de la información y sus transformaciones. Posteriormente se definen entidades y objetos y sus relaciones funcionales. A partir de estas entidades y relaciones funcionales se construye o modela la base utilizando los modelos lógicos de organización, que serán explicados posteriormente. Cabe mencionar que este nivel es el más independiente del computador donde se va a implementar la base de datos.

El siguiente nivel, se orienta a la implementación de los modelos de organización, se abordan acciones orientadas al desempeño, se empiezan a tomar en cuenta las limitaciones de los dispositivos físicos. Aquí se definen las estrategias de acceso para mejorar el desempeño en el manejo de datos (consulta, actualización).

A los niveles 3 y 4 se les llama interfase física-lógica. El nivel 2 refleja alternativas de organización física y formatos de almacenamiento de datos en dispositivos de almacenamiento lineal para cada una de las organizaciones definidas en el nivel 3. Algunos autores solo reconocen 3 niveles y manejan como uno solo al 3 y al 2.

El nivel 1, es más dependiente que los anteriores del equipo donde se implementa la base de datos, se refiere a las características y aspectos particulares de los dispositivos físicos de almacenamiento. En este nivel se ven aspectos como: la organización de los dispositivos en subdivisiones tales como bloques y sectores; el mapeo de datos dentro de estas subdivisiones; el almacenamiento de áreas de sobreflujo y características dependientes de cada máquina, así como la localización de datos de control, verificación de integridad etc.

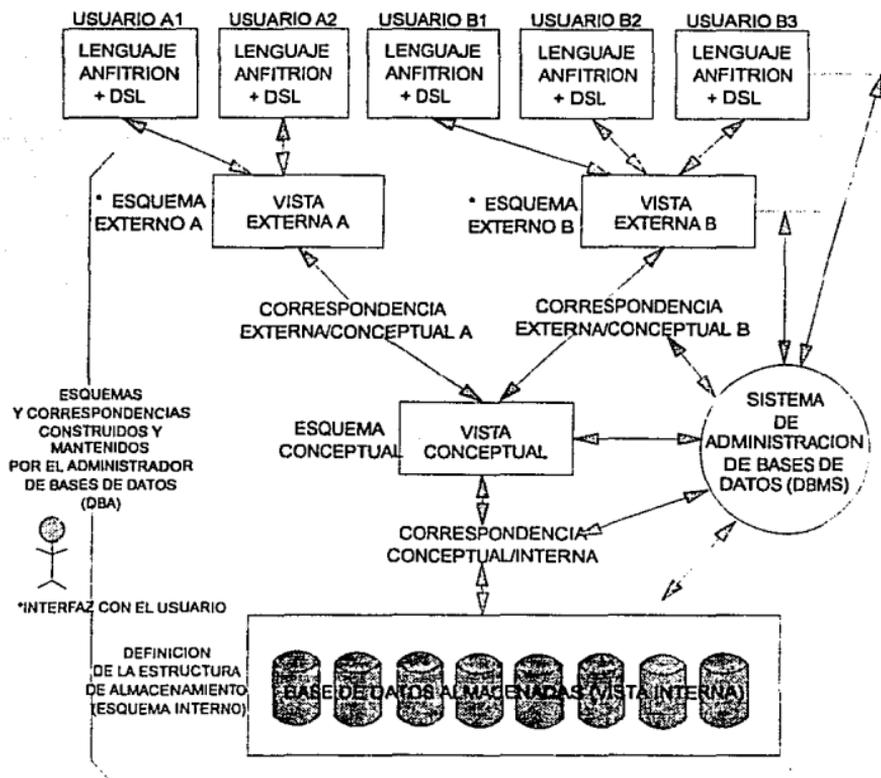


Figura I.3.3 Arquitectura del sistema de base de datos

1.3.4 Abstracción de datos

Un sistema manejador de base de datos es una colección de archivos interrelacionados y un conjunto de programas que permiten a varios usuarios acceder y modificar estos archivos. Uno de los principales fines de un sistema de bases de datos es proporcionar a los usuarios una representación simbólica de las piezas de información, a la cual llamaremos abstracción. Para que un sistema sea útil, los datos deben ser recuperados eficientemente. Con el fin de aumentar la eficiencia, se diseñan estructuras complejas de datos para representarlos en una base de datos. Sin embargo, como la mayoría de los usuarios de las bases de datos no son expertos en computación deben definirse varios niveles de abstracción desde los cuales se puede ver la base de datos, que son:

- Nivel Interno
- Nivel Conceptual
- Nivel Externo

Estos niveles fueron explicados en el punto anterior correspondiente al tema de Arquitecturas de Bases de Datos.

1.3.5 Bases de Datos Relacionales

Un sistema de Base de Datos debe ser capaz de representar y manipular entidades (registros o segmentos) y sus relaciones de manera fácil y conveniente. En el enfoque jerárquico de árbol, se representa la relación entre dos segmentos por la posición relativa de arriba hacia abajo y de derecha a izquierda de los tipos de segmentos involucrados. En el enfoque de red, las relaciones se representan mediante mecanismos de "set" con uso de apuntadores, los cuales enlazan a un tipo de registro propietario con un tipo de registro miembro. Para el caso de una Base de Datos grande y complicada, el modelo lógico y la manera como pueden tener acceso a él los usuarios, por medio de un lenguaje de manipulación de datos (DML) pueden volverse muy complejos. Aún más, el acceso a los datos se inclina demasiado a las rutas de acceso, en términos de los enlaces o posiciones jerárquicas que estableció el diseñador. De esta manera puede ocurrir que muchos cambios a la Base de Datos violen la independencia de los datos o afecten los programas de aplicación.

El enfoque relacional a Base de datos concebido por E.F. Cood, constituye un enfoque muy diferente para la descripción y manipulación "lógica" de los datos. Se esfuerza por evitar muchas de las desventajas que se han mencionado. En forma concisa, visualiza la Base de Datos lógica como una simple colección de tablas bidimensionales llamadas "Relaciones". Estas tablas son planas, en el sentido de que no hay grupos repetitivos. Los usuarios las comprenden y manejan fácilmente con poco o ningún entrenamiento en programación, y no implican consideración alguna sobre aspectos posicionables, de apuntadores o de rutas de acceso. Aparentemente es más fácil visualizar y manipular una tabla como la de la figura 1.3.5-a) que en su forma convencional equivalente dada en la figura 1.3.5-b), así mismo, es posible transformar

cualquier base de datos a una de tipo tabla plana o relacional, mediante la introducción de redundancia adicional.

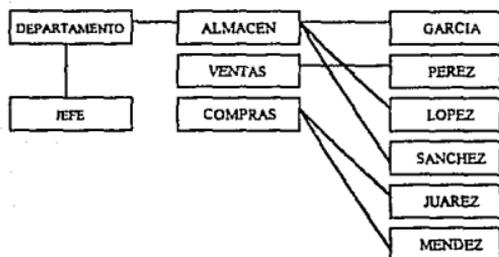


Figura 1.3.5-a Esquema visto por el usuario

DEPARTAMENTO	JEFE	ALMACEN	GARCIA
		ALMACEN	LOPEZ
		ALMACEN	SANCHEZ
		VENTAS	PEREZ
		COMPRAS	JUAREZ
		COMPRAS	MENDEZ

Figura 1.3.5-b Esquema tabular, relacional o plano

El enfoque relacional introduce terminología propia y exhibe una tendencia a usar términos poco convencionales relacionados con las matemáticas. El enfoque relacional se fundamenta en la teoría matemática de las relaciones, por lo cual, posee un buen fundamento teórico.

La figura I.3.5-c muestra la arquitectura relacional que se propone. La Base de Datos Global es un conjunto de relaciones, al que se hace referencia generalmente como modelo relacional de datos, relaciones base o base de datos relacional. El modelo de datos lo define el administrador de la base de datos (ABD) mediante un lenguaje de descripción del modelo relacional de datos.

Si el usuario se comunica con el modelo de datos mediante un sublenguaje de datos.

Descripción del modelo de datos del usuario mediante un lenguaje relacional de descripción de submodelo de datos (subesquema).

Descripción global del modelo de datos mediante un lenguaje relacional para descripción de modelos de datos (esquema).

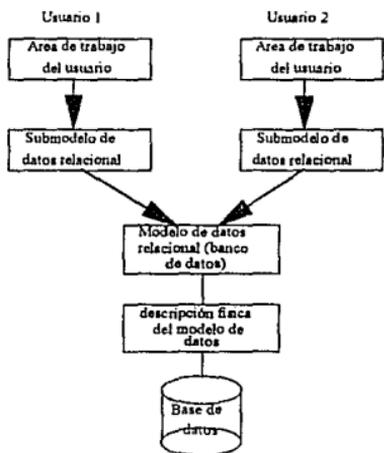


Figura I.3.5-c Arquitectura relacional propuesta

Al modelo particular de datos de un usuario, que se extrae del modelo de datos global, se le llama el submodelo relacional de datos o vista. Al submodelo de datos lo define el ABD y/o lo llama el usuario vía un lenguaje de descripción del submodelo relacional de datos. El submodelo de datos es una colección de relaciones que pueden derivarse de las del modelo de datos mediante ciertas operaciones relacionales. El lenguaje de

descripción del submodelo de datos incluirá las operaciones o mecanismos relacionales necesarios para formar submodelos permisibles a partir del modelo. De hecho, el lenguaje de descripción de submodelos es básicamente el mismo sublenguaje de datos, con ciertas adiciones para definición de datos.

Puede definirse un número arbitrario de submodelos sobre un modelo de datos dado. Un submodelo podría ser el modelo de datos completo. Un número arbitrario de usuarios puede compartir un modelo de datos dado vía los submodelos.

El modelo relacional

Desde una perspectiva histórica, el modelo de datos relacional es relativamente nuevo. Los primeros sistemas de base de datos fueron diseñados utilizando los modelos jerárquico y de red.

El modelo de datos relacional representa, como ya se mencionó, a la base de datos como una colección de tablas. Aunque las tablas son simples y de noción intuitiva, están en correspondencia directa entre el concepto de tabla y el concepto matemático de relación.

En años posteriores a la introducción del modelo relacional, se desarrolló una teoría para bases de datos relacionales. Esta teoría asiste en el diseño de bases de datos relacional y en el procesamiento eficiente de los requerimientos de información de los usuarios desde la base de datos.

Una base de datos relacional consiste en una colección de tablas, a cada una de las cuales se le asigna un nombre único. Cada tabla tiene una estructura similar a las

representadas en el modelo Entidad-Relación. Un renglón en una tabla representa una relación entre un conjunto de valores. Dado que una tabla es una colección de tales relaciones, encierra el concepto de tabla y el concepto matemático de relación, del cual el modelo de datos relacional toma su nombre.

En términos menos formales, una relación es una tabla bidimensional de "n" columnas constituidas por un conjunto de tuplos de "n" elementos (n-tuplos). Cada una de las columnas es una relación en un conjunto de valores de elementos de datos (tipo de atributo o campo) al que se le denomina el dominio.

Considere la siguiente tabla de depósitos:

Nom-Suc	Num-Cta	Nom-Clien	Saldo
Centro	101	Pérez	500
Coyoacán	215	García	700
Taxqueña	102	López	400
Valle	222	Ramos	750

Esta posee cuatro atributos: Nom-Suc, Num-Cta, Nom-Client y Saldo. Para cada atributo, hay un conjunto de valores permitidos, llamado dominio de un atributo. Por ejemplo para el atributo Nom-Suc, el dominio debe ser el conjunto de todos los nombres de sucursal. Denotemos a este conjunto con D1, D2 al conjunto de todos los números de cuenta (Num-Cta), D3 al conjunto de todos los nombres de clientes (Nom-Client) y D4 al conjunto de todos los saldos. Cada renglón debe consistir de 4 tuplos (v1,v2,v3,v4) donde v1 es el nombre de la sucursal (esto es, v1 está en el dominio de

D1), v2 es el número de cuenta (v2 está en el dominio D2), v3 es el nombre del cliente (v3 está en el dominio D3) y v4 es el saldo (v4 está en el dominio D4).

En el lenguaje formal de las matemáticas, dados "n" conjuntos D_1, D_2, \dots, D_n ; R es una relación sobre estos "n" conjuntos si R es un conjunto de n-tuplos cada uno de los cuales tiene su primer elemento en D_1 , su segundo elemento en D_2 , etc. Si la relación tiene "n" dominios ó columnas se dice que es de grado n. A las relaciones de grado 2 se les llama binarias, a las de grado 3 ternarias y a las de grado "n" se les denomina n-arias. El término tabla se refiere a una colección de tuplos de una relación dada.

Una relación o tabla es un arreglo bidimensional con las siguientes características:

1. Cada entrada en la tabla es un elemento de datos o dato elemental; no hay grupos repetitivos. Es decir, cada dominio debe representar a una sola relación. Se dice que una relación está normalizada si no tiene grupos repetitivos.
2. A cada columna, esto es, al dominio, se les asigna un nombre diferente y está constituido por valores del mismo dato elemental.
3. Todas las hileras o tuplos son distintas; no se permiten duplicados.
4. Las hileras y columnas pueden ordenarse en cualquier secuencia en cualquier momento, sin que esto afecte el contenido de la formación o la semántica aplicada.

Cada tuplo o relación debe poseer una llave que lo identifica unívocamente y lo diferencia de otros tuplos de esa relación. La llave es un dominio simple o una combinación de dominios. Una llave constituida por una combinación de dominios es no redundante si ninguna entidad de la llave puede eliminarse o borrarse sin destruir

la habilidad de identificar unívocamente a cada tuplo. Puede existir más de un conjunto de dominios que pueden constituir una llave; es decir que identifica unívocamente a un tuplo y que es no redundante. A estos conjuntos se les denomina llaves candidatas. La llave primaria es el conjunto de dominios que se selecciona para identificar a los tuplos. Normalmente debería ser el que tuviera el número mínimo de dominios.

Ventajas del modelo de datos relacional

La información es presentada al usuario final con un modelo de datos simple. Sus requerimientos están formulados en términos del contenido de la información y no refleja ninguna complejidad en los aspectos orientados al sistema. Un modelo de datos relacional es lo que el usuario ve, pero no necesariamente lo que físicamente se implementó.

Requerimientos de no procedimientos. Dado que no hay dependencia posicional entre las relaciones, no requiere reflejar alguna estructura preferida y por lo tanto puede ser de no procedimientos.

Independencia de Datos. El modelo de datos relacional elimina los detalles de estructura de almacenamiento y estrategia de acceso desde la interfaz de usuario. El modelo proporciona un grado relativamente grande de independencia de datos.

Desventajas del modelo relacional

Aunque algunos sistemas manejadores de base de datos basados en el modelo de datos relacional están disponibles actualmente, un DBMS relacional no ha alcanzado el nivel de desempeño de un modelo de datos jerárquico o de red. La pregunta que debe responderse está en relación a este aspecto. ¿Puede un modelo de datos relacional, usado por un DBMS, proporcionar el conjunto completo de capacidades operativas con la eficiencia requerida a gran escala?. Actualmente el adelanto de la tecnología está mejorando la rentabilidad y rapidez del hardware como respuesta a esta interrogante.

El proceso de normalización

El enfoque relacional posee bases matemáticas rigurosas que respaldan su teoría relacional, proporcionando simplicidad en las estructuras de datos utilizadas, facilitando su uso y modificaciones. Para poder obtener estas facilidades, el proceso de normalización es la clave.

Los objetivos del proceso de normalización son:

- Eliminar en lo posible todos los datos que mantengan anomalías
- Conservar toda la información
- Maximizar la flexibilidad

1. La estructura debe ser tal que haya lugar para todos los datos requeridos.
2. La redundancia que pueda existir deberá ser causada por los elementos que son identificadores o llaves. Por lo que se deberá tener cuidado de elegir aquellos que no estén sujetos a actualizaciones.
3. Los efectos indeseables son las anomalías que pueden presentarse en las operaciones de actualización, inserción y eliminación.

Anomalía de Inserción: No cabe almacenar nueva información sobre una entidad en particular hasta que se establece su relación con otra entidad.

Anomalía de Eliminación: La eliminación de un solo registro puede ocasionar la eliminación de toda una ocurrencia de una entidad.

Anomalía de Actualización: Si el valor de un atributo cambia, debe cambiar en los múltiples sitios donde se encuentre definido.

Esta capacidad de adaptabilidad de los cambios maximizan la independencia de uso particular de los datos.

La normalización requiere tres acciones sobre un atributo de una entidad. Estas son las siguientes:

- **Primera Forma Normal**
- **Segunda Forma Normal**
- **Tercera Forma Normal**

Estas se trataran detalladamente a continuación.

Dependencia funcional

El poder definir si una relación se encuentra en la primera, segunda o tercera forma normal, se basa en las dependencias funcionales que existan entre los atributos y los dominios particulares a esa relación. Las dependencias funcionales las determina directamente el significado o la semántica del contenido de la base de datos según la interpretación del diseñador de la base de datos.

Cuando se hizo la descripción del modelo E-R, se mencionó la existencia de una entidad independiente y otra subordinada. Esta subordinación está determinada por la relación que guardan entre ellas y se ejemplificó definiendo una relación en la cual el conjunto de entidad Transacciones dependía funcionalmente de Cuenta, ya que si la cuenta desaparecía, las transacciones que en ella se manejaban debían desaparecer, en cambio si se eliminaba alguna transacción la cuenta no se veía afectada. El modelo relacional se basa en este concepto para establecer sus relaciones funcionales entre atributos. De esta forma, una definición formal de dependencia funcional en el modelo relacional sería la siguiente:

Dada la relación R se dice que el atributo B es funcionalmente dependiente del atributo A si en cualquier instante de tiempo cada valor de A no tiene más de un valor B asociado con el en la relación R. El indicar que B es funcionalmente dependiente de A es equivalente a indicar que A identifica o determina a B, lo cual se denota como $A \Rightarrow B$. Esto último concuerda con la lógica matemática en la que, $A \Rightarrow B$ significa que A identifica a B, es decir, que si A tiene un cierto valor "a" entonces B debe tener un valor "b".

Primera forma normal

Uno de los objetivos del enfoque relacional es representar las bases de datos mediante relaciones planas o tablas. Por lo tanto cada identificador en una relación debe poseer un solo valor de cada uno de los atributos y no múltiples valores de estos. Si posee múltiples valores se dice que existen grupos repetitivos.

Se dice que una relación está en Primera Forma Normal si para cada valor específico de un identificador existe uno y sólo un valor de cada atributo. Es decir no hay "grupos repetitivos". Pero se mantiene un alto grado de redundancia.

Segunda forma normal

La prueba para determinar si una entidad está en Segunda Forma Normal es: que el valor de cualquier atributo que no es llave dependa de todos los atributos que forman la llave.

Para normalizar una entidad en la Segunda Forma, se crea una nueva entidad de los atributos que dependen parcialmente de una llave, siendo parte del identificador de esta nueva entidad el atributo del cual depende para mantenerla relacionada con la original.

Tercera forma normal

Se dice que una entidad se encuentra en Tercera Forma Normal si el valor de cada atributo depende de toda la llave y no de cualquier otro que no lo sea.

En esta forma normal, se buscan los atributos que están dependiendo de otro que no es una llave.

Para poner una entidad en Tercera Forma Normal, se crea una entidad con los atributos que no dependen de ningún atributo que forma la llave, siendo el identificador de la nueva entidad el atributo del cual era dependiente.

I.3.6 Bases de Datos en Ambiente Multiusuario

Para entender mejor lo que es un ambiente de bases de datos multiusuario podríamos imaginarnos que estamos escribiendo un libro en colaboración con otra persona, donde cada uno es responsable de hacer ciertos capítulos. Como resultado de este trabajo "concurrente", el trabajo avanza rápidamente. Trabajando de esta manera, podemos leer los capítulos del otro, pero no podemos editar o escribir en el capítulo que no nos corresponde, ya que de hacerlo así, el resultado sería un caos y un manuscrito que tal vez quedaría incompleto. Cuando necesitemos generar un índice, la otra persona debe dejar de escribir y darnos el control absoluto del manuscrito hasta que la generación del índice haya terminado.

Si comparamos este ejemplo con una base de datos multiusuario, el libro sería la base, los capítulos serían los registros, y nosotros los usuarios.

Manejo de las bases de datos multiusuarios

Para sistemas de Bases de Datos Multiusuarios los DBMSs proporcionan un balance entre dos objetivos conflictivos: la concurrencia y la integridad de los datos. Asegurar la integridad de los datos implicaría bloquear (lock) los archivos y los registros, lo cual reduce la habilidad de trabajar concurrentemente.

A diferencia de las bases de datos de un solo usuario, una aplicación multiusuario debe estar preparada para manejar las situaciones en las cuales un archivo no esté disponible cuando se le necesite, y en el cual pueda contener datos modificados la próxima vez que se le necesite. Al convertir las aplicaciones de un solo usuario a

aplicaciones multiusuario, el principal objetivo será el de maximizar la concurrencia tratando de mantener al mínimo el nivel de bloqueo (lock) de archivos que se requieran, y solamente bloquearlos mientras dure la operación.

Solamente el usuario que ha bloqueado un archivo puede escribir en él. Otros usuarios solo pueden leer los registros o archivos. Un archivo bloqueado puede ser accesado de nuevo dándole un comando para desbloquearlo (unlock). Este comando deberá liberar todos los bloqueos (locks) que se encuentren en el archivo o cuando el programa termine normalmente.

Procesamiento alternado

En el ambiente multiusuario, es común que varios usuarios necesiten usar un dato al mismo tiempo, por lo cual un programa multiusuario necesita saber que hacer cuando un usuario no puede obtener el lock deseado sobre un recurso. El programa podría seguir intentándolo hasta que se produzca una interrupción, o bien deberá reintentar un cierto número de veces antes de reportarlo al usuario. La aplicación debe incluir un procedimiento de error para manejar actividades de procesamiento alternas cuando un usuario no pueda acceder el archivo.

Control de concurrencia

Los sistemas multiusuarios bloquean (lock) los archivos y registros para prevenir que dos o más usuarios actualicen el mismo dato al mismo tiempo, lo cual causaría

actualizaciones perdidas. En una red típica, los pedidos de datos del usuario requieren que se copie el archivo de disco a la memoria de la estación de trabajo. Esto puede causar actualizaciones erróneas aún si los archivos tienen locks, porque varios usuarios pueden tener copias de los mismos datos en la memoria de sus estaciones de trabajo. Supongamos que dos usuarios tienen un archivo en sus estaciones. Un usuario modifica los datos y los vuelve a escribir a disco, pero el otro usuario puede actualizar la copia anterior de los datos que aún está en su memoria. Cuando este usuario lo escriba nuevamente a disco, éste va a reemplazar la actualización anterior. Algunos de los métodos utilizados para manejar las actualizaciones concurrentes son:

- La apertura de archivo exclusivos:

Este es el método más sencillo para controlar la concurrencia de actualizaciones y consiste en abrir los archivos involucrados en la actualización para uso exclusivo mientras dure la operación de lectura/escritura. Esto impide que otros usuarios puedan obtener copias de los archivos mientras estén ocurriendo las actualizaciones pero restringe el acceso de otros usuarios a archivos enteros por períodos largos e innecesarios. Por esta razón es altamente ineficiente.

- Marcar los archivos en uso (flagging):

Como parte de una aplicación se puede crear un archivo maestro que almacene en cada registro la llave primaria y una bandera que indique si un registro está en uso. Cada pedido de acceso (lectura o actualización) a un registro debe checar este archivo. Si un registro está en uso, se niega el acceso de otros usuarios a éste. Esto es más eficiente que el uso de archivos exclusivos, ya que

solamente pone lock a los registros restantes. Su inconveniente es la complejidad de programación, especialmente para una aplicación existente, porque cada pedido de acceso a un registro debe llamar primero a una rutina que revise la disponibilidad del registro. Si un registro está disponible, la bandera debe ser actualizada para indicar que está en uso, cuando se libera un registro, debe actualizarse nuevamente la bandera.

- Uso de semáforos:

Se puede implementar una función para controlar el acceso de registros basados en sus llaves primarias. Un semáforo es un mecanismo que asocia una etiqueta definida por el usuario (representando un recurso) con un contador que puede tomar un valor de 0 a 127. Se usa para controlar acceso a un recurso. Se definen operaciones de `OpenSemaphore`, `ExamineSemaphore`, `WaitOnSemaphore` y `CloseSemaphore`.

Cada semáforo tiene un contador abierto asociado a él. Un contador abierto es el número de aplicaciones que han usado la operación de `OpenSemaphore` y que todavía no han llamado la de `CloseSemaphore`. La información de contador de apertura tiene poco uso, ya que solamente indica el número de procesos que pueden pedir acceso a los recursos.

El valor del semáforo es el número de recursos disponibles. Este número es puesto inicialmente por el primer proceso que llama a `OpenSemaphore` para un recurso en particular. El valor del semáforo es mantenido por el sistema, y es el elemento crucial para controlar el acceso a un recurso. Cuando el valor se hace negativo, el proceso debe esperar a que se libere el registro, es decir debe

esperar a que uno o más recursos sean liberados con la instrucción `SignalSemaphore`.

Una aplicación que use semáforos para controlar el acceso concurrente debe llamar a `OpenSemaphore` con la llave primaria como etiqueta, y un contador inicial de 1. Debe llamar a `WaitOnSemaphore` cuando esté listo para acceder un registro; si el recurso no está disponible, la operación de `WaitOnSemaphore` pondrá al programa en cola hasta que el recurso esté disponible o a que expire el tiempo límite. Cuando una aplicación deje de usar un registro, debe llamar a `SignalSemaphore` para incrementar el contador y permitir a otro usuario acceder el recurso. Finalmente, la aplicación debe llamar a `CloseSemaphore` antes de terminar. Esto se ilustra en las figuras 1.3.6-a y 1.3.6-b.

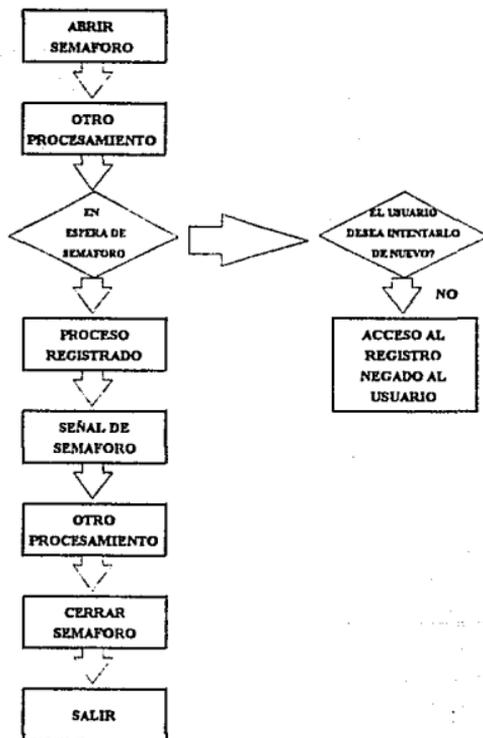


Figura 1.3.6-a Operación de uso de semáforos

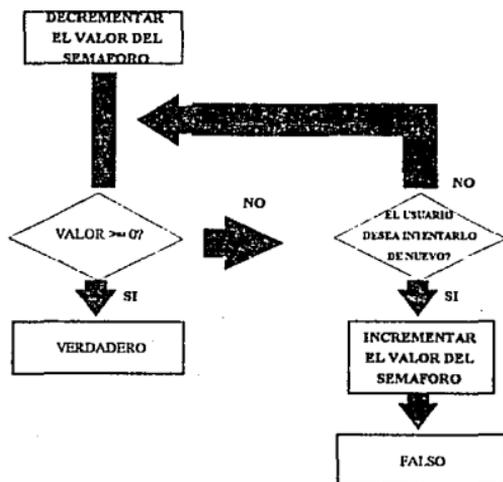


Figura 1.3.6-b Operación de Waitonsemaphore

- Comparar los valores de antes y después en arreglos:

Se pueden implementar procedimientos para verificar que un registro no haya sido modificado por otro usuario entre el momento en que el registro es llevado a la memoria de la estación de trabajo y el momento en que se actualiza. Esto es posible en productos que manejen arreglos.

En el momento en que el registro es leído (arreglo base), su contenido debe ser copiado en un arreglo (arreglo base), su contenido debe ser copiado en un arreglo (arreglo copia). Luego se copia el registro Base a un archivo temporal en el cual un usuario puede modificar los datos (arreglo de trabajo). Antes de escribir el registro nuevamente (arreglo del trabajo) se compara el arreglo copia con el arreglo base. Si hay diferencias, es porque el registro base se cambió entre el momento en que el usuario lo tomó y lo volvió a escribir. La edición que completó el último usuario queda desactualizada y necesita volverse a capturar. Este método se ilustra en la figura I.3.6-c.

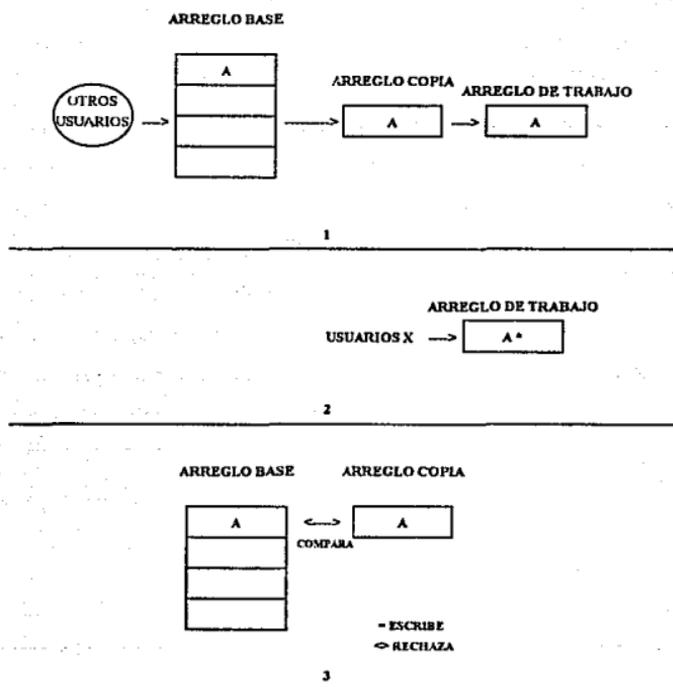


Figura 1.3.6-c Comparación de registros por medio de arreglos

Prevención de estancamientos (Deadlocks)

A menos de que se tenga cuidado, una aplicación multiusuario que lleva a cabo operaciones de lock de registros y archivos podría causar un problema de contención de recursos llamado estancamiento (deadlock).

Por ejemplo, tomemos las tablas, cuentas y clientes y los usuarios A y B. (Figura 1.3.6-a1). El usuario A tiene un lock sobre la cuenta XXX. El usuario B tiene lock sobre el cliente YYY. Si XXX es la cuenta del cliente YYY y A necesita al cliente YYY no podrá accederla porque B la tiene en un lock, y a su vez no soltará el lock sobre XXX porque no puede terminar su operación. Al mismo tiempo B tiene que acceder la cuenta de su cliente pero no puede hacerlo ya que A lo tiene en estado de lock, por lo tanto al no poder terminar su operación se quedara esperando a que A lo libere, pero A está esperando a que B libere el suyo.

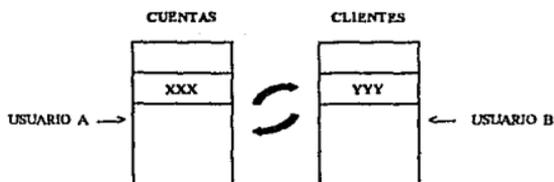


Figura 1.3.6-a1 Prevención de estancamientos (Deadlocks)

Esto ocurre cuando dos o más procesos están esperando poner lock a un recurso al que otro proceso ya le puso lock. Cuando esto pasa los procesos esperarán

indefinidamente a menos que el estancamiento sea detectado y resuelto por el sistema.

Algunas alternativas para prevenir estos estancamientos son:

- Abrir archivos en un orden preestablecido.

Una aplicación puede adoptar un estándar que pida que todos los archivos de la base sean abiertos y que se les ponga lock en un orden predeterminado, sin importar el orden en el que serán usados. Una desventaja de esta alternativa es que requiere que la aplicación abra los archivos en un orden que probablemente no vaya con sus necesidades. El diseño de la aplicación puede ser ineficiente debido al requerimiento de que los archivos sean puestos en lock de manera secuencial, además, algunos archivos pueden permanecer en lock por períodos prolongados si son abiertos mucho antes de que sean necesitados.

- Preasignación de archivos en áreas de trabajo específicas:

Esta opción también requiere que se creen estándares en la aplicación. Cada archivo debe ser abierto solo en áreas de trabajo preasignadas, las cuales deben ser seleccionadas en secuencias numéricas.

Esta alternativa tiene varias desventajas. En primer lugar no es apta para aplicaciones grandes que usen un gran número de archivos que excedan el número máximo de áreas de trabajo. En segundo lugar, el estándar puede ser más difícil de aplicar que la apertura de archivos en orden preestablecido. Cada programador debe tener acceso a una lista que indique el área de trabajo asignada a cada archivo en el sistema. Mantener este sistema puede ser un

verdadero problema cuando se tenga que cambiar las áreas preasignadas o cuando se agreguen nuevos archivos. Finalmente, este método evita los estancamientos cuando se trabaja con archivos, pero no con registros.

- Poner lock a los recursos necesarios desde la inicialización del programa:

Otra alternativa para evitar los estancamientos es hacer que cada programa de aplicación ponga lock a todos los archivos o registros que va a necesitar al comienzo de cada procedimiento. Si no se puede lograr el lock en todos los archivos/registros, el programa debe liberar los otros. Esto evita los estancamientos resultantes de poner lock a los archivos y registros porque un programa no esperará a que estén disponibles.

Esta solución obliga a la aplicación a saber que recursos va a necesitar, lo cual puede presentar problemas si el procesamiento varía dependiendo de los datos proporcionados por los usuarios. También hace el mantenimiento más difícil. Esta solución tiene el potencial de monopolizar recursos porque estos son puestos en lock mientras dure el proceso. Si se requieren muchos recursos, un programa puede esperar mucho para que todos los recursos estén disponibles simultáneamente.

- Implementar servicios de sincronización:

Este esquema se centra en una tabla que mantiene una lista de todos los registros a los que se va a poner lock. Una aplicación puede requerir que al conjunto de archivos contenidos en la tabla se les ponga lock, de modo que todo el conjunto de archivos quede con lock, o ninguno de ellos. La aplicación debe liberar los archivos antes de que termine. Este concepto es útil porque el conjunto de archivos puede ser liberado pero mantenido en la tabla para uso

posterior, esto se puede hacer físicamente especificando los registros en un archivo DOS con el offset de inicio y la longitud del registro. Los servicios de lock lógico presentan una solución más razonable, realizando un seguimiento de los locks lógicos colocados en los nombres de los registros. Esto coordina el acceso multiusuario a los datos compartidos sin poner locks físicos. Las aplicaciones necesitan un mayor grado de control interno para implementar efectivamente el lock lógico. La clave de esto es establecer un estándar para asignar nombres a los registros a los que se va a poner el lock.

Borrado de registros en una red

El borrado de registros en un sistema multiusuario presenta otros problemas. Cuando se borren registros, se debe planear empacar (pack) el archivo para remover los registros permanentemente, recapturando el espacio que ocupan. Como esta operación incluye el uso exclusivo del archivo, presenta un problema de concurrencia en un ambiente multiusuario en el cual los usuarios frecuentemente requieren acceso a varios archivos.

En el momento en que se está empacando un archivo, los usuarios no tienen acceso a éste, por lo tanto, se necesita una mejor alternativa. Una opción sería la de reciclar primero un registro, es decir ponerlo en blanco, y luego marcarlo como borrado. Cuando la aplicación necesite agregar un registro, puede "desborrar" el borrado y regresarlo a la aplicación como un registro nuevo. Cuando se acaben estos registros reciclados se necesitarán más registros nuevos. Lo que se recomienda en estos casos es anexas un lote de registros en blanco, y marcarlos como borrados, exceptuando los que se van a regresar a la aplicación.

Bases de datos distribuidas

Debido al crecimiento de los volúmenes de datos, al gran número de usuarios que accesan los datos, y a las mejoras en los medios de comunicaciones ha sido necesario distribuir las Bases de Datos en distintos puntos geográficos para facilitar su acceso y darle mayor rapidez a las operaciones que se realizan sobre ellas. Esto ha traído como consecuencia el concepto de las Bases de Datos Distribuidas.

En la figura I.3.6-a2 se observa un ambiente multiusuario en el que los datos se encuentran distribuidos, y la comunicación se hace a través de una red. Las características de los sistemas de Bases de Datos distribuidos multiusuarios son las siguientes:

- Los datos están dispersos en varios lugares.
- La localización geográfica de los datos es transparente para los programas que los utilizan.
- Cada nodo puede trabajar en forma autónoma.
- La integridad de los datos es controlada por el DBMS.

Para que una base de datos distribuida multiusuario cumpla con los objetivos de su creación, debe cumplir con las siguientes reglas, según Date.

- Autonomía local.
- No debe depender de un nodo centralizado.
- Debe operar continuamente.
- Debe ser independiente de su localización.
- Debe ser independiente de la fragmentación.

- Debe ser independiente en la generación de réplicas.
- Debe soportar el procesamiento de queries distribuido.
- Debe soportar el manejo de transacciones distribuido.
- Debe ser independiente del Hardware en que corra.
- Debe ser independiente del sistema operativo.
- Debe ser independiente de la red.
- Debe ser independiente del DBMS.

Hasta el momento, estas características no han sido implementadas en su totalidad y algunas de ellas se encuentran en fases experimentales.

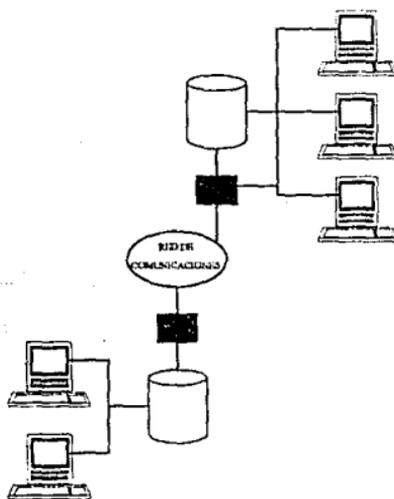


Figura I.3.6-a2 Ambiente multiusuario

1.3.7 Lenguaje estandar auxiliar en el desarrollo de los sistemas de Bases de Datos: SQL

El Lenguaje de Consultas Estructurado (SQL, por las siglas en inglés de Structured Query Language) ha sido el estándar para la descripción y el acceso de datos por casi una década, lo cual lo convierte en una tecnología esencial para el cómputo que sigue el esquema cliente/servidor. El único problema con el estándar de SQL es que existen varios de ellos.

No menos de ocho proyectos se están llevando a cabo para crear un estándar basado en SQL. Sólo el ANSI (Instituto de Estándares Nacionales Estadounidense) es responsable de tres estándares, ya sean publicados o en proceso.

El ANSI hizo el primer intento de crear un estándar de SQL en 1989 (SQL'89) y después siguió con una revisión en Octubre de 1992 (SQL'92). Pero como los estándares del ANSI eran ambiguos acerca de los detalles de la puesta en práctica, los productos que afirman apegarse a SQL a menudo no operan entre sí. La proliferación de distintos manejos de SQL por proveedores condujo a la formación de otros esfuerzos que involucran a consorcios de la industria, tales como SAG (el Grupo de Acceso a SQL) el cual fue formado para generar estándares de SQL basados en un subconjunto del SQL'89 del ANSI y para lidiar con los aspectos de conectividad y de la programación de interfases para clientes y servidores de SQL. En forma conjunta con el grupo X/Open, el SAG está construyendo un estándar basado en la implantación de esquemas cliente/servidor más avanzados actualmente.

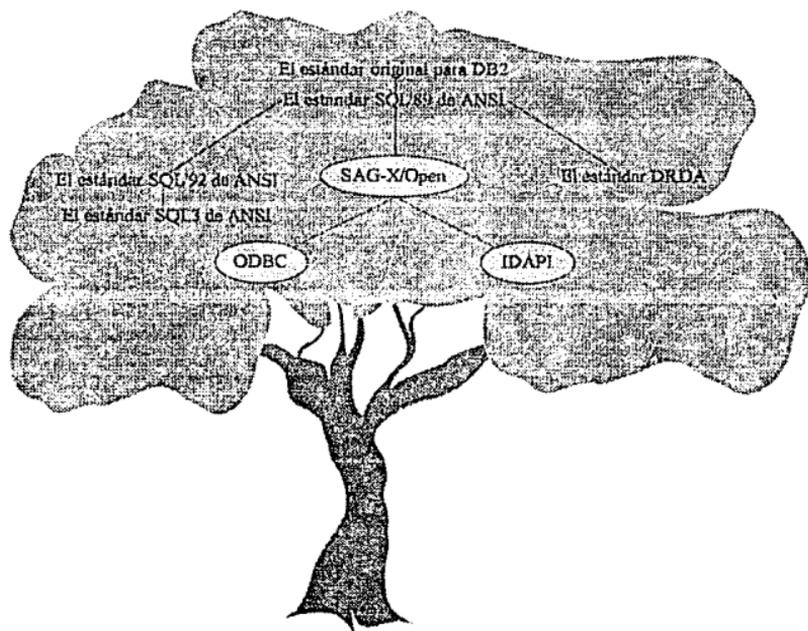
Muchas de las compañías mayores no están esperando a que fructifiquen los esfuerzos para establecer estándares, sino que están realizando sus propios intentos para crear estándares de facto, como es el caso de compañías como Microsoft (miembro de SAG) el cual presentó ODBC (Conectividad Abierta para Bases de Datos), un estándar para bases de datos SQL y API (Interfase para la Programación de Aplicaciones) basado en parte en un bosquejo del estándar del SAG. Borland, que también es miembro de SAG, está preparando su propio estándar de API SQL, llamado IDAPI (Interfase Integrada para la Programación de Aplicaciones de Bases de Datos), que está basada en la versión del SAG. Ambas compañías están promoviendo que otras apoyen sus estándares. Además, el ANSI está trabajando en la siguiente generación de su estándar de SQL, llamado SQL3. El SQL3 esperado para finales de esta década, el cual agregará al actual estándar SQL una mejor estructura, capacidad para bifurcación en el flujo de ejecución, control de flujo, manejo de errores y capacidades orientadas a objetos.

Finalmente compañías como IBM, está desarrollando su propio estándar para acceso a bases de datos, DRDA (Acceso Distribuido a Bases de Datos Relacionales) que incluye otra versión más de SQL.

Si además consideramos que cualquier base de datos "habla" su propio dialecto de SQL y tiene su propio conjunto de extensiones de este lenguaje, no debe sorprendernos que las instalaciones cliente/servidor constituyan la excepción y no la regla.

La falta de un estándar ampliamente aceptado eleva los costos de las bases de datos y otras herramientas relacionadas, y por otra parte, dificulta el mantenimiento a un ambiente cliente/servidor.

Todo lo mencionado anteriormente se puede apreciar de una mejor forma en la siguiente figura I.3.7.1 denominado Arbol Genealógico de SQL.



SQL es un lenguaje para acceso a datos diseñado para trabajar con bases de datos relacionales. Utilizado primero en el DB2 de IBM se convirtió en un estándar de facto a mediados de los ochenta.

SQL/89 de ANSI El estándar SQL del ANSI ratificado en 1989 fue el primer nivel de estandarización de SQL.

SQL/92 de ANSI Una extensión al estándar de 1989 que se convirtió en un estándar en los siguientes años, conforme los proveedores de bases de datos y de herramientas hicieron modificaciones para tener compatibilidad con él.

SQL3 de ANSI Un esfuerzo para expandir substancialmente el lenguaje, SQL3 proporciona más estructura y estándares para tener acceso a bases de datos orientadas a objetos. Estará disponible, posiblemente, a mediados de esta década

SAG (GRUPO DE ACCESO AL SQL) y X/Open, SAG es un consorcio de proveedores de base de datos y de herramientas. X/Open es un consorcio de la industria dedicado a publicar y reforzar el uso de estándares para sistemas abiertos. Estas dos agrupaciones están trabajando conjuntamente para establecer un estándar de SQL y una API basados en el SQL/89 de ANSI, y en los últimos avances en el ámbito cliente/servidor.

IDAPI (Interfase Integrada para la Programación de Aplicaciones de Bases de Datos), IDAPI está basada en el trabajo de SAG y de X/Open y refleja el intento de Borland por definir un API para SQL.

DRDA (Acceso Distribuido a Bases de Datos Relacionales), El estándar emergente de IBM para acceso a bases de datos a través de todas sus plataformas incluye otra incursión en SQL.

Figura 1.3.7.1 El Arbol Genealógico de SQL

¿Qué es SQL?

Las bases de datos relacionales almacenan datos en tablas (renglones y columnas) en forma similar a como lo hacen las hojas de cálculo. El lenguaje SQL incluye comando para describir, crear y borrar estas tablas, así como para actualizar los datos conservados en ellas. Por ejemplo, para extraer los nombres de los escritores de bases de datos que no cobren muy caro, se podría usar la siguiente instrucción:

```
SELECT Escritor
FROM Base_escritores
WHERE Pago_por_palabra=0.5
```

También se pueden insertar instrucciones de SQL dentro de otros lenguajes tales como C y COBOL.

Usando SQL, una computadora cliente puede tener acceso a datos en cualquier parte de una red. Los comandos de SQL no hacen referencia a la ubicación de los datos; los datos pueden ser llevados de una plataforma a otra sin afectar el código de la aplicación.

Teóricamente, SQL les permite a los proveedores de base de datos crear productos compatible con bases de datos y herramientas de otros proveedores. Casi todas las herramientas en el ámbito cliente/servidor esperan comunicarse con la base de datos servidora usando comandos de SQL. Estas herramientas pueden tener acceso a otras fuentes de datos también, si estas fuentes cuentan con una interfase compatible con SQL. Sin embargo, en la práctica, los numerosos sabores diferentes de SQL hacen

que la integración de diferentes herramientas de bases de datos sea, en el mejor de los casos difícil.

SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. El nombre "SQL" es una abreviatura de Structured Query Language (Lenguaje de Estructuras de Consulta). Por razones históricas, SQL se pronuncia generalmente "sequel", pero también se emplea la pronunciación alternativa "S.Q.L.". Como su nombre implica, SQL es un lenguaje de informática que se puede utilizar para interacciones con una base de datos. SQL trabaja con un tipo específico de base de datos, llamada base de datos relacional.

La figura I.3.7.2 muestra cómo funciona SQL. El sistema informático de la figura tiene una base de datos que almacena información importante. Si el sistema informático estuviere en una empresa comercial, la base de datos podría almacenar datos de inventario, producción, venta o nómina. En una computadora personal, la base de datos podría almacenar datos referentes a los cheques que usted haya firmado, lista de personas y sus números de teléfono, o datos extraídos de un sistema informático mayor. El programa informático que controla la base de datos se denomina *sistema de gestión de base de datos (data base management system)* o DBMS.



Figura I.3.7.2 Uso de SQL para acceso a la base de datos

Cuando se necesita recuperar datos de la base de datos, se utiliza el lenguaje SQL para efectuar la petición. El DBMS procesa la petición SQL, recupera los datos solicitados y los devuelve. Este proceso de solicitar datos de la base de datos y de recibir los resultados se denomina consulta (query) a la base de datos; de aquí el nombre Structured Query Language.

El nombre Structured Query Language es realmente y en cierta medida inapropiado. En primer lugar, SQL es mucho más que una herramienta de consulta, aunque ese fue su propósito original y recuperar datos sigue siendo una de sus funciones más importantes. SQL es útil para controlar todas las funciones que un DBMS proporciona a sus usuarios, incluyendo:

- **Definición de datos.** SQL permite a un usuario definir la estructura y organización de los datos almacenados y de las relaciones entre ellos.
- **Recuperación de datos.** SQL permite a un usuario o a un programa de aplicación recuperar los datos almacenados de la base de datos y utilizarlos.

- **Manipulación de datos.** SQL permite a un usuario o a un programa de aplicación actualizar la base de datos añadiendo nuevos datos, suprimiendo datos antiguos y modificando datos previamente almacenados.
- **Control de acceso.** SQL puede ser utilizado para restringir la capacidad de un usuario para recuperar, añadir y modificar datos, protegiendo así los datos almacenados frente a accesos no autorizados.
- **Compartición de datos.** SQL se utiliza para coordinar la compartición de datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros.
- **Integridad de datos.** SQL define restricciones de integridad en la base de datos, protegiéndola contra corrupciones debidas a actualizaciones inconsistentes o a fallos del sistema.

Por tanto SQL es un lenguaje completo de control e interacción con un sistema de gestión de base de datos.

En segundo lugar, SQL no es realmente un lenguaje informático completo tal como COBOL, FORTRAN o C. SQL no dispone de la sentencia IF para examinar condiciones, ni de la sentencia GOTO para bifurcaciones, ni de las sentencias DO o FOR par iteraciones. En vez de ello, SQL es un *sublenguaje* de base de datos, consistente en unas treinta sentencias especializadas para tareas de gestión de bases de datos. Estas sentencias SQL se *incorporan* a otro lenguaje, como COBOL, FORTRAN o C para extender ese lenguaje y permitirle utilizar el acceso a la base de datos.

Finalmente, SQL no es un lenguaje particularmente estructurado, especialmente cuando se compara con lenguajes altamente estructurados tales como C o Pascal. En vez de ello, las sentencias SQL se asemejan a frases en inglés completadas con "palabras de relleno" que no añaden nada al significado de la frase pero que hace que se lean más naturalmente. Hay unas cuantas inconsistencias en el lenguaje SQL, y también existen algunas reglas especiales para impedir la construcción de sentencias SQL que parecen perfectamente legales, pero que no tienen sentido.

A pesar de la imprecisión de su nombre, SQL ha emergido como el lenguaje estándar para la utilización de bases de datos relacionales. SQL es a la vez un potente lenguaje y un lenguaje relativamente fácil de aprender.

El papel de SQL

SQL no es en sí mismo un sistema de gestión de base de datos, ni un producto autónomo. Usted no puede ir a una tienda de informática y "comprar SQL". En su lugar, SQL es parte integral de un sistema de gestión de base de datos, un lenguaje y una herramienta para comunicarse con el DBMS. La figura I.3.7.3 muestra algunos de los componentes de un DBMS típico, y cómo SQL actúa como el "adhesivo" que los une.

La máquina de base de datos es el corazón de DBMS, responsable de estructurar, almacenar y recuperar realmente los datos en el disco. Acepta peticiones SQL procedentes de otros componentes DBMS, tales como una facilidad de formularios, un escritor de informes o una facilidad de consultas interactivas, desde los programas de aplicación escritos por el usuario e incluso desde otros sistemas informáticos. Como muestra la figura, SQL juega muchos papeles diferentes:

- SQL es un lenguaje de consultas interactivas. Los usuarios escriben órdenes SQL en un programa SQL interactivo para recuperar datos y mostrarlos en la pantalla, proporcionando una herramienta conveniente y fácil de utilizar para consultas de la base de datos.

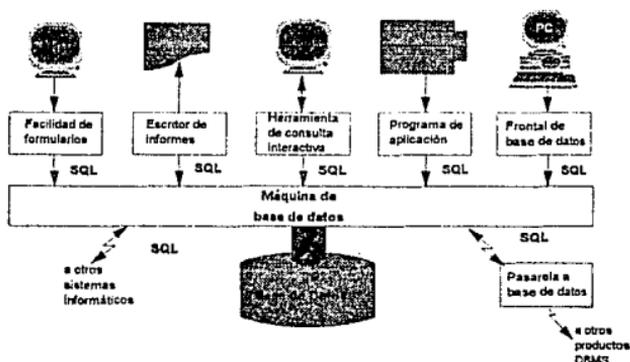


Figura I.3.7.3 Componentes de un sistema típico de gestión de base de datos

- SQL es un lenguaje de programación de base de datos. Los programadores insertan órdenes SQL en sus programas de aplicación para acceder a los datos de la base. Tanto los programas escritos por el usuario como los programas de utilidad de la base de datos (tales como los escritores de informe y las herramientas de entrada de datos) utilizan esta técnica para acceso a la base de datos.
- SQL es un lenguaje de administración de base de datos. El administrador de la base de datos responsable de gestionar una base de datos en un minicomputador

o en un maxicomputador utiliza SQL para definir la estructura de la base de datos y para controlar el acceso a los datos almacenados.

- SQL es un lenguaje cliente/servidor. Los programas de computadora personal utilizan SQL para comunicarse sobre una red de área local con servidores de base de datos que almacenan los datos compartidos. Muchas aplicaciones novedosas están utilizando esta arquitectura de cliente/servidor, que minimiza el tráfico por la red y permite que tanto los PC's como los servidores efectúen mejor su trabajo.
- SQL es un lenguaje de base de datos distribuidos. Los sistemas de gestión de base de datos distribuidos utilizan SQL para ayudar a distribuir datos a través de muchos sistemas informáticos conectados. El software DBMS de cada sistema utiliza SQL para comunicarse con los otros sistemas, enviando peticiones para acceso a datos.
- SQL es un lenguaje de pasarela de base de datos. En una red informática con una mezcla de diferentes productos DBMS, SQL se utiliza a menudo en una pasarela (gateway) que permite que nuestro producto de DBMS se comunique con otro producto.

SQL ha emergido por tanto como una herramienta potente y útil para enlazar personas y sistemas informáticos a los datos almacenados en una base de datos relacional.

Características y beneficios de SQL

SQL es a la vez un lenguaje fácil de entender y una herramienta completa para gestionar datos. He aquí algunas de las principales características de SQL y las fuerzas del mercado que le han hecho tener éxito:

- Su independencia de los vendedores.
- Su portabilidad a través de sistemas informáticos.
- Los estándares SQL.
- Su apoyo de IBM.
- Su fundamento relacional .
- Su estructura de alto nivel semejante al inglés.
- Las consultas interactivas .
- Su acceso a la base de datos mediante programas.
- Las vistas múltiples de base de datos .
- Su definición dinámica de datos.
- La arquitectura cliente/servidor.

Estas son las razones por las que SQL ha emergido como la herramienta estándar para gestionar datos en computadoras personales, minicomputadoras y maxicomputadoras.

Independencia de los vendedores

SQL es ofertado por todos los principales vendedores de DBMS, y ningún producto nuevo de base de datos puede tener éxito sin el soporte de SQL. Una base de datos basada en SQL y los programas que la utilizan pueden transferirse de un DBMS al DBMS de otro vendedor con mínimo esfuerzo de conversión y poco reentrenamiento del personal. Herramientas de base de datos basadas en SQL que funcionan con varios productos de DBMS, tales como escritores de informe y generadores de aplicación, están comenzando a aparecer. La independencia del vendedor proporcionada así por SQL es una de las razones más importantes de su popularidad.

Portabilidad a través de sistemas informáticos

Los vendedores de DBMS en SQL ofertan sus productos sobre sistemas informáticos que van desde computadoras personales y estaciones de trabajo hasta redes de área local, minicomputadoras y maxicomputadoras. Las aplicaciones basadas en SQL que comienzan en sistemas monousuario pueden ser transferidas a sistemas mayores de minicomputadoras o maxicomputadoras cuando crecen. Los datos procedentes de bases de datos corporativas basadas en SQL pueden ser extraídas y remitidas a bases de datos departamentales o personales. Finalmente, las computadoras personales pueden ser utilizados para construir prototipos de aplicaciones de base de datos basadas en SQL antes de transferirlas a un sistema multiusuario costoso.

Estándares de SQL

El American National Standards Institute (ANSI) y la International Standards Organization (ISO) han publicado conjuntamente un estándar oficial para SQL. SQL se ha convertido también en un estándar del U.S. Federal Information Processing Standard (FIPS), lo que le convierte en un requerimiento esencial para los grandes contratos informáticos del gobierno. En Europa, X/OPEN, un estándar para un entorno de aplicación por cable basado en UNIX, ha añadido SQL como el estándar para acceso a base de datos. La Open Software Foundation (OSF), un grupo de vendedores de UNIX, planea basar su estándar de acceso a base de datos en SQL. Estos estándares sirven como sello oficial de aprobación para SQL y han acelerado su aceptación en el mercado.

Apoyo de IBM

SQL fue inventado originalmente por investigadores de IBM, y desde entonces se ha convertido en un producto estratégico para IBM. SQL es un componente esencial de la Systems Application Architecture (SAA), la marca de IBM para la compatibilidad de sus diversas líneas de productos. El soporte SQL está disponible en todas las cuatro familias de sistemas incluidas bajo SAA: las computadoras personales PS/2, los sistemas de medio rango AS/400 y las maxicomputadoras de IBM que ejecutan los sistemas operativos MVS y VM. Este extenso soporte de IBM ha acelerado la aceptación del mercado de SQL, y proporcionado una clara señal de las intenciones de IBM para que otros vendedores de sistemas y bases de datos las sigan.

Fundamento relacional

SQL es un lenguaje para base de datos relacional, y se ha popularizado juntamente con el modelo de base de datos relacional. La estructura tabular, de filas y columnas de una base de datos relacional, es intuitiva para los usuarios, y hace que el lenguaje SQL se mantenga simple y fácil de entender. El modelo relacional tiene también un fuerte fundamento teórico que ha guiado la evolución y la implementación de las bases de datos relacionales.

Estructura de alto nivel en inglés

Las sentencias SQL parecen sencillas frases en inglés, lo que hace que SQL sea fácil de aprender y entender. Esto es en parte debido a que las sentencias de SQL describen los datos a recuperar, en lugar de especificar *cómo* hallar los datos. Las tablas y columnas de una base de datos SQL pueden tener nombres largos y descriptivos. Como consecuencia, la mayoría de las sentencias SQL "dicen lo que significan", y pueden ser leídas como frases claras y naturales.

Consultas interactivas

SQL es un lenguaje de consultas interactivas que proporciona a los usuarios acceso a los datos almacenados. Utilizando SQL interactivamente, un usuario puede obtener respuestas incluso a cuestiones complejas en minutos o segundos, en fuerte contraste con los días o semanas que le llevaría a un programador escribir un programa de

informe a medida. Debido a la potencia de consulta de SQL, los datos son más accesibles y pueden ser utilizados para ayudar a una organización a tomar decisiones mejores y más informadas.

Acceso a la base de datos mediante programas

SQL es también un lenguaje de base de datos utilizado por los programadores para escribir aplicaciones que acceden a una base de datos. Las mismas sentencias SQL se utilizan para acceso interactivo y programado, de modo que las partes de acceso a base de datos de un programa pueden ser comprobadas primero con SQL interactivo y luego insertadas dentro del programa. En contraste, las bases de datos tradicionales proporcionan un conjunto de herramientas para acceso mediante programas y una facilidad de consulta aparte para peticiones, sin ninguna sinergia entre los dos modos de acceso.

Vistas múltiples de datos

Utilizando SQL, el creador de una base de datos puede dar a diferentes usuarios de la base de datos vistas diferentes de su estructura y contenidos. Por ejemplo, la base de datos puede ser construida de modo que cada usuario sólo vea datos de su propio departamento o de su propia región de ventas. Además, los datos procedentes de diferentes partes de la base de datos pueden combinarse y presentarse al usuario como una simple fila/columna de una tabla. Las vistas de SQL pueden ser utilizadas

de este modo para mejorar la seguridad de una base de datos y acomodarla a las necesidades particulares de los usuarios individuales.

Lenguaje completo de base de datos

SQL fue inicialmente desarrollado como un lenguaje de consulta, pero su potencia va ahora más allá de la recuperación de datos. SQL proporciona un lenguaje completo y consistente para crear una base de datos, gestionar su seguridad, actualizar sus contenidos, recuperar los datos y compartirlos entre muchos usuarios concurrentes. Los conceptos de SQL que son aprendidos en una parte del lenguaje pueden ser aplicados a otras órdenes de SQL, haciendo más productivos a sus usuarios.

Definición dinámica de datos

Utilizando SQL, la estructura de una base de datos puede ser modificada y ampliada dinámicamente, incluso mientras los usuarios están accediendo a los contenidos de la base de datos. Este es un avance importante sobre los lenguajes de definición de datos estáticos, que impiden el acceso a la base de datos mientras su estructura está siendo modificada. SQL proporciona de este modo máxima flexibilidad, permitiendo que una base de datos se adapte a exigencias cambiantes mientras continúan sin ser interrumpidas las aplicaciones en línea.

Arquitectura cliente/servidor

SQL es un vehículo natural para implementar aplicaciones utilizando una arquitectura cliente/servidor distribuida. En este papel, SQL sirve como enlace entre los sistemas informáticos "frontales" (*front-end*) optimizados para interacción con el usuario y los sistemas "de apoyo" (*back-end*) especializados para gestión de base de datos, permitiendo que cada sistema rinda lo mejor posible. SQL también permite que las computadoras personales funcionen como frontales de bases de datos mayores dispuestas en minicomputadoras y maxicomputadoras, proporcionando acceso a datos corporativos desde aplicaciones informáticas personales.

I.4 DISEÑO DE BASES DE DATOS

El diseño de bases de datos se refiere al proceso de organizar los campos de datos necesarios para una o más aplicaciones, poniéndolos en una estructura organizada. Esta estructura debe contemplar las relaciones necesarias entre los campos cumpliendo al mismo tiempo con las restricciones físicas del sistema específico de manejo de bases de datos que se esté utilizando. Existen dos partes en el proceso: una es el diseño lógico de la base de datos y la otra es el diseño físico de la base de datos.

- El diseño lógico consta de dos componentes:

a) El primero significa organizar los campos de datos en agrupamientos no redundantes basados en las relaciones de los datos.

b) El segundo abarca una organización inicial de esos agrupamientos lógicos en estructuras basadas en la naturaleza del sistema de manejo de datos y de las aplicaciones que utilizarán los datos.

- El diseño físico depende de la implementación y toma los resultados del diseño lógico y lo refina de acuerdo a las características del sistema de manejo de datos que se esté utilizando.

Varias razones hacen que el diseño cuidadoso de la base de datos sea esencial; éstas incluyen la redundancia de datos, el desempeño en la aplicación, la independencia y seguridad de los datos y la facilidad de programación. Todos estos factores son importantes en el ambiente del procesamiento de datos, y todos pueden ser afectados cuando el diseño no es el apropiado.

Redundancia de datos

El diseño adecuado de la base de datos es esencial para evitar las redundancias tanto de un archivo como entre archivos. La cantidad de redundancia aparente de los datos (campos llave y no llave) varía de acuerdo al sistema manejador de datos (DBMS) para el que se están diseñando los archivos. En todos los casos, al menos en las primeras etapas del diseño, el objetivo es el eliminar toda la redundancia entre los campos. En etapas posteriores al diseño, conforme los datos toman forma de estructura del DBMS específico, puede o no existir redundancia abierta entre los campos.

Desempeño

El desempeño, velocidad operacional de las aplicaciones y sistemas, puede verse afectado o influenciado por un cierto número de factores en el medio ambiente de la base de datos y que están bajo el control de quien la diseña. Muchos de éstos son específicos del producto que se está utilizando; esto es, depende de las características de DBMS particular, pero otros tienen una naturaleza lo suficientemente general que pueden eliminarse al hacer un diseño cuidadoso de la base de datos.

Independencia de los datos

La independencia de los datos, o capacidad de modificar su estructura sin afectar programas ya existentes, es fundamentalmente una función del modelo de datos utilizado por el DBMS específico. Todos estos sistemas proporcionan un mejor grado de independencia de los datos al compararlos con el procesamiento normal de archivo. Más allá de los aspectos específicos de la estructura de los datos y del diseño de la base de datos, la forma en que se escribe un programa puede afectar la independencia de los datos.

Seguridad de los datos

Por lo general los sistemas de manejo de la base de datos tienen una variedad de opciones interconstruidas para la seguridad de los datos. Estas varían desde claves de acceso asociadas con un usuario específico o con datos particulares, hasta varias formas de prohibir que todos, excepto algunos usuarios, efectúen ciertas operaciones con determinados datos. La seguridad de los datos se transforma en un tema del diseño de la base de datos cuando la cantidad mínima de datos que el sistema es capaz de devolver tomándolos de la base en una llamada de programa es mayor que la cantidad que la persona que ejecutó el programa tiene derecho a ver. En ese caso, es posible que la estructura tenga que diseñarse de manera que esos datos, que de otra forma estarían colocados juntos en una unidad recuperable, estén divididos en varias unidades.

Facilidad de programación

Aunque el empleo de los sistemas de manejo de la base de datos requiere que los programadores aprendan nuevos conceptos y protocolos, el objetivo general del empleo de los DBMS es disminuir la complejidad de programación y las tasas de error esperado de programación.

Dentro del nivel de programación mejorado que permite el empleo de los DBMS, algunas decisiones en el diseño de la base de datos pueden en cierta medida dificultar o facilitar la tarea del programador. Desafortunadamente las decisiones que simplifican las especificaciones de programación tienen por lo general un efecto adverso sobre la redundancia de los datos, o en el desempeño del sistema o en ambos. .

1.4.1 Análisis de datos

La figura 1.4.1 muestra un flujo de sucesos para el análisis de una base de datos. Antes de que se pueda comenzar la evaluación de los requerimientos de una base de datos, se deben de comprender los objetivos y ámbitos globales del sistema para el que se va a desarrollar la base de datos. Posteriormente se desarrolla un modelo de información completo y a detalle.

El modelo de información incluye un diccionario de datos que define todos los elementos de datos en términos de la información que se usa para desarrollar el elemento.

Posteriormente, se deben de definir las características lógicas y físicas de la base de datos. Con el modelo de la información y la especificación del modelo como guía, se define la organización lógica de los datos. La organización lógica debe de considerar los requerimientos de acceso, modificación, asociación de los datos, así como otros aspectos orientados al sistema. Una vez que ha sido establecida la organización lógica de los datos, debe de desarrollarse la organización física. La organización física de una base de datos define la estructura de archivos, los formatos de los registros, las características de procesamientos dependientes del hardware y las características del DBMS. Finalmente, ha de realizarse una revisión completa del esquema y características físicas.

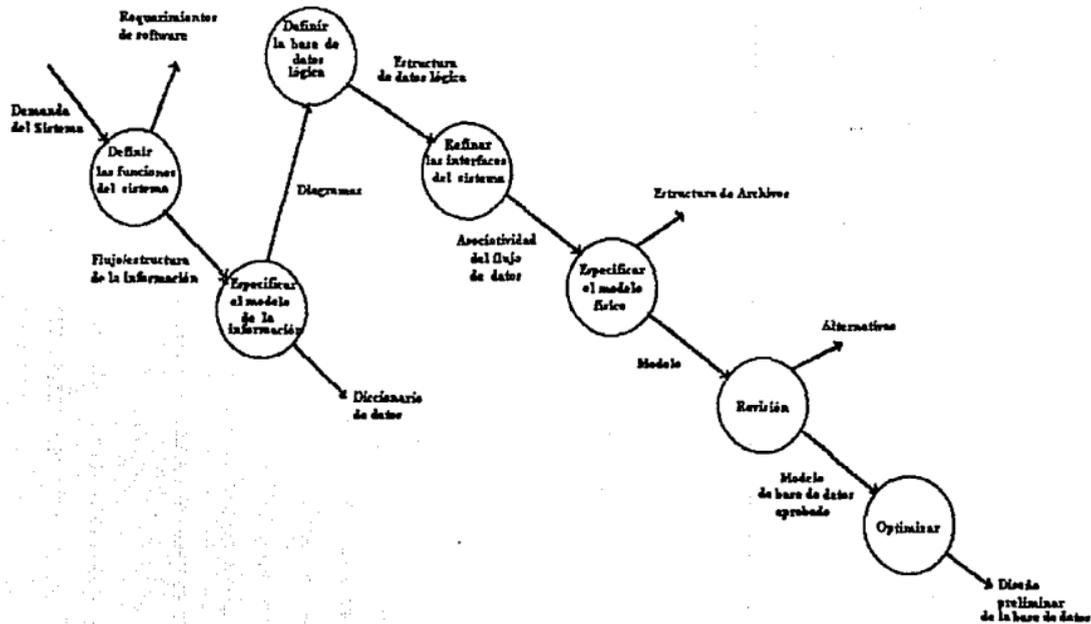


Figura I.4.1 Análisis de datos

I.4.2 Etapas del proceso de diseño

Elementos del proceso de diseño

Las principales clases de entradas y resultados del proceso de diseño son:

Entradas:

- Requerimientos de información general.
- Requerimientos de procesamiento.
- Especificaciones de DBMS.
- Configuración de hardware/sistema operativo.
- Especificaciones del programa de aplicación.

Resultados:

- Estructura lógica de la base de datos (vista del usuario)
- Estructura del diseño físico (almacenamiento)

Los requerimientos de información general representan las descripciones de varios usuarios de la organización para la cual los datos son reunidos, los objetivos de la base de datos , y las vistas de los usuarios de los cuales los datos deben de ser colectados y almacenados. Estos requerimientos son considerados como independientes de los procesos porque no están atados a ningún sistema de manejo de datos específico o aplicación. El diseño de bases de datos basado en estos requerimientos es considerado ventajoso a largo plazo para las bases de datos que deben ser adaptables a los requerimientos cambiantes del procesamiento.

Diseño de una base de datos

El diseño de una base de datos, envuelve un proceso de tres etapas:

1. Definición de los datos.
2. Refinamiento de los datos.
3. Establecimiento de relaciones entre los atributos.

Definición de datos

En la primera etapa, la definición de los datos se enlistan todos los atributos importantes involucrados en la aplicación. Para hacerlo, se debe de examinar la aplicación en detalle para determinar exactamente que tipo de información debe de ser guardada en la base de datos.

Un punto importante a tener en cuenta durante esta etapa del diseño, es que se deben de listar todos los posibles atributos para el sistema de gestión de base de datos. Se pueden listar más atributos de los que realmente se necesitan para la aplicación en particular, no es ningún problema, ya que los atributos realmente innecesarios serán eliminados durante la etapa de refinamiento de datos.

Refinamiento de datos

Durante esta etapa se refinan los atributos de la lista inicial, de forma que los atributos conformen una descripción exacta de los tipos de datos que se necesitarán en la base de datos. En este punto del proceso es vital considerar sugerencias de tantos usuarios de la base de datos como sea posible.

En otros casos particulares, algunos refinamientos pueden hacerse evidentes rápidamente y otros no tanto. Así que, volviendo a repasar la lista de atributos se irán haciendo más obvios algunos refinamientos necesarios.

Mientras se refina la lista de atributos, es conveniente calcular la cantidad de espacio en caracteres que se necesitará para mantener la mayor entrada posible de cada atributo en concreto.

Establecimiento de las relaciones

Durante la tercera etapa, la representación de las relaciones entre los atributos puede ayudar a determinar cuales son importantes y cuales lo son menos. Una manera de determinar las relaciones entre los atributos es cuestionar las mismas preguntas que se le plantearán a la base de datos.

Cuando se establecen las relaciones entre variables, también se puede descubrir que se necesita un atributo adicional, mismo que se deberá añadir al diseño.

I.4.3 Modelo Entidad-Relación

Modelos de datos

Para definir la estructura de una base de datos es necesario definir el concepto de modelo de datos. Un modelo es una representación de objetos y eventos del mundo real. Un modelo de datos es una representación abstracta (una descripción) de los datos por medio de sus entidades, eventos, actividades y sus asociaciones dentro de una organización.

Los modelos lógicos se utilizan para describir los datos en los niveles conceptual y de vista. Son flexibles y permiten especificar claramente las limitantes de los datos.

El modelo de entidad-relación es una técnica de las más comunes para el diseño de bases de datos, y requiere de una serie de pasos para producir una estructura que sea aceptable para el DBMS. Es el representativo de la clase de los modelos lógicos basados en objetos.

Beneficios de uso de la metodología de Entidad-Relación (ER)

Los enfoques convencionales para el diseño lógico de bases de datos normalmente sólo tienen una fase: convertir la información de los objetos del mundo real directamente al esquema del usuario. La metodología ER para el diseño lógico de bases de datos consiste en dos fases principales:

1. Definir el esquema empresarial usando diagramas entidad-relación y,
2. Traducir el esquema empresarial al esquema del usuario.

Las ventajas son:

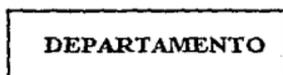
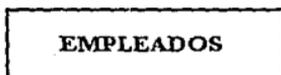
1. La división de las funciones y el trabajo en dos fases hace que el diseño de la base de datos sea más simple y más organizado.
2. El esquema empresarial es fácil de diseñar ya que no está restringido por las capacidades del DBMS y es independiente del almacenamiento y de consideraciones de eficiencia.
3. El esquema empresarial es más estable que el esquema del usuario. Si se desea cambiar de un DBMS a otro, el esquema empresarial será el mismo, ya que éste es independiente del DBMS que se use.
4. El esquema empresarial expresado por un diagrama entidad-relación es más fácil de entender por gente sin conocimientos de computación.

Conceptos en los que se basa la metodología de Entidad-Relación

Como se mencionó en el capítulo anterior, el modelo de datos ER se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados **entidades y relaciones**.

Entidades

Una entidad es una 'cosa' que se puede distinguir. Las entidades se pueden clasificar en diferentes tipos de entidades, como empleado y departamento. En un diagrama entidad-relación (DER), las entidades se representan con un rectángulo. Un sustantivo corresponde al nombre de la entidad en el DER.



Una entidad es el objeto principal del cual se tiene que almacenar información, normalmente denotando una persona, lugar, cosa o evento de interés.

Relaciones

Pueden existir relaciones (asociaciones) entre las entidades. Por ejemplo trabaja-en es una relación entre las entidades empleado y departamento.



En la notación de diagramas ER, una relación se representa con un rombo con líneas conectando las entidades relacionada normalmente con un verbo correspondiente a la relación. También se debe de especificar el tipo de asociación de las relaciones (uno a uno, uno a muchos, muchos a muchos).

Diseño del modelo Entidad-Relación

Este diseño requiere varios pasos para producir una estructura que sea aceptable para el DBMS. Los pasos del diseño del modelo entidad-relación son:

- Análisis de datos.
- Producir y optimizar el modelo de entidad.
- Desarrollo del esquema lógico.
- Proceso de diseño de la base de datos física.

Fase de análisis de datos

En esta fase se identifican los recursos de datos de la organización. El enfoque de análisis de datos y el énfasis que se le da a cada tarea depende mucho de los objetivos del proyecto. El análisis de datos se usa para :

- Determinar las fuentes de datos fundamentales de la organización.
- Permitir el diseño de estructuras de archivos flexibles capaces de soportar varias aplicaciones relacionadas.

- Ayudar al desarrollo de la aplicación o conversión, proporcionando un entendimiento fundamental de los datos involucrados.
- Formar las bases para el control de los datos, seguridad y auditoría de las aplicaciones y sistemas resultantes.
- Organizar los hechos relevantes relacionados con la organización de los datos.
- Ayudar a la unificación de la organización indicando las partes comunes entre sus departamentos y los requerimientos de datos.
- Proporcionar las bases para evaluar la capacidad de estructuración de los sistemas de manejo de bases de datos.
- Identificar las entidades que son relevantes para solucionar el problema de procesamiento de datos.
- Determinar las relaciones entre estas entidades.
- Establecer definiciones de procesos y datos en un diccionario de datos.
- Producir el modelo de entidad.

El objetivo primario del análisis de datos es el de proporcionar las bases para el diseño, se deben de identificar las entidades que son necesarias para resolver el problema definido por el usuario. Durante las etapas iniciales del análisis, es posible que no se conozcan todos los atributos de todas las entidades. Sin embargo, a medida que estos se determinen, se debe de documentar la definición del atributo y su papel en un diccionario de datos.

Elaboración del Modelo de Entidad

Durante la fase de análisis se determinan las entidades mayores y sus relaciones. Estas entidades y sus relaciones se representan en Modelos de Entidad, el cual representa la relación entre las clases de entidades.

La representación nos permite incluir solo aquellas entidades que se requieren para resolver un problema particular del procesamiento de datos. El modelo de entidad es esencialmente una vista del mundo real de los datos.

Durante la fase del modelaje se definen las clases y relaciones de entidad más significativas, sin embargo el modelo deberá ser revisado, modificado o extendido como resultado del conocimiento sobre las nuevas entidades que se descubran. El modelo se usa para:

- Reducir la redundancia en las relaciones.
- Determinar cuales entidades son significativas al modelo y a los requerimientos del usuario.
- Resolver las entidades no binarias entre entidades.

Etapas de la integración de los modelos de Entidad

Las etapas requeridas para integrar los modelos de entidad son:

- * Identificar cada sinónimo u homónimo en los diferentes modelos. Esta tarea es más fácil si se usa un diccionario de datos. Los componentes con homónimos

deben ser renombrados. Los componentes con sinónimos deben de usar el mismo nombre.

- * Los modelos de entidad para dos áreas de datos se integran superponiendo los tipos de entidad que sean idénticos o similares en los diferentes modelos de entidad. Esto puede incrementar el número total de atributos del tipo de entidad, ya que las entidades idénticas pueden usar diferentes atributos.

- * Como resultado de la integración, el modelo compuesto de entidad puede contener relaciones redundantes. Esta redundancia puede ser eliminada, sin embargo, determinar las relaciones que son directamente significativas y cuales son redundantes puede presentar dificultades que pueden ser resueltas solamente a través de un buen entendimiento del ambiente.

Combinación de modelos de Entidad

Al convertir bases de datos existentes en su equivalente de modelo de entidad, el diseñador puede llegar a diferentes modelos dependiendo de los programas o aplicaciones de los cuales los modelos fueron derivados. Se debe de intentar remover las redundancias e inconsistencias al combinar los modelos de varios programas para quedarse con un modelo integrado. Esta combinación permitirá al diseñador determinar lo siguiente:

- Cuales son las clases de entidad y los atributos comunes.

- Las inconsistencias en los nombres y uso de los atributos. Estas inconsistencias existen cuando dos entidades con diferentes nombres se ve que son la misma entidad.
- La eficacia del modelo en términos de satisfacer las necesidades del usuario.
- Si algunos atributos considerados en una entidad son realmente miembros de otra clase de entidad o de nuevas clases de entidad.
- La existencia de inconsistencias en las relaciones.

Este modelo combinado puede ser usado como la estructura para revisiones posteriores para llegar a un modelo de entidad integrado que sirva a un área de datos mayor, en lugar de varios modelos pequeños orientados a unas cuantas aplicaciones.

Agrupamiento de clases de Entidades

El agrupamiento de clases de entidades puede hacerse en la etapa de diseño físico o lógico. En la etapa del diseño físico esto puede hacerse basándose en consideraciones de desempeño. Las clases de entidades pueden juntarse o separarse en diferentes bases físicas dependiendo de los requerimientos de acceso.

El agrupamiento lógico de clases de entidades depende de la naturaleza de los datos y sus estructuras. Es necesario, pero no suficiente, decir que los atributos se agrupan en una clase de entidad, y éstas se agrupan en un modelo de entidad para satisfacer los requerimientos de procesamiento. El agrupamiento de clases de entidad se hace enteramente para satisfacer los siguientes requisitos:

- El área a que sirven los datos o de la cual se originan.
- La estructura de datos inherente.
- La vista del usuario.
- Los usos de los datos.
- Las consultas que se hacen de los datos.
- Las necesidades de procesamiento del usuario.

I.4.4 Diseño del esquema lógico

La vista de la aplicación puede definirse como el conjunto de datos que son requeridos por una aplicación particular para satisfacer una necesidad específica de procesamiento de datos. Tenemos vistas de aplicación de:

- Una clase de entidad.
- Agrupamientos de clases de entidades.
- Agrupamientos de clases de entidad y base de datos físicas.
- Agrupamiento de bases de datos físicas.

El esquema lógico puede ser definido como el mapeo del modelo de entidad en la construcción proporcionada por el manejador de la base de datos. En general el esquema lógico indica como se almacenará y accederá el modelo. En el diseño del esquema lógico tal vez sea necesario hacer algunos cambios al modelo para adecuarse al DBMS. El modelo de entidad no es el esquema lógico. El modelo de entidad tiene las siguientes características:

- Es una representación de la vista de datos del mundo real.
- Proporciona las bases para continuar con el análisis y diseño de la base de datos.
- No está restringido a ningún sistema manejador de bases de datos (DBMS).
- No es implementable directamente.
- Una estructura estable de referencia a la cual se puede agregar nuevas entidades, atributos y relaciones si la organización así lo requiere.

I.4.5 Conversión del esquema lógico a una base de datos física

Los detalles de esta fase dependen de las características del manejador escogido para el diseño. Esta transformación requiere las siguientes selecciones:

- Bases de datos físicas y tipos de relaciones lógicas, ya sea unidireccionales o bidireccionales, relacionadas físicamente.
- Métodos de acceso.
- Segmentos, estructuras jerárquicas y representaciones de datos, incluyendo tipo y tamaño.
- Índices secundarios.
- Tipos de apuntadores en la relación.

Adicionalmente de las selecciones anteriores, la implementación también incluye:

- Asignación de dispositivos de almacenamiento.
- Carga y organización de las bases de datos.

El esquema lógico debe de hacerse de manera que lo único que se deje a los diseñadores de la base física sea la selección de los métodos de acceso y los índices secundarios. Hubbard indica que se deben de seguir las siguientes reglas durante el diseño físico:

- Cada clase de entidad debe de ser tratada como una base de datos física.
- Si dos clases de entidades comparten una relación entre un atributo y la llave primaria por lo menos, entonces las estructuras deben de consistir en dos bases de datos físicas con conexión virtual o física entre ellas.
- Las relaciones padre-hijo deben ser definidas en una sola base de datos física.
- Los segmentos que se usen frecuentemente deben ser mantenidos lo más cercanamente posible a su raíz.
- Reducir el tiempo de búsqueda de grupos de datos grandes usando índices secundarios.
- Los segmentos de tamaños variables no deben ser colocados en el mismo grupo de datos si se hacen inserciones y borrados frecuentes.

1.5 SISTEMAS MANEJADORES DE BASES DE DATOS

Todos los sistemas de bases de datos están basados en una arquitectura (ANSI/SPARC), cuyo esquema simplificado se muestra en la figura 1.5.1. Si se observa con una perspectiva más amplia, podemos considerar que un sistema de base de datos posee una estructura muy sencilla compuesta por dos partes: una *Sección Posterior* y un conjunto de *Secciones Frontales*.

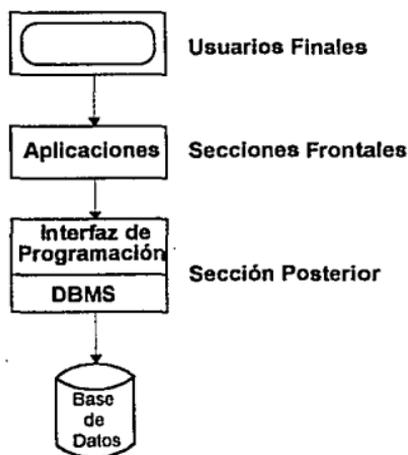


Figura 1.5.1 Configuración básica simplificada de un sistema de base de datos

De la figura I.5.1, se puede observar que la sección posterior la constituye en sí el Sistema Manejador de Bases de Datos cuyas siglas en Inglés son DBMS (Data Base Management System), las cuales de aquí en adelante usaremos para hacer referencia a dichos sistemas. De esta forma, la sección posterior permite llevar a cabo todas las funciones básicas de un DBMS, como son: la definición de datos, manipulación de los datos, seguridad, integridad, etc. Así, *sección posterior* es tan sólo un sobrenombre que se le dá al DBMS.

Por el otro lado, las *secciones frontales* son las diversas aplicaciones ejecutadas dentro del DBMS, tanto las escritas por los usuarios como las "integradas" que son las proporcionadas por el proveedor del DBMS o bien por otros proveedores de programas, no habiendo diferencia alguna entre ambas para la sección posterior.

En otras palabras, entre la base de datos física misma y los usuarios del sistema existe un nivel de programas conocidos comúnmente como el Sistema Manejador de Bases de Datos (DBMS). El DBMS maneja todas las solicitudes de acceso formuladas por los usuarios a la base de datos. Así una de las funciones generales de DBMS es distanciar a los usuarios de la base de datos de los detalles a nivel de equipo de manera muy similar a la forma como los sistemas de lenguajes de programación evitan a los programadores de aplicaciones la necesidad de ocuparse de detalles al nivel de lenguaje de máquina. De esta forma, el DBMS presenta a los usuarios una vista de la base de datos en un nivel un tanto por encima del nivel del equipo y hace posibles sus operaciones expresadas en términos de esa vista de nivel más alto.

El sistema manejador de la base de datos es por supuesto el conjunto de programas que maneja todo el acceso a la base de datos. Conceptualmente lo que sucede es lo siguiente:

1. Un usuario solicita acceso, empleando algún sublenguaje de datos determinado, por ejemplo SQL.
2. El DBMS interpreta esa solicitud y la analiza.
3. El DBMS inspecciona, en orden el esquema externo de ese usuario, la correspondencia externa/conceptual asociada, el esquema conceptual, la correspondencia conceptual/interno, y la definición de la estructura de almacenamiento.
4. El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenada.

La descripción anterior hace pensar que todo el proceso es interpretativo, pues sugiere que los procesos de analizar la solicitud, inspeccionar los diversos esquemas, etcetera, se realizan todos al momento de la ejecución. La interpretación, por su parte, casi siempre implica un desempeño pobre debido al aumento de tiempo de ejecución. Sin embargo, en la práctica puede ser posible compilar las solicitudes de acceso antes del momento de la ejecución.

Examinando las funciones del DBMS con un poco más de detalle, encontramos que dentro de dichas funciones se incluyen por lo menos las siguientes:

Definición de Datos

El DBMS debe ser capaz de aceptar definiciones de datos (esquemas externos, el esquema conceptual, el esquema interno, y todas las correspondencias asociadas) en versión fuente y convertirlas en la versión objeto apropiada. Dicho de otro modo, el DBMS debe incluir componentes procesadores de lenguaje para cada uno de los diversos Lenguajes de Definición de Datos (DDL, Data Definition Language). EL DBMS también debe entender las definiciones del DDL y debe poder utilizar estos conocimientos para interpretar y responder las solicitudes de los usuarios.

Manipulación de Datos

EL DBMS debe ser capaz de atender las solicitudes del usuario para extraer, y quizá poner al día datos que ya existen en la base de datos o para agregar en ella datos nuevos; en otras palabras, el DBMS debe incluir un componente procesador de Lenguaje de Manipulación de Datos (DML, Data Manipulation Language).

En general, las solicitudes en el DML pueden ser "planeadas" o "no planeadas":

Una solicitud planeada es aquella cuya necesidad se previó mucho tiempo antes de que tuviera que ejecutarse por primera vez. El administrador de la base de datos (DBA, Data Base Administrator) habrá afinado con toda probabilidad el diseño físico de la base de datos a fin de garantizar un buen desempeño para estas solicitudes.

Una solicitud no planeada, en cambio, es aquella cuya necesidad no se previó, sino que surgió de improviso. El diseño de la base de datos puede ser o no ideal para la solicitud específica de que se trate. En general el logro del mejor desempeño posible con solicitudes no planeadas representa un reto considerable para el DBMS.

Las solicitudes planeadas son características de las aplicaciones "operacionales" o de "producción"; las no planeadas son representativas de las aplicaciones de "apoyo a decisiones". De esta forma, las solicitudes planeadas casi siempre se originan en programas de aplicación previamente escritos, en tanto que las solicitudes no planeadas, por definición, se emitirán de manera interactiva.

Seguridad e Integridad de los Datos

El DBMS debe supervisar las solicitudes de los usuarios y rechazar los intentos de violar las medidas de seguridad e integridad definidas por el administrador de la base de datos (DBA).

Recuperación y Concurrencia de los Datos

El DBMS o en su defecto algún componente de software relacionado con él, al que por lo regular se denomina administrador de transacciones, debe cuidar del cumplimiento de ciertos controles de recuperación y concurrencia.

Diccionario de Datos

El DBMS debe incluir una función de Diccionario de datos. Puede decirse que el diccionario de datos es una base de datos por derecho propio, pero una base de datos del sistema, no del usuario. El contenido del diccionario puede considerarse como datos acerca de los datos, es decir, definiciones de otros objetos en el sistema, y no sólo de datos en bruto. En particular, en el diccionario de datos se almacenarán físicamente todos los diversos esquemas y correspondencias tanto de sus versiones fuente como en las versiones objeto. Un diccionario completo incluirá también referencias cruzadas para indicar cuáles programas usan cuáles partes de la base de datos, cuáles usuarios requieren cuáles informes, qué terminales están conectadas al sistema, y cosas por el estilo. Es más, el diccionario podría estar integrado a la base de datos a la cual define, e incluir por tanto su propia definición. Deberá ser posible consultar el diccionario igual que cualquier otra base de datos de modo que se pueda saber cuáles programas o usuarios podrían verse afectados por alguna modificación propuesta para el sistema.

Desempeño

Por último, el DBMS deberá ejecutar todas las funciones antes identificadas en la forma más eficiente posible.

Como conclusión, en forma de resumir todo lo anterior se puede decir que el DBMS constituye la interfaz entre el usuario y el sistema de la base de datos. La interfaz del usuario puede definirse como una frontera del sistema, más allá de la cual todo resulta

invisible para el usuario.

El DBMS es definitivamente el componente de software más importante de todo el sistema, pero no es el único. Entre los demás pueden mencionarse las utilerías, las herramientas para desarrollar aplicaciones, las ayudas para el diseño, los generadores de información ,etc. También es importante hacer notar que los DBMS's se clasifican de acuerdo al tipo de base de datos que manejan (Jerárquicas, Reticulares o Relacionales).

Una vez definido lo que es un sistema manejador de bases de datos y sus funciones principales, pasaremos a describir tres de los más importantes DBMS's que ofrece el mercado hoy en día a los usuarios de sistemas de bases de datos. El tipo de DBMS que describimos es el que maneja bases de datos relacionales (RDBMS ,Relational Data Base Management System), ya que es el modelo de base de datos más usado actualmente.

I.5.1 ORACLE7

La base de datos ORACLE es ampliamente usada y ocupa una posición importante en el ambiente de las bases de datos sobre el cual tiene una gran influencia. Junto con INGRES es pionera en la implementación de bases de datos relacionales, y encabeza el mercado en la promoción de software portable. Se puede decir que fue una de las primeras en adoptar al SQL (Structured Query Language) como un lenguaje estándar de acceso.

De ningún modo se puede decir que la portabilidad es el único aspecto de ORACLE que funciona como la fuerza conductora, especialmente con las bases de datos. ORACLE tiene la habilidad para trabajar en las más populares plataformas de hardware y explotar los diferentes ambientes tales como es el caso del hardware agrupado (VAX y RS/6000), máquinas SPM y masivamente hardware paralelo. Para ofrecer esto, ORACLE ha desarrollado una arquitectura de software flexible y escalable, la cual varía necesariamente entre cada máquina.

La versión 7 de ORACLE (ORACLE7) es un Sistema Manejador de Bases de Datos Relacional (RDBMS), escalable y completamente portable a más de 80 plataformas de hardware y sistemas operativos, desde sistemas de escritorio a mainframes y supercomputadoras. Este RDBMS es adaptable a una arquitectura de Servidor Multi-Hebras (Multi-Threaded), rindiendo un alto desempeño escalable a un gran número de usuarios en todas las arquitecturas de hardware, incluyendo multiprocesos simétricos. ORACLE7 ofrece control de concurrencia con completo y no restringido bloqueo a nivel de registro, además de contención libre de consultas. Este provee un costo basado en consultas optimizadas y en los recursos limitados. Las aplicaciones de ORACLE7

pueden correr rodeando el reloj, debido a que las funciones del sistema tales como: respaldo, recuperación, y la administración de la base de datos ocurren en línea sin interrumpir la transacción del procesamiento. Con este RDBMS, tanto los desarrolladores como los usuarios finales podrán tratar una base de datos distribuida física como una sola base de datos lógica.

Mientras los diferentes productos competidores ofrecen bastantes capacidades distribuidas avanzadas, ORACLE7 ofrece poco más que accesos remotos de lectura y escritura. Otros productos proporcionan funcionalidad de bases de datos en forma de triggers, procedimientos almacenados, reglas de integridad, etc., pero ORACLE7 ofrece un poco de cada uno de estos aspectos.

ORACLE7 provee su propio lenguaje SQL y los procedimientos de acceso a los datos desde al menos un fuente con SQL. Una sola sentencia SQL puede consultar datos desde múltiples bases de datos y realizar complejas uniones de tablas (Joins). Como se mencionó anteriormente, corre en la mayoría de las plataformas de hardware y sobre más de 40 sistemas operativos tales como: UNIX, VMS, MVS, VM, HP MPE/XL, Siemens, ICL, OS/2, Macintosh y Novell Netware.

Debido a lo extenso de su portabilidad, la base de datos ORACLE7 se encuentra por si misma en competencia con una amplia variedad de productos que van desde DB2 de IBM en las mainframes hasta Gupta's SQL Base en una PC LAN.(Personal Computer Local Area Network). Existe una variedad de plataformas tal como Prime, Data General o la Apple Mac donde ORACLE7 es la única opción real disponible de bases de datos portable, y en varias de estas plataformas ha significado un porcentaje importante del mercado.

En las tablas I.5.1 y I.5.2 se pueden observar algunos datos técnicos de interés proporcionados por el fabricante de ORACLE7.

Ventajas

La principal ventaja que ofrece ORACLE7 sobre sus principales competidores, es que es un sistema escalable y completamente portable a más de 80 plataformas de hardware, que van desde sistemas de escritorio hasta mainframes y supercomputadoras, y obviamente corriendo bajo una amplia variedad de sistemas operativos.

Detalles del Producto	Descripción	Plataformas Hardware	Soportado	Sistemas Operativos	Soportado
Vendedor	Oracle	Hardware basado en UNIX	SI	UNIX	SI
Fecha de la primera versión del producto	1979	Mainframes IBM	SI	MVS	NO
Última versión y fecha de liberación	7.0 (Julio 1992)	Digital VAX	SI	VM	NO
Fecha estimada para la próxima versión	Por Anunciar	IBM PC	SI	VMS	SI
		Otros	ICL,VME	OS/2	SI
				MS-DOS	Planeado
				Otros Sistema Operativos	DG/AOS, BTOS, Primos, GCOS, VOS, HP MPE/XC
				Software de comunicación	TCP/IP, SPX/IPX, LU 6.2,etc.

Tabla I.5.1 Detalles, Hardware y Sistema Operativos ORACLE

Parámetros Básicos del Producto	Soportado	Tipos de Datos	Soportado
Número máximo de registros por tabla	Depende del Hardware	Booleano	NO
Número máximo de columnas por registro	Depende del Hardware	Punto flotante	NO
Número máximo de caracteres por campo alfanumérico	Depende del Hardware	Punto Fijo ó Money	NO
Número máximo de tablas por base de datos	Depende del Hardware	Caracter de longitud variable	SI
Memoria Total mínima recomendada (en el servidor)	Depende del Hardware	Fecha (Date)	SI
Memoria recomendada por usuario (en el servidor)	Depende del Hardware	Time	SI
Espacio en disco mínimo requerido	Depende del Hardware	Texto Largo	VARChar - 52K Largo - 52GB
Número máximo de usuarios por base de datos	Depende del Hardware	Objetos grandes (IMAGENES)	Long RAW - 52GB
		Tipos de datos definidos por el usuario	NO
		Tipos de datos compuestos definidos por el usuario	NO
		Especificación de reglas de conversión	NO
		Distinción entre el valor NULL y los valores del dominio	SI

Tabla 1.5.2 Parámetros Básicos y Tipos de Datos ORACLE

En cuanto aspectos de funcionalidad se refiere, ORACLE7 ofrece triggers, procedimientos almacenados, alertas de las bases de datos entre otras. También incluye capacidades distribuidas para el manejo de bases de datos distribuidas, con la cual los usuarios finales podrán tratar varias bases de datos físicas distribuida como una sola base de datos lógica.

ORACLE7 también proporciona una compuerta transparente de herramientas que permite a los usuarios construir compuertas SQL para otras bases de datos y sistemas de archivos. Esta compuerta programable soporta llamadas de procedimientos remotos para todo tipo de sistemas externos.

Finalmente, ORACLE7 proporciona un conjunto de herramientas de desarrollo de aplicaciones muy importantes, entre las cuales se encuentran el SQL*Forms y SQL*Plus, que son las principales, pero otros componentes tratan con la generación de reportes, manejo de redes y facilidades de usuario final, tales como las hojas de cálculo y los gráficos, todo esto gracias a la utilización de lenguajes de cuarta generación (4GL).

Desventajas

Se podría decir que la principal desventaja que presenta ORACLE7, es su gran demanda de memoria para poder almacenar datos en ella, ya que su técnica de almacenamiento hace que esta sea una memoria hambrienta en los sistemas con cargas de trabajo mezcladas, y si la memoria es insuficiente, es probable que el disco sea castigado.

Otra desventaja que presenta ORACLE7 con respecto a sus competidores, es la de que todavía no ofrece un ambiente significativo de Programación Orientada a Objetos, además de que no explota completamente las plataformas paralelas.

1.5.2 INGRES

La base de datos INGRES lleva la batuta y aventaja en algunos aspectos a la mayoría de sus competidores al proporcionar funcionalidad distribuida, optimización de consultas sofisticadas, optimización distribuida, manejo de objetos y alertas de la base de datos.

Actualmente, el énfasis de los productos INGRES está orientado fuertemente hacia los objetos. Para esto, INGRES provee uno de los pocos lenguajes de cuarta generación (4GL) orientado a objetos disponibles en el mercado y posee capacidades para el manejo de objetos dentro de la base de datos lo que lo hace estar por encima de sus competidores. La popularidad de Microsoft Windows ha promovido un fuerte interés en Ingres'Windows 4GL, pero el interés en las capacidades de objetos de la base de datos INGRES ha sido menor, tal vez por que los usuarios no han logrado comprender el potencial de sus usos y beneficios.

Para ofrecer soporte a sistemas de producción y ambientes OLTP, el manejador de bases de datos relacional, INGRES, permite un modelo de organización de los negocios a través del manejo de datos, el conocimiento, y los objetos. Está diseñado para proveer manejo de los datos y capacidades para desarrollar aplicaciones a través de diferentes plataformas de hardware y de software.

INGRES proporciona la habilidad para definir múltiples Servidores Multi-Hebra para acceder y actualizar datos compartidos. El manejo del conocimiento facilita esfuerzos de políticas en las empresas y asegura la integridad referencial. INGRES Event Alert puede usarse para crear aplicaciones que respondan dinámicamente a los

requerimientos de la empresa. Los objetos pueden ser definidos y manejados vía los comandos SQL.

Los módulos de INGRES como son: INGRES/Windows4GL, un lenguaje de cuarta generación (4GL) orientado a objetos y manejador de aplicaciones, y el INGRES/Visión, un desarrollador de aplicaciones están disponibles por separado. Las aplicaciones corren idénticamente en cualquier sistema y los datos pueden procesarse desde cualquier sitio. INGRES es portable en IBM, DEC, Data General, HP, ICL, Pyramid, Sequent, UNISYS y plataformas DOS, y en sistemas corriendo bajo los sistemas operativos: VMS, ULTRIX, MVS, VM/CMS, UNIX, y DOS. La arquitectura del servidor INGRES provee procesamiento tanto para el servidor como para los CPUs anfitriones.

En las tablas 1.5.3 y 1.5.4 se pueden observar algunos datos técnicos de interés característicos de INGRES proporcionados por su fabricante.

Ventajas

Al igual que la mayoría de RDBMS, INGRES está diseñado para proporcionar un manejo eficiente de los datos, además de ofrecer capacidades para que los usuarios puedan desarrollar aplicaciones bajo diferentes plataformas de hardware y software. Así, para los usuarios resulta transparente el hecho de que los datos puedan ser procesados desde cualquier sitio y las aplicaciones se ejecuten idénticamente en los diferentes sistemas.

Otros aspectos en los que INGRES aventaja a la competencia, es en cuanto a su excelente funcionalidad y optimización distribuidas, por supuesto el manejo de objetos

y las alertas de bases de datos.

El énfasis de los productos INGRES está orientado fuertemente hacia los objetos, por lo que provee uno de los pocos lenguajes de cuarta generación orientado a objetos disponibles en el mercado.

Provee soporte a redes através de sus módulos Ingres Net e Ingres Star, los cuales ofrecen capacidades de Cliente/Servidor por medio de diferentes ambientes de red y el manejo de bases de datos distribuidas permitiendo la existencia de un diccionario de datos global.

Finalmente, INGRES ofrece un conjunto de herramientas que conforman un ambiente integrado de desarrollo bastante completo.

Detalles del Producto	Descripción	Plataformas Hardware	Soporta do	Sistemas Operativos	Soportad o
Vendedor	Ingres	Hardware basado en UNIX	SI	UNIX	SI
Fecha de la primera versión del producto	1980	Mainframes IBM	SI	MVS	SI
Última versión y fecha de liberación	6.4 (Nov. 1991)	Digital VAX	SI	VM	NO
Fecha estimada para la próxima versión	6.5 (1993)	IBM PC	SI	VMS	SI
		Otros	Bull & ICL	OS/2	SI
				MS-DOS	SI
				Otros Sistema Operativos	VME,Bull, VMS de seguridad
				Software de comunicación	TCP/IP, SPX/IPX, AT&T

Tabla I.5.3 Detalles, Hardware y Sistema Operativos INGRES

Parámetros Básicos del Producto	Soportado	Tipos de Datos	Soportado
Número máximo de registros por tabla	No limitado	Booleano	SI
Número máximo de columnas por registro	300	Punto flotante	SI
Número máximo de caracteres por campo alfanumérico	2008	Punto Fijo ó Money	SI
Número máximo de tablas por base de datos	No limitado	Caracter de longitud variable	SI
Memoria Total mínima recomendada (en el servidor)	8MB	Fecha (Date)	SI
Memoria recomendada por usuario (en el servidor)	50K-150K	Time	SI
Espacio en disco mínimo requerido	80MB Servidor	Texto Largo	SI
Número máximo de usuarios por base de datos	No limitado	Objetos grandes (IMAGENES)	SI
		Tipos de datos definidos por el usuario	SI
		Tipos de datos compuestos definidos por el usuario	NO
		Especificación de reglas de conversión	SI
		Distinción entre el valor NULL y los valores del dominio	SI

Tabla I.5.4 Parámetros Básicos y Tipos de Datos INGRES

Desventajas

Uno de los aspectos en los que hay que tener mucho cuidado, es en la implementación del diseño de la base de datos INGRES, ya que de lo contrario estará propensa a los cuellos de botella durante la consulta y actualización en sistemas intensivos.

INGRES adolece en la contención de datos causandole problemas con la actualización de aplicaciones intensivas debido a que sólo puede realizar bloqueos a un nivel de página de la base de datos, las cuales son páginas fijas de 2K.

Otra desventaja es el soporte de la lectura sucia (Dirty Read), la cual permite a un usuario leer datos sin considerar a los demás usuarios que están realizando operaciones de inserción, actualización y borrado, lo que no garantiza que las vistas de la base de datos sean consistentes.

I.5.3 SYBASE

Es difícil establecer la filosofía para un producto como SYBASE. Pero ciertamente su facilidad de uso es relevante para el trabajo de los administradores de bases de datos. El énfasis está en dejar decidir que hacer sin tener la preocupación de como hacerlo. En varios aspectos esto es una excelente aproximación, pero su desempeño es bajo, ya que las funciones automatizadas probablemente nunca igualarán a los ajustes en línea específicos de un ambiente en particular. Un mayor énfasis está en la conectividad con otros ambientes, un área donde SYBASE ha tenido un liderazgo sobre muchos productos de la competencia.

En la próxima versión de SYBASE las tres características más importantes serán: las herramientas orientadas a objetos, alta capacidad de redes distribuidas y soporte para sistemas grandes. La versión más reciente incluye un desarrollo conjunto con NCR.

SYBASE es un sistema manejador de bases de datos relacional (RDBMS) con la capacidad de ejecutar aplicaciones en línea. Este RDBMS está basado en la arquitectura Cliente/Servidor y el Servidor SQL, en el cual las funciones de manipulación de los datos pueden ser manejadas separadamente desde las funciones de interfase de usuario.

El Servidor SQL de Sybase maneja los datos y la memoria, teniendo la capacidad de administrar múltiples bases de datos y múltiples usuarios. Registra la localización actual de los datos en el disco, mapeando la descripción lógica de los datos con sus datos físicos almacenados. Mantiene datos y procedimientos caches siempre en la memoria. El Servidor SQL compila y ejecuta sentencias T-SQL (Transact SQL),

siempre y cuando las peticiones sean hechas en un formato estándar de SQL, retornando siempre los resultados de los programas cliente.

El Transact-SQL que maneja el servidor SQL de Sybase es una versión mejorada del ANSI Standard. Este incluye construcciones para: 1) Definición de Datos; 2) Manipulación de Datos, y 3) Control de Datos. Dentro de las mejoras de Sybase se incluyen las siguientes:

- Control de Flujo de sentencias.
- Tipos de datos definidos por el usuario y otros tipos de datos adicionales.
- Procedimientos Almacenados y Triggers.

Al igual que ORACLE7 e INGRES, el Servidor SQL SYBASE está diseñado bajo una arquitectura Multi-Hebra, en donde el servidor por sí mismo puede administrar a múltiples usuarios. No se apoya en el sistema operativo de la computadora host para llevar a cabo la multi-tárea, reduciéndole de esta forma la carga de trabajo.

El servidor SQL de Sybase está estructurado para que pueda trabajar en ambientes de RED, no siendo un producto de máquina-única modificado para red. El diseño del servidor reduce significativamente el tráfico de la RED. Los Clientes y Servidores pueden estar físicamente separados en máquinas diferentes. El acceso se hace especificando un Nombre Lógico el cual representa al Servidor; este nombre es trasladado a una dirección de la red vía un archivo ASCII pequeño el cual reside en el ambiente de cada máquina (Archivo de Interface). El mismo mecanismo también es usado para habilitar que múltiples servidores residan en una misma máquina y así permitir que los servidores se comuniquen con otros servidores.

Sybase proporciona un conjunto de herramientas SQL (SQL Toolset) que tienen una variedad de aplicaciones. Dentro de estas herramientas se encuentran las siguientes:

- ISQL (SQL Interactivo) para consultas al momento.
- Data WorkBench para trabajar con SQL y Report WorkBench.
- Bulk Copy para transferencias dentro y fuera de la base de datos.
- APT para el desarrollo de aplicaciones con ventanas.
- Etc.

El Servidor SQL está disponible para un amplio rango de sistemas de computación que van desde PC's hasta Minicomputadoras.

En las tablas I.5.5 y I.5.6 se pueden observar algunos datos técnicos de interés proporcionados por el fabricante de SYBASE.

Ventajas

Dentro de las ventajas que ofrece SYBASE está su arquitectura de Cliente/Servidor y el Servidor SQL que permiten la manipulación de datos separadamente desde las interfaces de usuario.

La arquitectura Cliente/Servidor ofrece una excelente conectividad con otras plataformas de hardware y software.

El Servidor SQL está estructurado para soportar ambientes de redes y su diseño reduce significativamente el tráfico en estas.

Detalles del Producto	Descripción	Plataformas Hardware	Soporta do	Sistemas Operativos	Soporta do
Vendedor	Sybase Softwae	Hardware basado en UNIX	SI	UNIX	SI
Fecha de la primera versión del producto	1987	Mainframes IBM	SI	MVS	SI
Última versión y fecha de liberación	4.8 (1992)	Digital VAX	SI	VM	NO
Fecha estimada para la próxima versión	5.0 (Mediados 1993)	IBM PC	SI	VMS	SI
		Otros	Stratus	OS/2	SI
				MS-DOS	SI
				Otros SO	
				Software de comunicación	TCP/IP, Novell, DECNET, SNA, LU 6.2 & LAN Manager

Tabla I.5.5 Detalles, Hardware y Sistema Operativos SYBASE

Parámetros Básicos del Producto	Soportado	Tipos de Datos	Soportado
Número máximo de registros por tabla	No limitado	Booleano	SI
Número máximo de columnas por registro	255	Punto flotante	SI
Número máximo de caracteres por campo alfanumérico	2GB	Punto Fijo ó Money	SI
Número máximo de tablas por base de datos	32,000	Caracter de longitud variable	SI
Memoria Total mínima recomendada (en el servidor)	8MB	Fecha (Date)	SI
Memoria recomendada por usuario (en el servidor)	40K	Time	SI
Espacio en disco mínimo requerido	50MB Servidor	Texto Largo	SI
Número máximo de usuarios por base de datos	Sobre 1024	Objetos grandes (IMAGENES)	SI
		Tipos de datos definidos por el usuario	SI
		Tipos de datos compuestos definidos por el usuario	NO
		Especificación de reglas de conversión	SI
		Distinción entre el valor NULL y los valores del dominio	SI

Tabla I.5.6 Parámetros Básicos y Tipos de Datos SYBASE

Los clientes y los servidores pueden estar separados físicamente en máquinas diferentes.

SYBASE trabaja con Transact-SQL que es una versión mejorada del SQL ANSI estándar. Este incluye construcciones para la Definición, Manipulación y Control de Datos.

Finalmente, SYBASE ofrece un ambiente de desarrollo que consiste de herramientas de ciclo de vida SQL, las cuales están divididas en tres clases: herramientas CASE, herramientas SQL y herramientas de prueba Sybase

Desventajas

Existen dudas con respecto al performace de SYBASE en ambientes grandes OLTP, ya que no ofrece el soporte requerido por los programadores en estos ambientes.

Una de las desventajas más significativas es la falta de portabilidad total, ya que las aplicaciones desarrolladas en una plataforma no pueden ser fácilmente ejecutadas bajo otra diferente.

CAPITULO II

ANALISIS DE LA PROBLEMÁTICA DE LA SUBDIRECCION DE ABASTECIMIENTOS DEL GRUPO PEÑOLES

II.1 ANTECEDENTES

Servicios Industriales Peñoles, es una empresa prestadora de servicios administrativos de planeación y desarrollo, de asesoría técnica, de comercialización, de capacitación, y de todo aquello que contribuya al mejoramiento de las actividades productivas de cada una de las diferentes empresas de Peñoles, que a su vez se conjuntan en Grupos y Divisiones.

Estas empresas, controlan a más de 35 unidades productivas en por lo menos 30 localidades diferentes del país, además de las oficinas comercializadoras en Brasil y Estados Unidos; participación en sociedades con empresas en Argentina y España, un Centro de Investigaciones Técnicas, en Monterrey, Nuevo León, así como uno de Investigaciones Geológicas en Chihuahua, Chihuahua.

Las Divisiones y Grupos que conjuntan a las diferentes empresas de Peñoles son las siguientes:

- a) GRUPO MINAS
- b) GRUPO METALES
- c) DIVISION QUIMICOS INDUSTRIALES
- d) DIVISION REFRACTARIOS
- e) DIVISION DE LOGISTICA Y ABASTECIMIENTOS

f) GRUPO FINANZAS

g) GRUPO JURIDICO Y DE RELACIONES

h) GRUPO DE PLANEACION Y DESARROLLO

A manera de ubicar a la Subdirección de Abastecimientos dentro de la empresa, podemos decir que ésta forma parte de la Dirección de Logística y Abastecimientos, la cual representa a la División con el mismo nombre, esto se puede observar en la figura II.1.1.

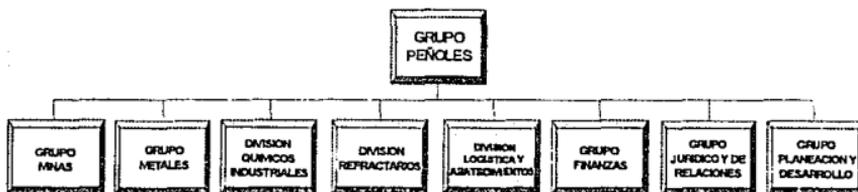


Figura II.1.1 Organigrama general del Grupo Peñoles

División de Logística y Abastecimientos

Esta división es responsable de la adquisición de bienes y servicios, resguardo y custodia de los materiales en cuestión, para asegurar la continuidad de las operaciones del Grupo Peñoles.

Subdirección de Abastecimientos

La misión de esta Subdirección es la de establecer las políticas, sistemas, procedimientos y estrategias específicas para las negociaciones y adjudicaciones de pedidos y contratos para abastecer con calidad y al mejor costo a las compañías del grupo Peñoles de los bienes y servicios que se requieran de proveedores externos.

Las principales funciones que se llevan a cabo dentro de la Subdirección, son las siguientes:

Identificación de Productos

- Convenios de Suministro.
- Contratos de Servicios Técnicos.
- Contratos para manejo de Materiales.

Identificación de Servicios

- Asesorías sobre Bienes y Servicios.
- Investigación y Evaluación de Proveedores.
- Atención a Proveedores y Contratistas.
- Asesorías sobre Administración de Inventarios.
- Cotizaciones de Proveedores y Contratistas.
- Normatividad para Compras Locales.

Identificación de Clientes

	UNIDADES OPERATIVAS
	<ul style="list-style-type: none"> - ALMACENES - DEPTOS. MANTENIMIENTO - DEPTOS. PRODUCCION
ABASTECIMIENTO	INGENIERIA Y CONSTRUCCIONES
	DEPTOS. ADMINISTRATIVOS.

Unidad Organizacional de la Subdirección de Abastecimientos

Se entiende por Unidad Organizacional aquél conjunto de personas que tienen por meta la cristalización de los objetivos de la Subdirección, y por ende de la Dirección de Logística y Abastecimientos.

En la figura II.1.2. se muestra el organigrama representativo de la Dirección de Logística y Abastecimientos.

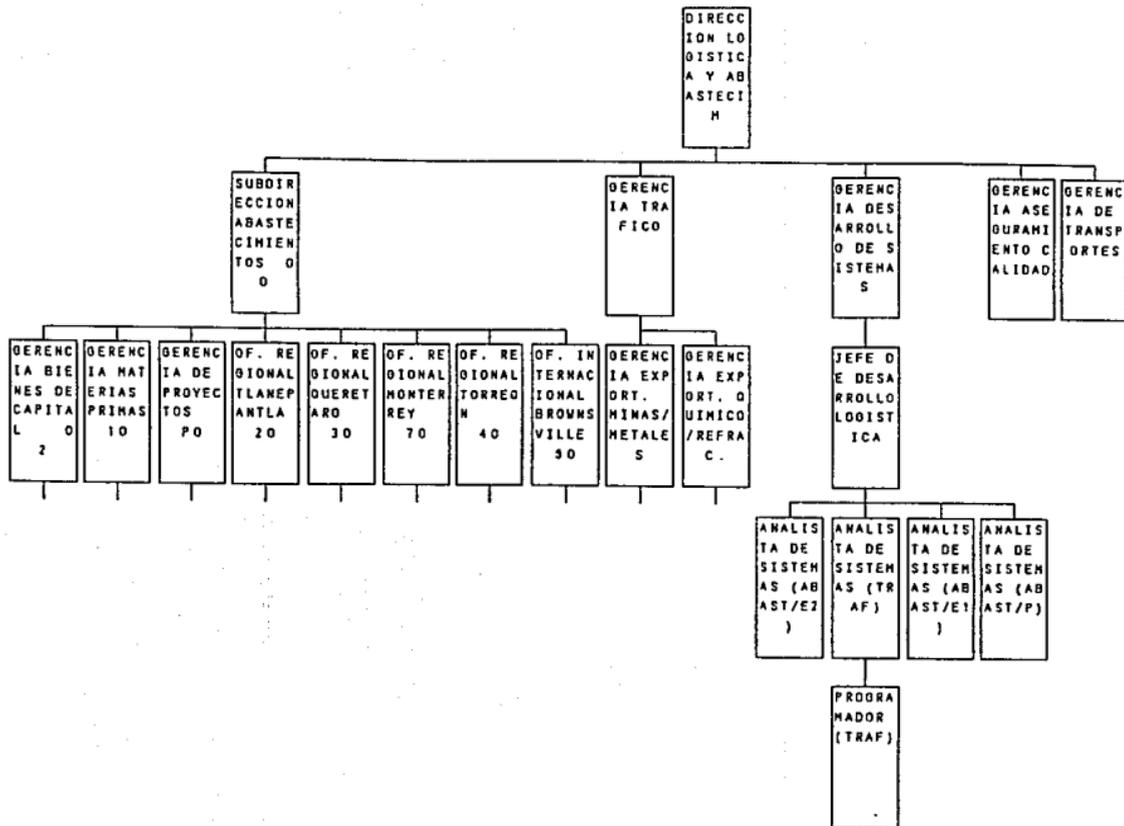


Figura II.1.2. Organigrama de la Dirección Logística y Abastecimientos

Gerencia de Bienes de Capital

MISION:

Abastecer de maquinaria, equipo, refacciones y servicios a las diferentes empresas del grupo Peñoles, en el tiempo y con la calidad requerida.

Los principales giros en los cuales esta área puede abastecer son:

- Requisiciones que necesiten autorización de desembolso.
- Vehículos.
- Material para Hospitales.
- Equipo Minero.
- Equipo Eléctrico.
- Equipos de Proceso.
- Máquinas Herramienta.
- Equipo y Refacciones para mantenimiento preventivo y correctivo

Gerencia de Materias Primas

MISION:

Lograr y efectuar el abastecimiento de materias primas dentro de los mejores análisis de mercado tanto a nivel nacional como internacional, para cubrir de la

forma más eficiente, económica y oportuna, los requerimientos de plantas y unidades del grupo.

Clásicamente las líneas que maneja ésta Gerencia están divididas en dos:

MATERIAS PRIMAS.

- Metales y minerales no ferrosos.
- Material y herramientas para barrenación en minas.
- Cables para minas.

CONSUMIBLES.

- Empaques y envases.
- Aceites y lubricantes.
- Combustibles.
- Flejes.
- Tuberías y conexiones.
- Válvulas.
- Explosivos.
- Materiales estructurales y de refuerzo para construcción.

Gerencia de Proyectos

MISION:

Administrar y supervisar el abastecimiento de materiales y equipo para el Servicio Corporativo y nuevos proyectos del grupo Peñoles (Unidades, Plantas y Oficinas).

El abastecimiento de lo que es el Servicio Corporativo, puede ser del siguiente orden:

- Mobiliario.
- Equipo de cómputo.
 - Hardware y Software (sólo de importación).
- Consumibles (según políticas).
- Telecomunicaciones.
- Vehículos.
- Activo Fijo.

Para proyectos es:

- Equipo.
- Materiales para construcción.

El proyecto está basado en lo que se llama Programa de Proyectos el cual es suministrado por Ingeniería, y se compone de:

- Equipo de proceso.
- + Equipo eléctrico.
- + Materiales estructurales y de refuerzo.
- Tuberías, válvulas y conexiones.
- Material eléctrico.
- Instrumentación.
- * Misceláneos para construcción.

+ Material y equipo que en ocasiones se evalúa con Ingeniería de Proyecto, si no, sólo se compra.

* No siempre se aplica este rubro en un plan de proyectos.

Oficinas Regionales

Con lo que respecta a las Oficinas Regionales, podemos decir que se encuentran ubicadas en diferentes puntos del País, siendo las siguientes: Tlanepantla, Querétaro, Monterrey, Torreón, Chihuahua y la Oficina Internacional de Brownsville.

MISION:

Compra de materiales y refacciones para su uso y reventa en las diferentes compañías del grupo, cubriendo únicamente los costos de operación.

Antecedentes de la automatización de la Información de la Subdirección de Abastecimientos

Una vez identificadas las funciones y la estructura organizacional de la Subdirección de Abastecimientos, cabe señalar que gran parte de éstas funciones y las relaciones existentes entre las diferentes Gerencias y Oficinas Regionales, se encontraban ya automatizadas por un sistema de compras previamente desarrollado.

Los aspectos que caracterizaron al sistema en su concepción original, fue en un principio su mala planeación dado que su desarrollo se fue dando conforme a las necesidades de los usuarios, creándose de esta forma pequeños sistemas independientes, que posteriormente pasaron a formar parte de este primer sistema.

La herramienta en la que se desarrolló el sistema en sus primeras etapas, fue SISINF, que en su tiempo fue poderosa y se adaptaba a las necesidades de la empresa. De esta forma, el sistema comenzó a funcionar a partir de 1985 hasta la liberación del actual Sistema de Abastecimientos.

Con lo que respecta al nuevo sistema es preciso mencionar que este formará parte de un sistema horizontal, en donde estará integrado con otros sistemas complementarios.

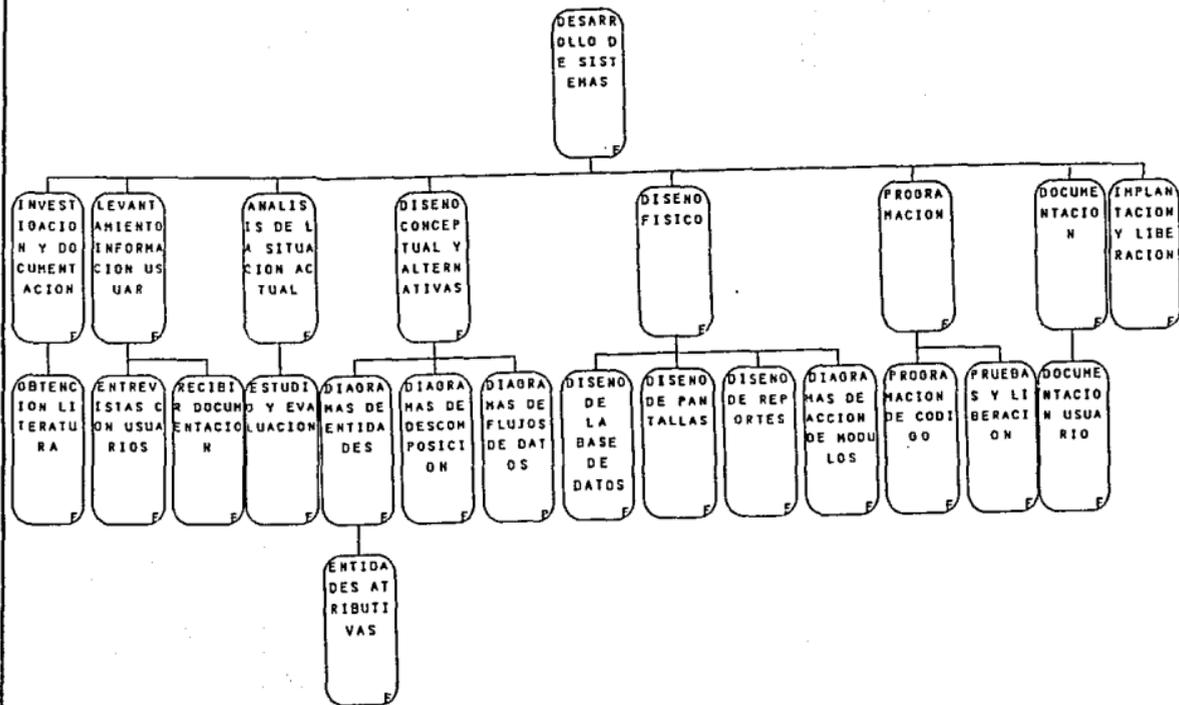


Figura II.1.3 Etapas en el desarrollo de un Sistema de Información.

Una vez presentado el panorama de los antecedentes que atañen a nuestro sistema y antes de comenzar con la descripción del proceso de análisis, se presentan a grandes rasgos las etapas del desarrollo de un sistema (figura II.1.3) y los diagramas de flujo que describen el proceso que se sigue en el departamento de Sistemas de la Dirección de Logística y Abastecimientos para el desarrollo de un nuevo sistema, figuras II.1.4 y II.1.5.

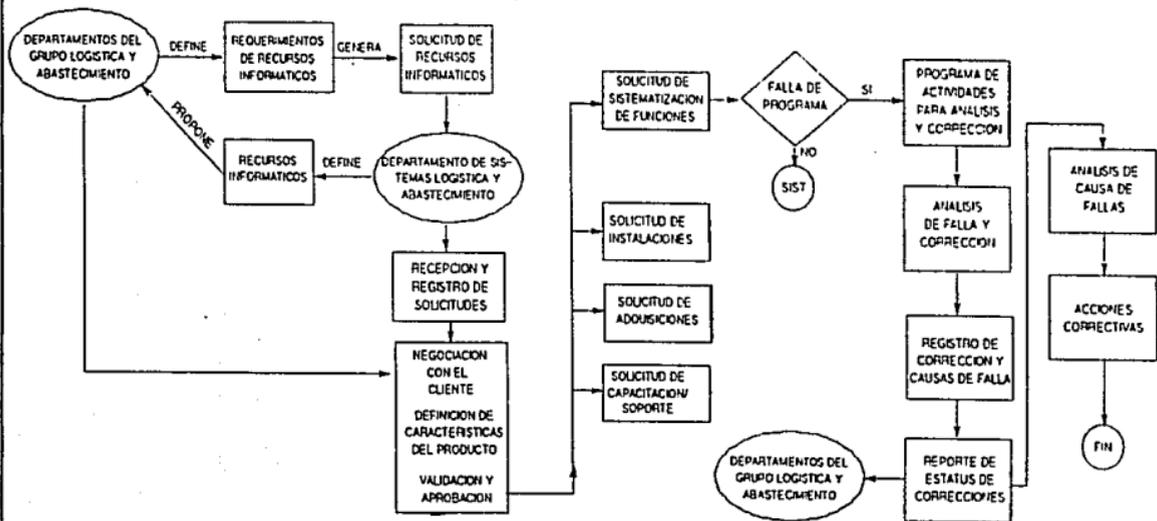


Figura II.1.4 Proceso que sigue el departamento de Sistemas para el desarrollo de un nuevo sistema.

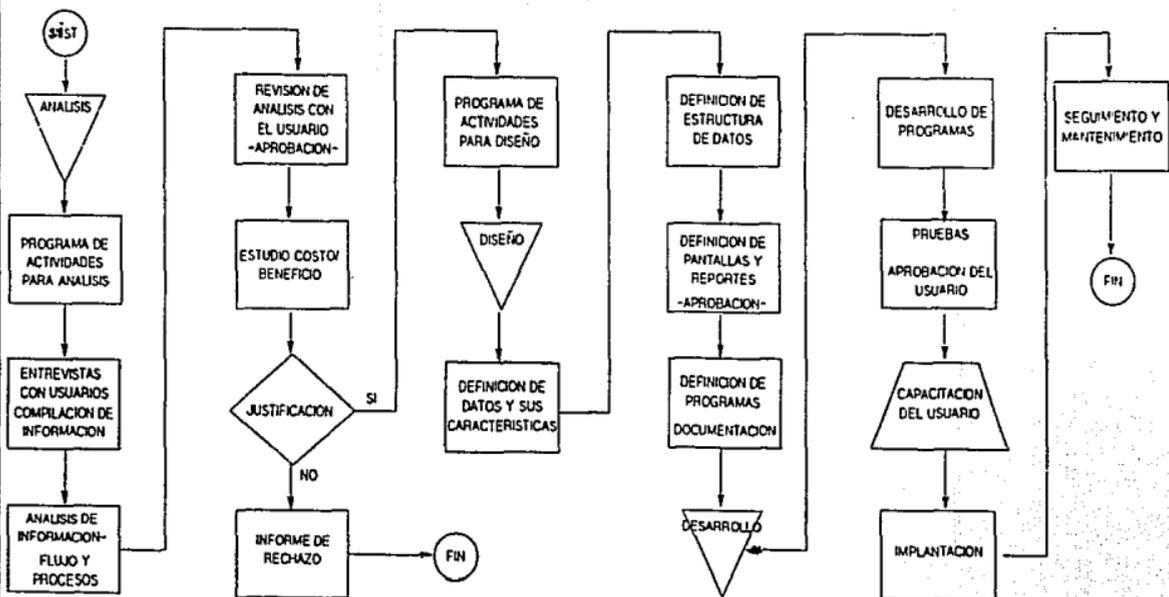


Figura II.1.5 Flujo entre las Etapas del desarrollo de un Sistema de Información.

II. 2 DETECCION DE LA PROBLEMÁTICA

II.2.1 Puntos Críticos

Dentro de la empresa surge la necesidad de producir un cambio en cuanto al manejo de los sistemas de información, debido a que se detectaron los siguientes problemas:

- Las limitantes que empezó a presentar la herramienta original (SISINF) en que fue desarrollado, ya que esta no evolucionó a la par con las necesidades de la empresa, tales como procesamiento y transferencia de información, comunicaciones, interfaces con otros sistemas, etc. Esto provocó la necesidad de migrar a una nueva plataforma que cubriera dichos requerimientos.
- Los problemas ocasionados por el sistema de compras anterior, mostrados en la figura II.2.1.

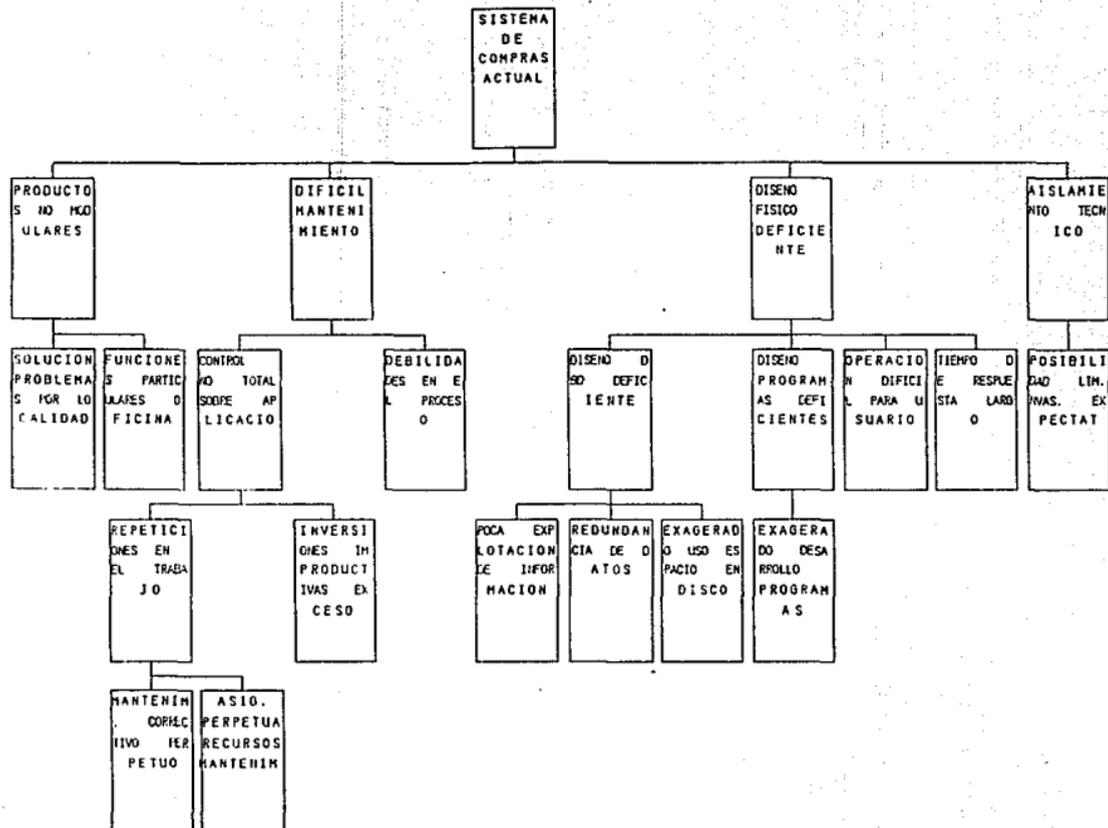


Figura II.2.1. Problemas detectados en el Sistema de Compras anterior

II.2.2 Consecuencias

- **Productos no modulares.**

Repetición de datos, recaptura y funciones que no interesan dependiendo de la localidad.

Catálogos siempre desactualizados y no coherentes dependiendo de la localidad.

Solución problemas por localidad.

Al dar soluciones por localidad, el sistema se convierte en uno por cada una de ellas, arrojando como resultado un difícil mantenimiento ya que esto demanda un mayor número de recursos humanos para resolverlos; de aquí que el tiempo de respuesta a la solución se lenta.

Funciones Particulares Oficina:

La falta de definición de políticas y procedimientos administrativos en las oficinas regionales, causó que el sistema de compras fuera instalado en su totalidad teniendo como consecuencia duplicidad de funciones y tareas en cada una de ellas.

- **Difícil Mantenimiento.**

Por el mal diseño de archivos y la gran cantidad de programas que a los cuales SISINF obligó a hacer, el mantenimiento y control era demasiado costoso.

Control No Total Sobre Aplicaciones.

El particularizar módulos o sistemas para cada oficina regional trajo como consecuencia el no control sobre la aplicación, además de no estar necesariamente actualizados, la diversidad de programas sean comunes pero adaptados a la localidad, ocasionó también la pérdida de control sobre estos.

Repeticiones en el Trabajo.

Nuevamente la no planeación de metas, procedimientos, objetivos y el desconocimiento del trabajo mismo por parte de la relación usuario-sistemas, ocasionó en varios eventos la repetición de tareas.

Mantenimiento Correctivo Perpetuo.

Fue una consecuencia más de la falta de planeación y fallas en el diseño físico del sistema. Esto ocasionó que prácticamente no hubiera desarrollo y la gente estuviera dedicada al mantenimiento correctivo.

Asignación Perpetua de Recursos de Mantenimiento.

El área de sistemas para la Dirección de Logística y Abastecimientos debe cubrir todas las áreas de dicha dirección y no asignar más del 50% del tiempo del área de sistemas a únicamente abastecimientos.

Inversiones Improductivas en Exceso.

Consecuencia de objetivos indefinidos y trabajos no planeados.

Debilidades en el Proceso.

Problema ocasionado por el desarrollo de sistemas sin una metodología y por el desconocimiento de la herramienta de trabajo.

- **Diseño de Base de Datos Deficiente.**

Desconocimiento en la metodología del desarrollo de sistemas y de la herramienta de trabajo.

Poca Explotación de Información:

Esto debido a que este sistema en un principio no fue planeado como tal, si no sólo como programas aislados que cumplían una función, al ir creciendo, jamás se previó una estructura adecuada para la explotación de información.

Redundancia de Datos.

Desconocimiento total de la metodología de desarrollo de sistemas y de conceptos elementales como la normalización de archivos.

Datos repetidos en diferentes archivos, como mismos datos con diferentes nombres, datos diferentes en diferentes localidades, particularidades por localidad.

Exagerado uso de Espacio en Disco.

Básicamente los puntos que hicieron el uso indiscriminado de espacio en disco son los siguientes:

- Sistemas no modulares.
- Sistemas no parametrizados.
- Redundancia de datos.
- Objetivos y planeación deficiente.
- Diseños físicos deficientes.
- Metodología de desarrollo inexistente o no apropiada.

- **Diseño de Programas Deficientes.**

Está ocasionado por los siguientes factores:

- Objetivos y funciones inciertas.
- Diseño de algoritmos ineficientes.
- Descripción de programas inexistentes.
- Desconocimiento de la herramienta de trabajo.
- Trabajos no planeados y de rápida solución.

Exagerado Desarrollo de Programas.

Proliferación de programas por puntos citados en el padre de este problema, y por falta de documentación de los mismos.

- **Operación Difícil para Usuarios.**

El hecho de tener una operación difícil del sistema, dió como resultado un mayor tiempo en el registro de transacciones disminuyendo la posibilidad de análisis y seguimiento en compradores.

- **Tiempo de Respuesta Largo.**

Consecuencia de :

- Diseño físico deficiente.
- Desconocimiento de la herramienta de trabajo.

- **Aislamiento Técnico.**

Al desarrollarse este sistema en una herramienta poco conocida en el mercado, todas las posibles conexiones entre este y demás productos de alcance internacional están virtualmente nulificadas.

Posibilidades Limitadas de Nuevas Expectativas.

El desarrollar sistemas a través de una herramienta de trabajo poco conocida en el mercado, orilló a no poder tener interfaces hacia otros productos.

II.3 IDENTIFICACION Y ANALISIS DE LOS REQUERIMIENTOS DEL AREA

La determinación de requerimientos es el estudio de un sistema para conocer como trabaja y dónde es necesario efectuar mejoras. Los estudios de sistemas dan como resultado una evaluación de la forma como trabajan los métodos empleados y si es necesario o posible realizar ajustes. Estos estudios consideran métodos tanto basados en computadoras como manuales; es decir, no se circunscriben exclusivamente a estudios de cómputo.

Un requerimiento es una característica que debe incluirse en un nuevo sistema. Esto puede ser la inclusión de determinada forma para capturar o procesar datos, producir información, controlar un actividad de la empresa o brindar soporte a la Subdirección de Abastecimientos. Es así como la determinación de requerimientos vincula el estudio de un sistema existente con la recopilación de detalles relacionados con él.

El primer paso en la determinación de requerimientos es comprender la situación. Ciertos tipos de requerimientos son tan fundamentales que son comunes en casi todas las situaciones. Dar respuesta a un grupo específico de preguntas será de gran ayuda para comprender los requerimientos básicos del área.

Actividades de la determinación de requerimientos

Es útil ver la determinación de requerimientos a través de tres grandes actividades: anticipación, investigación y especificación de requerimientos descritos en la siguiente tabla.

ACTIVIDAD	DESCRIPCION
Anticipación de Requerimientos	Prever las características del sistema con base en la experiencia previa. Esto puede llevarnos a investigar áreas y aspectos que de otra forma no serían tomados en cuenta.
Investigación de Requerimientos	Estudio y documentación del sistema actual utilizando para ello técnicas para hallar hechos, análisis de flujo de datos y análisis de decisión.
Especificación de Requerimientos	Análisis de los datos que describen el sistema para determinar que tan bueno es su desempeño, que requerimientos se deben satisfacer y las estrategias para alcanzarlos.

Anticipación de Requerimientos

La experiencia de los analistas en un área en particular y el contacto con sistemas en un ambiente similar al que se encuentra bajo investigación tiene influencia sobre el estudio que éstos realizan. Su experiencia les permite anticipar ciertos problemas o características y requerimientos para un nuevo sistema. Por lo tanto, es probable que las características que investigan del sistema actual, las preguntas que formulan o los métodos que utilizan estén basados sobre esta familiaridad.

Tener las bases necesarias para saber que preguntar y que aspectos investigar puede ser de beneficio sustancial para la organización .

Por otra parte, si se introducen sesgos o atajos al conducir la investigación, entonces es muy probable que la anticipación de requerimientos se convierta en un problema por tanto, siempre deben darse lineamientos para estructurar una investigación alrededor de cuestiones básicas con la finalidad de evitar consecuencias indeseables de la anticipación de requerimientos.

Investigación de Requerimientos

Esta actividad es la más importante para el análisis de los requerimientos. Se debe estudiar el sistema actual con la ayuda de varias herramientas y documentar sus características para más adelante, emprender el análisis.

La investigación de requerimientos depende de las técnicas para encontrar datos y métodos para documentar y describir las características del sistema.

Especificaciones de Requerimientos

Los datos obtenidos durante la recopilación de hechos se analizan para determinar las especificaciones de los requerimientos, es decir, la descripción de las características del nuevo sistema. Esta actividad tiene tres partes relacionadas entre sí:

- **Análisis de datos basados en hechos reales**

Se examinan los datos recopilados durante el estudio, incluidos en la documentación de flujo de datos y análisis de decisiones, para examinar el grado de desempeño del sistema y si cumple con las demandas de la organización.

- **Identificación de requerimientos esenciales**

Características que deben incluirse en el nuevo sistema y que van desde detalles de operación hasta criterios de desempeño.

- **Selección de estrategias para satisfacer los requerimientos**

Métodos que serán utilizados para alcanzar los requerimientos establecidos y seleccionados. Estos forman la base para el diseño de sistemas, los cuales deben cumplir con la especificación de requerimientos.

La especificación de requerimientos implica una gran responsabilidad para los analistas de sistemas ya que la calidad del trabajo realizado en esta etapa se verá reflejada más adelante en las características del nuevo sistema.

Para la obtención de la información de los requerimientos fue necesario en un principio comprender el funcionamiento del sistema de compras anterior, una vez entendido el funcionamiento de los procesos que este llevaba a cabo, se procedió a la detección de los nuevos requerimientos a través de la combinación de entrevistas con los usuarios, observación, recopilación de documentos y revisión de manuales escritos sobre políticas, regulaciones y procedimientos de operación, identificándose así los procesos ya existentes que se adaptaban a las nuevas necesidades de información y eliminando aquellos que no serían utilizados.

De esta forma se encontró que los requerimientos de la Subdirección de Abastecimientos estaban basados en una serie de factores críticos de éxito, entendiéndose por factor crítico de éxito (FCE) como la aspiración fundamental de la empresa o grupo de trabajo a la cual se quiere alcanzar y que es necesario cumplir para encontrar los objetivos o metas de la empresa.

Estos factores críticos se pueden observar en la figura II.3.1., y se describen a continuación:

				TRANSPORTACION RAPIDA Y EFICIENTE	
				RECEPCION RAPIDA DE DOCUMENTOS	
			MIN. TIEMPO EN CICLO DE COMPRA		
			INFORMACION OPORTUNA		
		CONFIABILIDAD DE LA COTIZACION			
COMUNICACION EFICIENTE Y RAPIDA	I	D	D	D	D
CONFIABILIDAD SIST. AUTOMATIZADO	I	D	D	D	I
CONOCIMIENTO DE PROVEEDORES	D	D	I	I	
CONOCIMIENTO DE TRANSPORTES	D	D	I	I	D

Figura II.3.1. Factores Críticos de Exito (FCE)

Comunicación Eficiente y Rápida:

La comunicación entre los cuatro elementos principales en el proceso de compra (Requisitor, Comprador, Expedidor y Proveedor) debe ser rápida para agilizar la compra, confiable en sus datos pues de eso depende que se compre exactamente lo que se necesita, se transporte de manera oportuna y segura, y que se entregue cuando se requiere.

Confiabilidad del Sistema Automatizado:

La Subdirección de Abastecimiento requiere de un sistema de cómputo que le proporcione información confiable en sus datos, rápida y de la manera más clara posible.

Conocimiento de Proveedores:

El conocimiento de proveedores implica que se tenga información sobre las características de los productos que ofrece y que son útiles a la empresa, así como también de los tiempos de entrega, formas de pago, origen, confiabilidad, etc.

Confiabilidad de la Cotización:

Implica que una cotización se válida en el tiempo, conseguida de la manera más rápida posible y que sea una opción totalmente viable para la colocación de un pedido. Al decir viable para la colocación del pedido nos referimos a precios, lugares de entrega, confianza en el proveedor, etc.

Información Oportuna:

Proporcionar la información del estado de los tráficos, requisiciones y pedidos.

Recepción rápida de documentos:

La recepción y captación rápida de documentos de trabajo tales como las requisiciones hechas por unidades, así como las cotizaciones hechas por los proveedores impactarán en la reducción del ciclo de una orden de compra, y así su abastecimiento a unidades.

Esta recepción de documentos puede ser de forma manual o en forma automática de acuerdo a las facilidades que se tengan en cada unidad.

Los FCE antes mencionados son afectados directamente (D) ó indirectamente (I) por los problemas detectados en el sistema anterior de acuerdo a la figura II.3.2.

	REQUERIMIENTOS DEL CLIENTE							
	PLANEACION PROCESOS				PORTABLE PRODUCCION			
	OBTENCION DE LA CALIDAD				OPERATIVIDAD			
	FACILIDAD DE INTERFASES				EFICIENCIA EN EL PROCESO			
	ATENCIÓN RECLAMACIONES CLIENTES							
AISLAMIENTO TECNICO		I	D		D			
ASIGNACION PERPETUA RECURSOS	D				I			I
MANTENIMIENTO								
CONTROL NO TOTAL SOBRE APLICACION		D			I			
DEBILIDADES EN EL PROCESO	I	D	I	I	D			I
DIFICIL MANTENIMIENTO	D							
DISEÑO DE BD DEFICIENTE		D	I		I		D	I
DISEÑO FISICO DEFICIENTE		D	I		D			D
DISEÑO PROGRAMAS DEFICIENTES		D			D			I
EXAGERADO DESARROLLO PROGRAMAS		D			D			
EXAGERADO USO ESPACIO EN DISCO		I			I			
FUNCIONES PARTICULARES OFICINA	D	D	D		D		D	D
INVERSIONES IMPRODUCTIVAS EN EXCESO	D				I			
MANTEN. CORRECTIVO PERPETUO	I							I
OPERACION DIFICIL PARA USUARIO					I	D	I	
POCA EXPLOTACION DE INFORMACION					D		D	D
POSIBILIDAD LIM. NVAS. EXPECTATIVAS					I		I	D
PRODUCTOS NO MODULARES	I		D	D		D		
REDUNDANCIA DE DATOS		D	I		I		I	
REPETICIONES EN EL TRABAJO	I	D			D	I		
SISTEMA DE COMPRAS ACTUAL	D	D	D	D	D	D	D	D
SOLUCION PROBLEMAS POR LOCALIDAD	D	D	D	D	I	I	D	
TIEMPO DE RESPUESTA LARGO	D	D			D			

Figura II.3.2 Como afectan los problemas a los Factores Críticos de Exito.

También, en base a los factores críticos, se encontró que los requerimientos de información de la Subdirección de Abastecimientos se clasifican de la siguiente manera (Figura II.3.3):

Infraestructura

- Cotización de Monedas
- Catálogo de Proveedores
- Catálogo de Materiales
- Catálogo de Compañías
- Catálogo de Bancos
- Catálogo Corporativo

Estadísticas

- Desempeño de compra
- Compras por producto
- Desempeño de proveedor
- Demanda esperada
- Plan de compra de productos

Pagos

- Ordenes de Pago Elaboradas
- Ordenes de Pago Pendientes
- Ordenes de Pago por Proveedor

Movimientos de Materiales

- Situación del embarque
- Notificación de envío
- Remisiones de envío

Pedidos

Pedidos por estatus

Resúmenes de Pedidos

Pedidos pendientes

Modificaciones

Pedidos por compañías

Pedidos por comprador

Pedidos por proveedor

Control de pagos

Cotizaciones

Solicitud de cotizaciones

Requisiciones

Resúmenes de requisiciones

Requisiciones sin colocar

Requisiciones canceladas

Requisiciones colocadas

Requisiciones recibidas

Requisiciones por comprador

Requisiciones por compañía

Requisiciones por estatus

Requisiciones pendientes

Analizando dichos requerimientos y de acuerdo a la problemática se encontró que los principales puntos a atacar son los que se listan a continuación:

- Los problemas se deben atacar desde un punto de vista organizacional y de planeación, jamás desde un punto de vista operativo.
- Al descentralizar el sistema, tendremos una mejor funcionalidad y control sobre la información que se genere en cada localidad, menor uso de espacio en disco y funciones específicas por y para cada integrante de la localidad.
- Cualquier sistema debe ser parametrizado hasta donde sea posible, dando lugar a funcionalidad en él, así como información exclusiva de la localidad.
- Al estar parametrizado el sistema se tendrá un control total sobre las aplicaciones. Otro punto será establecer normas más estrictas sobre modificación y actualización en las diferentes localidades.
- Independencia entre oficinas, sistemas parametrizados dependientes de un sistema centralizado.
- Un diseño eficiente de archivos significará un ahorro considerable en el mantenimiento del sistema.

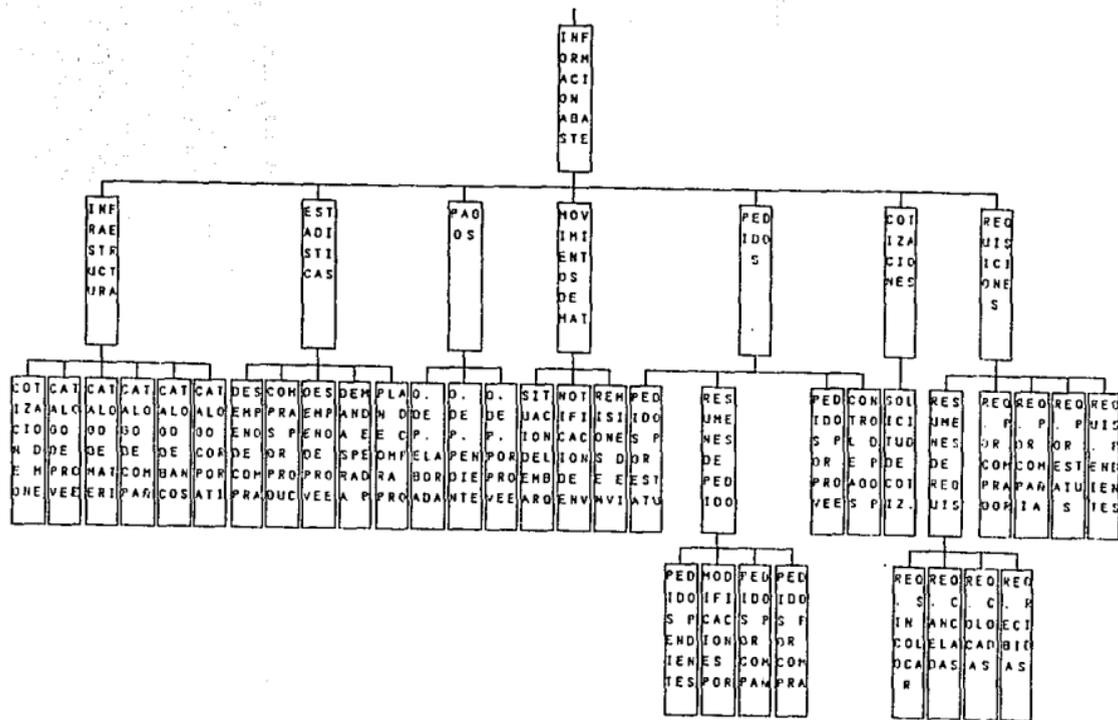


Figura II.3.3 Requerimientos de Información de la Subdirección de Abastecimientos

- Una correcta planeación y una detección oportuna de fallas en el diseño físico del sistema, provocará que el mantenimiento correctivo sea mínimo.

- Al no tener una asignación constante para el mantenimiento correctivo del sistema de compras, se tendrá más tiempo para hacer análisis y diseños de nuevas aplicaciones que beneficiarán a todos.

- Dominar las herramientas de trabajo, comprensión total de los conceptos que se van a manejar en el proceso y su relación con otros.

- Conocimiento de la metodología de trabajo

- Prever una estructura adecuada para la explotación de información.

- Conocimiento total de la metodología de desarrollo de sistemas y de conceptos de normalización de archivos.

- Al evitar el uso exagerado de espacio en disco, se obtendrán los siguientes beneficios:
 1. Mayor eficiencia en procesos automatizados.
 2. Espacio en disco suficiente para otras aplicaciones.

- Al diseñar eficientemente los programas los beneficios serán:

1. Procesos más eficientes.
2. Mantenimiento correctivo minimizado.
3. Objetivos y funciones satisfechas.

- Al evitar un desarrollo exagerado de programas los beneficios resultantes serán:

1. Programas específicos para funciones específicas.
2. Documentación.
3. Control sobre el desarrollo.

- Al disminuir los tiempos de respuesta implícitamente se arrojarán los resultados siguientes:

1. Tiempo de respuesta adecuado al tipo de transacción realizada.
2. Mejor distribución en el tiempo para la realización de las funciones y responsabilidades de la unidades organizacionales.

- Menor tiempo en el registro de transacciones, disminuyendo la posibilidad de error al tener un sistema amigable y de fácil operación.

- Evitar el aislamiento técnico al desarrollar el sistema con una herramienta que proporcione una interfase para conectares con otros productos.

- Al aumentar la posibilidad de nuevas expectativas los beneficios generados son los que a continuación se mencionan:

1. Herramientas de trabajo desarrolladas por firmas mundialmente reconocidas, que tienen las posibilidades económicas para investigación y desarrollo, y así mantenernos siempre actualizados de acuerdo a la tendencia mundial en el desarrollo de software.
2. Crecimiento ilimitado de acuerdo a las misiones que marque la Dirección de Logística y el Grupo Peñoles.

Analizando la información anterior, se tiene que los productos, entendiéndose por estos, entes que tienen un beneficio para algo o para alguien, estos equivalen a procesos, los cuales son actividades que se ejecutan repetidamente dentro de una empresa, cada ejecución produce un efecto específico sobre entidades o información acerca de ellas. Los productos que ofrece la Gerencia de Sistemas son los que se muestran en la figura II.3.4.

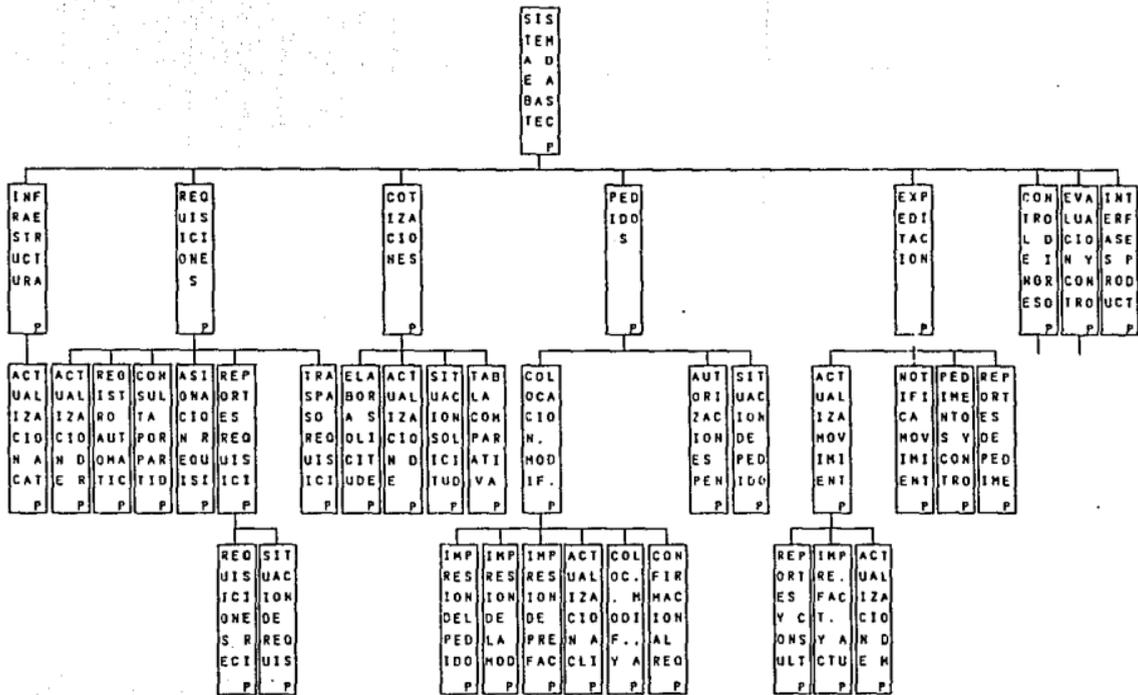


Figura II.3.4 Productos que ofrece de la Gerencia de Sistemas

II.4 ANALISIS DE LA INFORMACION

Este análisis sirvió para comprender mejor los requerimientos de información, el uso que se le da, como y para que se utiliza. Para esto fue necesario entender el proceso administrativo (Figura II.4.1) que sigue la Subdirección de Abastecimientos y cuyos pasos se describen a continuación en el orden en que se efectúan:

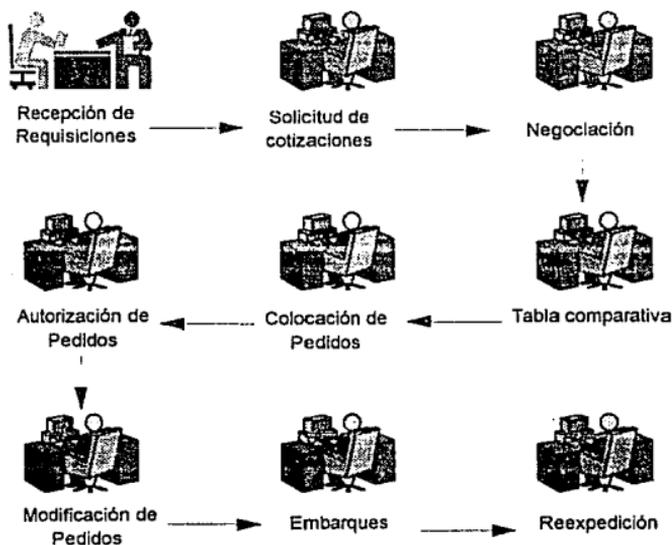


Figura II.4.1 Proceso Administrativo de la Subdirección de Abastecimientos

Proceso Administrativo

1. Recepción de Requisiciones.

Es la primera etapa del proceso en la cual se llevan a cabo la recepción de las requisiciones que llegan de las siete oficinas regionales de compras: Chihuahua, Querétaro, Monterrey, Torreón, Tlalnepantla ,México y Wideco (Brownsville). En este momento a las requisiciones se les asigna un estatus "SC" de Solicitando Cotización.

Las requisiciones de acuerdo a la línea de material solicitada , son clasificadas en tres categorías:

- Bienes de Capital.

- Vehículos.

- Equipo minero.

- Equipo eléctrico.

- Equipo de proceso.

- Equipo para hospitales.

- Máquinas herramienta.

- Equipo y refacciones para equipo correctivo y preventivo.

- Etc.

- Materias Primas.

- Metales y minerales no ferrosos.

- Material y herramientas para barrenación en minas.

- Cables para minas.

Empaques y envases.

Aceites y lubricantes.

Combustibles.

Flejes.

Explosivos.

Etc.

- **Administrativos.**

Papelería.

Equipo de cómputo.

Equipo para oficina.

Suministros.

Etc.

Una vez hecha esta clasificación, se tiene que determinar la procedencia del material (importación o nacional) y el monto de la requisición, ya que esto determinará a que oficina regional será enviada esta, así por ejemplo, las requisiciones de bienes de capital, las de materiales especializados de importación que representen un costo muy alto se mandan directamente a la oficina regional de México, y aquellas requisiciones de importación que no necesiten un trato especializado y que su costo esté dentro del presupuesto de compra se mandan a la oficina regional de Brownsville. Por otra parte, la requisiciones de compra nacionales que no representen un costo muy alto se mandan a las demás oficinas regionales.

2. Solicitud de Cotización.

En esta etapa se elaboran varias solicitudes de cotización para proporcionárselas a los posibles proveedores que pueden cotizar las requisiciones. Al momento de emitir la solicitud de cotización el estatus de la requisición cambia a "EC" de Esperando Cotización.

3. Negociación.

Es la etapa donde el comprador negocia con los diferentes proveedores las condiciones de compra, tales como precios por partida, descuentos, tiempos de entrega, etc., llevándose un registro de todas estas negociaciones. El estatus de la requisición sigue en "EC".

4. Tabla comparativa.

Es donde el comprador tomando los datos de las negociaciones finales hace una comparación de las condiciones de compra ofrecidas por los diferentes proveedores. Es aquí donde el comprador toma la decisión sobre a cual o a cuales proveedores se les asignará el pedido que ampare dicha requisición.

Es importante hacer notar que estos tres últimos puntos son opcionales y al criterio del comprador, lo que se debió considerar en el momento de hacer el diseño del sistema.

5. Colocación de pedidos.

Es cuando se elabora el pedido que ampara a una o varias requisiciones con todas las especificaciones que el proveedor necesita. En este momento al pedido se le asigna un estatus Por Autorizar "PA" y la requisición para a un estatus de En Tramite Final "TF". Si todas las partidas de una requisición han sido amparadas se le asigna el estatus de Colocado Totalmente "CT".

6. Autorización de pedidos.

Es cuando las personas responsables de autorizar, como son los Directores, Subdirectores, Gerentes y Compradores, dan el visto bueno al pedido, plasmando su firma. Las personas tienen la capacidad de autorizar un pedido dependiendo de su monto límite asignado.

Al momento de autorizar un pedido su estatus pasa a Por Entregar "XE".

7. Modificación de pedidos.

En esta etapa se realizan cambios a los datos del pedido como pueden ser precios, cantidades, fechas, etc. Cuando se hace una modificación el pedido pasa de nuevo al estatus "PA" y después de que se autoriza el pedido pasa al estatus que tenía anteriormente.

8. Embarques.

Se refiere a cuando el proveedor envía el material al almacén especificado por algún medio de transporte, como puede ser Barco, Tren, Avión, Camión u otros.

Para el caso de pedidos nacionales el proveedor se encarga del embarque y notifica de esto a la oficina regional correspondiente, en este momento al pedido se le asigna el estatus de Embarcado "EM". Por otro lado, cuando se trata de pedidos de importación los cuales se colocan con proveedores extranjeros el procedimiento varia, ya que el proveedor manda el material a la frontera con el agente aduanal indicado o especificado, lo cual tiene que notificar a la oficina de compras correspondiente. Cuando llega el embarque a la frontera el agente aduanal lo notifica, y de aquí en adelante la oficina de compras se desliga de la responsabilidad, en este momento el estatus del pedido pasa a Recibo en Frontera "RF". El departamento de Tráfico se encarga de pagar al agente aduanal los impuestos y los fletes para hacer que el embarque llegue a su destino.

9. Reexpedición.

Es cuando el embarque va en camino a los almacenes y la oficina regional de compras responsable está esperando la confirmación de que el almacén ya recibió el material, en este momento el pedido pasa a un estatus de Esperando "MR". De esta forma cuando el almacén confirma que el material ha sido recibido el pedido pasa a un estatus de Recibido "RB". Una vez que el comprador checó que el material fue recibido

satisfactoriamente y que no hubo ningún problema con el proveedor cierra el pedido poniéndole un estatus de Pedido Cerrado "CE".

El ciclo se cierra cuando el estatus de la requisición es de Colocada Totalmente "CT" y el estatus del pedido es "CE".

De acuerdo al análisis anterior se encontró que los datos requeridos para llevar a cabo cada una de las actividades anteriores son los que se muestran a continuación:

Datos requeridos:

FUNCION	DATOS
<p>Captura de Requisiciones</p>	<p>Cve. comprador Centro (Almacén) Consecutivo de la requisición Cuenta de cargo Cve. autorización desembolso Monto estimado Depto. solicitante Solicitante Equipo Folio Línea Sublínea Descripción Genérica Fecha de recepción Fecha de elaboración Fecha de requisición Días estimados colocación Tipo (R,A,S) Prioridad (A,B,C) Activo Fijo (S/N) Número de partidas Partidas colocadas Partidas programadas Número de notas Estatus requisición Fecha de estatus</p>
<p>Consulta por partida</p>	<p>Clave comprador Centro Consecutivo requisición Número de partidas Partidas colocadas Consecutivo partida Cantidad Código material Unidad de medida Descripción Estatus Línea Sublínea</p>

Solicitud de Cotización	Número de solicitud Fecha de solicitud Centro Requisición Fecha límite cotización Notas Partidas totales requisición Partidas en la solicitud
Cotización	Clave cotización Fecha cotización Razón social proveedor RFC Número de solicitud Centro Consecutivo requisición Fecha vigencia Moneda Condición de pago Condición entrega Lugar entrega Tiempo entrega en días Descuento general Descuento por partida Descuento por pronto pago
Colocación pedidos	Clave comprador Centro Rango reqs. amparadas Clave cotización Descuento Proveedor Número de pedido Consecutivo requisición Estatus Cotización Vigencia
Partida por pedido	Centro Consecutivo requisición Solicitante Clave del pedido Estatus Cotización Consecutivo Partida

	Cantidad/Pedido Precio unitario Descuento Tiempo entrega Programado (S/N) Unidad de medida Descripción
Modificación Pedidos	Clave comprador Centro Clave pedido Consecutivo pedido Estatus Fecha de estatus Tipo pedido Fecha elaboración Clave gerente autoriza Fecha de autorización Número de modificación Número partidas pedido Número de notas Cotización Proveedor Persona que atiende Forma de pago Días forma de pago Descuento por pronto pago Condición de entrega Lugar de entrega Día consignación Domicilio a consignar Domicilio a facturar Convenio de suministro Fecha de embarque Días embarque Moneda Tipo cambio Valor bruto Descuento IVA Valor neto Impuesto

Pedidos Programados	Consecutivo partida Cantidad Fecha inicio programación Número de periodos Tipo periodo (Mes,Sem., Día) Número entregas Entregas programadas Fecha programada Cantidad por periodo Cantidad total
---------------------	--

II. 5 ALTERNATIVAS DE SOLUCION

Tomando en cuenta los requerimientos de información de la Subdirección de Abastecimientos, tanto inmediatos como futuros se presentan dos opciones de solución que deberán ofrecer los recursos necesarios para satisfacer los requerimientos que se mencionan a continuación:

- Sistema Modular.
- Navegación amigable.
- Parametrizado.
- Estructura de datos eficiente.
- Portable.
- Facilidad en interfaces.
- Interface con otros productos.
- Herramientas para el desarrollo de un sistema de calidad.
- Documentación del sistema.

Cada uno de estos se explican más a detalle a continuación.

Sistema Modular.

Un módulo de procesos de información está caracterizado por el hecho de sólo manejar información homogénea dentro de él, haciéndolo así independiente.

Navegación Amigable.

Es la facilidad con que se opera un evento, brindando al usuario final mayor sencillez y comprensión en el uso del sistema, y la posibilidad de conexión de varios procesos entre sí.

Parametrizado.

Es el que todas aquéllas variables y funciones de un proceso o grupo de éstos, sean de dominio externo, es decir, variables y funciones no deben de ser parte del proceso, sino este las debe tomar de un lugar específico en donde se definan éstas.

Estructura de Datos Eficiente.

Estructura de datos totalmente normalizada para así tener un fácil y total explotación de la información.

Portable.

Es el poder instalar un sistema o módulo de este en una localidad específica bajo posiblemente otra plataforma de cómputo sin que exista problema.

Facilidad en Interfaces.

Es la facilidad que se tiene en la comunicación entre un sistema y otro.

Interface con otros Productos.

Es la facilidad y transparencia que se tiene en la conectividad entre dos o más herramientas de trabajo. Su objetivo principal es el poder hacer una gran explotación de información con herramientas especializadas a resolver un tópico en especial.

Herramientas para el Desarrollo de un Sistema de Calidad.

Es el desarrollo de un sistema bajo una herramienta de trabajo de Calidad Internacional Reconocida, con la que se pueda asegurar la permanencia del producto y su conectividad con otras herramientas de la misma clase.

Documentación del sistema.

Es la documentación que debe existir para el usuario y sistemas facilitando así un mejor aprovechamiento del producto.

Las alternativas de solución propuestas para llevar a cabo la satisfacción de los requerimientos antes mencionados se detallan a continuación:

- Actualización del sistema en SISINF
- Implementación en Bases de Datos Relacionales. (RDBMS)

Cada una de las opciones anteriores se describe brevemente y se evalúa de acuerdo a los siguientes factores:

- Tamaño de la empresa
- Necesidades de comunicación
- Tipo de sistemas a desarrollar
- Adaptabilidad al equipo actual
- Recursos económicos

Actualización del sistema en SISINF.

Esta opción fue considerada porque en un principio se pensó que se podría ahorrar tiempo, trabajo y costo al momento de actualizar o rediseñar el sistema, debido a que este ya se encontraba desarrollado en SISINF.

En primer lugar se consideró el hecho de que ya se disponía de dicha herramienta y por consiguiente del equipo necesario, aprovechando de esta forma las ventajas que representaba y al mismo tiempo evitando la adquisición de una nueva herramienta de trabajo. Pero esto representaba enfrentarse a un problema, debido a que SISINF en su época había sido un herramienta poderosa, pero que se fue quedando rezagada en el avance tecnológico, esto se observaba en sus limitantes de manejo de información y comunicaciones.

- **Tamaño de la empresa.**

Este aspecto estaba cubierto desde un principio ya que como se ha mencionado existía un sistema de abastecimientos que de alguna forma u otra cubría las necesidades de los usuarios de la subdirección de abastecimientos. Además de que SISINF ha sido la herramienta estándar de desarrollo utilizada en la mayoría de los departamentos de desarrollo de sistemas.

- **Necesidades de comunicación.**

En este aspecto se comprobó que SISINF no contaba con las herramientas de comunicación en cuanto a interfases con otros sistemas de bases de datos, lo que implicaba un aislamiento de información con respecto a los sistemas instalados en las demás oficinas regionales.

- **Tipo de sistemas a desarrollar**

Este aspecto se refiere básicamente a que si los sistemas implementados en los diferentes departamentos de desarrollo de sistemas de la empresa son de tipo administrativo o técnico. Desde este punto de vista SISINF cumplía con dicho requisito, ya que debido a la naturaleza administrativa del sistema de abastecimiento se adaptaba completamente.

- **Adaptabilidad al equipo actual**

Este aspecto no representó ningún problema para SISINF, ya que originalmente la herramienta ya se encontraba instalada bajo la plataforma de hardware de la que se disponía, que en este caso era una minicomputadora HP3000 serie 800, bajo el sistema operativo MPE XL.

- **Recursos económicos.**

Debido a que ya se disponía de la herramienta y por consiguiente del equipo no implicaba gasto alguno en cuanto a hardware y software, incurriendo solo en costos de recursos humanos, significando esto un ahorro considerable para la empresa.

Implementación en Bases de Datos Relacionales. (RDBMS)

Debido a la necesidad de buscar una herramienta con la cual se cubrieran las deficiencias presentadas por SISINF que como ya se mencionó es la herramienta en la cual se desarrolló el sistema anterior de abastecimientos, se pensó en DBMS que maneja bases de datos relacionales, por las siguientes razones:

1. Aplicaciones portables
 2. Bases de datos distribuidos.
 3. Herramientas Front-End que traducen los diagramas del diseño a código (CASE).
 4. Trabajar bajo el concepto Cliente/Servidor.
 5. Tener una herramienta de buen nivel con el soporte técnico eficiente y oportuno, para la resolución de aquéllos problemas que se pudieran presentar.
 6. Herramienta que evolucionara acorde a la tecnología de vanguardia.
-
- **Tamaño de la empresa.**

Los manejadores de bases de datos actuales ofrecen la posibilidad de correr en diferentes plataformas que van desde redes de computadoras hasta mainframes. Esto proporciona una gran flexibilidad en caso de un crecimiento futuro de los sistemas de información de la empresa, sin que ello representara cambios drásticos en dichos sistema.

Por otro lado, se pensó en una posible migración hacia otras plataformas de hardware, lo que llevó a considerar también el hecho de que la herramienta debería ser portable, evitando de esta forma rediseñar nuevamente el sistema en caso de que se llegará a dar el cambio hacia otro equipo.

- **Necesidades de comunicación**

En este sentido se encontró que los RDBMS actuales están diseñados para cubrir las necesidades de comunicación que se necesiten dentro de cualquier empresa, ya que estos ofrecen las capacidades de manejo de bases de datos distribuidas, compuertas de comunicación que ofrecen interfase con otros RDBMS y estructuras de archivos que no son bases de datos, manejo de comunicación en redes de computadoras entre otras.

De acuerdo a lo anterior, se observa que los RDBMS cubrían las principales necesidades de comunicación del sistema actual de abastecimientos en cuanto a transmisión de información y la interfase con otros sistemas de bases de datos no relacionales como es el caso de SISINF, que es un sistema manejador de bases de datos del tipo Reticular. Esto fue de suma importancia ya que de esta forma se pudo llevar a cabo con éxito el paso de la información del sistema anterior al sistema actual.

- **Tipo de sistemas a desarrollar**

Otro punto a favor de los RDBMS fue el hecho de que son herramientas diseñadas cien por ciento para el desarrollo de aplicaciones de tipo administrativo, lo que influyó notablemente para la elección de dichas herramientas, ya que como se mencionó la mayoría de los sistemas desarrollados en Peñoles son de tipo administrativo.

- **Adaptabilidad al equipo actual**

El aspecto que llevó a considerar desde un principio a los sistemas manejadores de bases de datos relacionales como una alternativa viable, fue precisamente el hecho de que eran herramientas que se adaptaban a las plataformas de hardware y software actuales sin la necesidad de pensar en ese momento en la adquisición de equipo nuevo.

Las plataformas de hardware y software que se tomaron en cuenta para considerar dichas herramientas fueron: una minicomputadora HP9000 series 850 por la parte de hardware y el sistema operativo UNIX por la parte de software.

- **Recursos económicos**

En este punto como es de suponerse los RDBMS representaron un costo considerablemente alto con respecto a la opción anterior, ya que implicaba la

adquisición de la herramienta con sus respectivos módulos que de alguna u otra forma su costo era demasiado elevado debido a que son aplicaciones de vanguardia en cuanto a tecnología se refiere. Además de que fue necesaria la capacitación a todo el personal involucrado en el desarrollo de los sistemas, lo que representó otro costo que añadido al anterior aumento todavía más el costo total.

En general esta fue la única desventaja de los RDBMS, si se le puede llamar desventaja, ya que en general todo cambio representa algún sacrificio y en este caso fue el costo.

CAPITULO III

DISEÑO E IMPLEMENTACION DEL SISTEMA DE ABASTECIMIENTOS

III.1 ELECCION DEL MANEJADOR DE BASES DE DATOS

Antes de llevar a cabo el desarrollo y la implementación del nuevo sistema, se analizaron diferentes Manejadores de Bases de Datos, basándose en las necesidades del sistema para el manejo de información, requerimientos, herramientas para programación y facilidad de uso.

Para la elección del DBMS, se partió de la manera en que debía de interactuar con el usuario, la manera en que a este le es más común pensar respecto a la organización de sus datos, por lo tanto, la consideración por preferencia del usuario es manejar formas tabulares para organizarlos, ya que esta es la manera que le representa mayor sencillez de comprensión y manejo de la información. Además, esto no sólo es significativo para el usuario, sino también para las personas involucradas en el diseño y programación, por la facilidad que implica relacionar distintas tablas.

Por tales motivos, se pensó en DBMS que manejaran la filosofía del modelo relacional de datos y el poder que este le proporciona a las computadoras en cuanto a rapidez, capacidad de almacenamiento, aplicaciones portables, conectividad y facilidad de uso.

III.1.1 Criterios de evaluación

Los productos que se analizaron, cumplían con la filosofía del Modelo Relacional de Datos y con los requerimientos del sistema.

Las herramientas analizadas fueron:

- ORACLE
- INGRES
- SYBASE

Para llevar a cabo la selección, se consideraron los siguientes criterios generales:

- Estructura y Arquitectura
- Funcionalidad
- Funcionalidad distribuida
- Performance
- Manejo de CPU
- Memoria
- Entrada /Salida Disco
- Ambiente de desarrollo
- Utilerías

Para cada uno de los criterios se analizaron los tres DBMS elegidos para poder llegar a la elección de uno.

III.1.1.1 Estructura y Arquitectura

ORACLE

Con la versión 7 (ORACLE7), la base de datos básica de ORACLE incluye aspectos de rendimiento (Performance), tales como el limitador de recursos y las herramientas del Administrador de la Base de Datos (DBA, Data Base Administrator).

Son tres las opciones que caracterizan la estructura de la base de datos ORACLE; una opción *Procedimental* que provee triggers, procedimientos almacenados, alertas de la base de datos y otras funciones, una opción *Distribuida* que incluye todas las capacidades distribuidas valga la redundancia, que han sido añadidas, y por último una opción de *Servidor Paralelo*, la cual está dirigida a la pérdida acoplada y masivamente a las plataformas paralelas. En adición a lo anterior, ORACLE también puede proporcionar una compuerta transparente de herramientas (Transparent Gateway Toolkit) que permite a los usuarios construir compuertas SQL para otras bases de datos y sistemas de archivos. Esta compuerta programable soporta llamadas de procedimientos remotos para todo tipo de sistemas externos.

La arquitectura interna de la versión 7 se muestra en la figura III.1.1.1. En donde los buffers, colas, procedimientos SQL compartidos y los repetidos accesos, están todos contenidos dentro del "Área Global del Sistema" (SGA; System Global Area) . Los usuarios pueden estar configurados para tener un *Servidor Dedicado* o un *Servidor Compartido*.

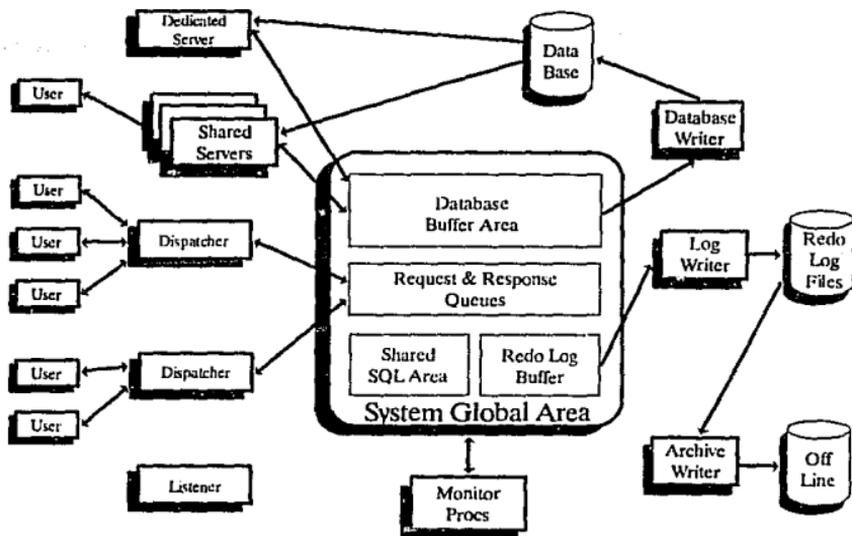


Figura III.1.1.1. Arquitectura Interna de ORACLE 7

INGRES

El conjunto de productos INGRES es extenso, este incluye una selección de lenguajes de cuarta generación (4GLs) y una variedad de otras opciones de desarrollo incluyendo un generador de código. El soporte a redes es proporcionado por los módulos *Ingres Net* e *Ingres Star*. Ingres Net ofrece las capacidades de Cliente/Servidor a través de una variedad de ambientes de red tales como: DESCNet, TCP/IP, SNA, OSI, Novell SPX/IPX y otros. Por otro lado, Ingres Star provee el manejo de bases de datos distribuidas, permitiendo la implementación de un diccionario de datos global. En adición a esto, Ingres Gateways provee acceso a SQL de otras bases de datos y archivos incluyendo DB2, SQL/DS, IMS y Rdb.

Con lo que respecta a las capacidades relacionales de INGRES, este incluye características de seguridad y de puesta a tono, tal como es el *Manejador de Conocimientos*, que permite la definición de reglas a nivel de la base de datos, y el *Manejador de Objetos*, que permite la definición y el manejo de los objetos.

La arquitectura de INGRES es de *Cliente Multiservidor*, la cual permite procesos *front-end* en sistemas cliente para ejecutar aplicaciones, mientras que un servidor o servidores *back-end* proveen el procesamiento de la base de datos. Esta arquitectura consta de tres partes principales: el servidor DBMS, los accesos al sistema (logging) y el bloqueo del sistema y una facilidad general de comunicación. El servidor es de tipo Multi-Hebra, el cual permite el múltiple acceso de usuarios a la base de datos. Los servidores pueden configurarse de acuerdo a las necesidades.

 SYBASE

El conjunto de productos SYBASE esta dividido en tres grandes áreas:

1. Servidores de Software.
2. Herramientas de ciclo de vida SQL.
3. Compuertas para software.

Existen tres versiones del servidor de software: Servidor SQL Microsoft, Servidor SQL de seguridad y el Servidor SQL Sybase. El último de los tres es su principal producto, que corre bajo UNIX, VMS, OS/2 y NLM (Texto basado para Windows); Stratus es también soportado. APT es el lenguaje de cuarta generación (4GL), el cual usa el mismo diccionario de la base de datos. Por otro lado, la herramienta CASE puede ser usada como un auxiliar para el desarrollo de aplicaciones.

La estrategia de la compuerta SYBASE es significativa, diferente a la de la mayoría de los demás vendedores, quienes sólo proporcionan una compuerta con otros productos de base de datos. SYBASE consta de dos componentes principales: el Cliente Abierto (Open Client) y el Servidor Abierto (Open Server), los cuales potencialmente proveen un ruta de migración para una serie de programas existentes y es usada en compuertas de productos para soportar el acceso a Rdb, Oracle, Ingres, Informix y DB2.

El Servidor Abierto provee la capacidad para conectar SYBASE con aplicaciones cliente de sistemas que no operan con SQL por medio de un mecanismo remoto de llamadas a procedimientos, figura III.1.1.2. Un producto especial de Servidor Abierto está disponible para CICS, el cual es más sofisticado que el que puede ofrecer la

mayoría de los competidores, además de que hace a SYBASE más atractivo para los usuario de mainframes IBM.

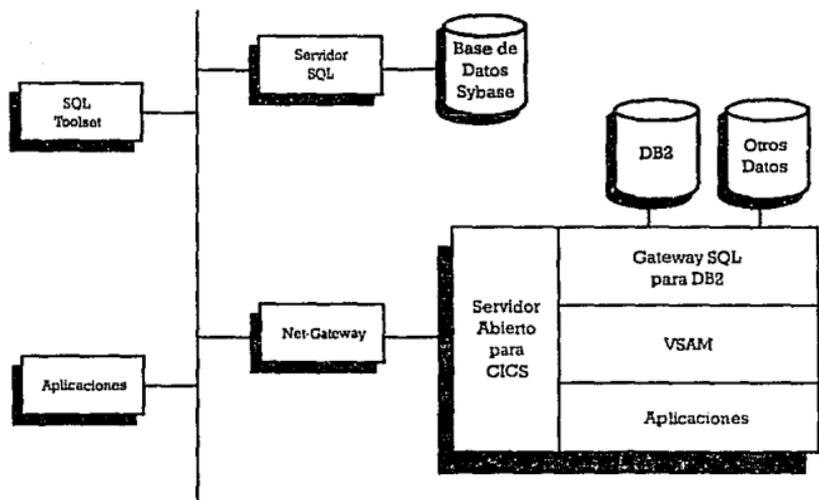


Figura III.1.1.2. Conectividad SYBASE - IBM

La arquitectura de SYBASE es pura Cliente/Servidor. El manejo de datos y de las transacciones son llevados a cabo distintamente por los usuarios y las funciones de aplicación. Esto es manejado por la ruta del "Servidor Programable", usando triggers, procedimientos almacenados, que facilitan el uso óptimo del ambiente de las redes distribuidas.

Las interfaces Cliente/Servidor Abierto (Open Client/Server) permiten a las aplicaciones correr bajo una variedad de software de presentación y provee acceso a otros productos bajo la sombra de SYBASE. Esto constituye una flexible, sensible y evolucionaría estrategia.

III.1.1.2 Funcionalidad

‡ ORACLE.

El grueso de la funcionalidad de ORACLE es provista por la *opción Procedimental*. El lenguaje PL/SQL de ORACLE es usado para escribir procedimientos almacenados; este es un lenguaje de programación basado en Ada, con sentencias SQL fijas que añaden todas las capacidades procedimentales que le hacen falta a SQL. Este es el mismo lenguaje usado en conjunción con Oracle SQL-Forms y con los procedimientos almacenados, constituyendo de esta forma una extensión lógica del conjunto de productos.

Los procedimientos almacenados pueden ser llamados directamente por los usuarios o por los triggers. Los procedimientos anidados, arreglos de procesamiento, y recursión son todos soportados. Para definir de alguna forma los triggers, podemos decir que son procedimientos PL/SQL que ejecutan automáticamente las operaciones de INSERT, UPDATE y DELETE. Estos pueden fijarse como triggers antes o después de cada evento, y pueden ser disparados uno a la vez por una sentencia SQL o por un renglón. La implementación es bastante comprensiva y está basada en el estándar ANSI/ISO SQL3.

Por otra parte, ORACLE también provee alertas para la base de datos, las cuales pueden ser usadas para notificar a los usuarios de cambios críticos detectados en la base de datos. Estas pueden dispararse por un cambio predefinido de valor y pasan un mensaje a cualquier aplicación que está registrada como interesada en el evento.

El soporte para tipos de datos no estándar es limitado, aunque el almacenamiento de texto es soportado como una opción separada con una interfase separada, pero usando la base de datos de ORACLE.

✦ INGRES

En los últimos años la tendencia con las bases de datos relacionales ha sido la de fortalecer más y más su funcionalidad interna. INGRES ha sido uno de los líderes en este aspecto ofreciendo una mejor sofisticación que la mayoría de los competidores, proporcionando el almacenamiento de reglas, la definición de alertas de la base de datos y el manejo de objetos. Además de que la base de datos también puede manejar la seguridad y el control de los recursos. En la figura III.1.2.1 se muestra a grandes rasgos la funcionalidad de INGRES.

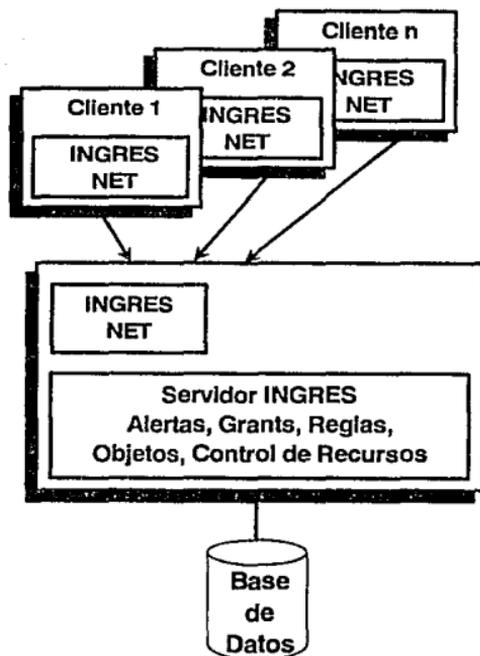


Figura III.1.2.1 Funcionalidad de INGRES

El Manejador de Objetos de Ingres soporta ciertos tipos de objetos orientados hacia la funcionalidad; estos son: la definición de nuevos tipo de datos, la definición de funciones y la definición de nuevos operadores. Probablemente el más importante sea la habilidad para definir nuevos tipo de datos, refiriéndose a los tipos de datos abstractos, aunque los tres están estrechamente relacionados. El manejador de objetos permite al DBA definir nuevos tipo de datos en una sola localización.

El Manejador de Conocimientos permite al DBA definir muchas de las reglas que pueden de cualquier modo estar mantenidas en la base de datos, tal es el caso de las reglas de integridad referencial, pero también permite reglas del negocio que deberán ser definidas e impuestas. Todas las reglas y los procedimientos asociados con estas son almacenados en la base de datos, definiéndose uno a la vez. La base de datos entonces asegura que las reglas y las acciones sean llevadas a donde sean necesarias, y el número de reglas y procedimientos que pueden ser asociados con una tabla no tiene límite.

Las alertas de la base de datos son una extensión de las reglas definidas en dicha base de datos. Cuando una regla es disparada, un proceso externo puede ser activado, esto puede resultar en un mensaje que es enviado a los usuarios interesados.

El uso de los manejadores de Objetos y Conocimientos necesita de consideraciones cuidadosas en el diseño de aplicaciones.

† SYBASE

SYBASE ha estado al frente del mercado en cuanto a la implementación completa de triggers y procedimientos almacenados.

En breve, el Servidor de la base de datos ejecutará la integridad referencial y las reglas de los negocios a través de todos los usuarios. Estas son definidas sólo una vez, almacenadas en la base de datos, y entonces usadas y compartidas por todas las

aplicaciones. Esto hace la vida fácil para los desarrolladores, reduciendo el tráfico en las redes y mejorando la optimización de las consultas.

Dentro de SYBASE, el DBA está limitado a tres triggers por tabla (Insert, Update y Delete), pero los triggers pueden ramificarse en base al if-then-else y las llamadas a procedimientos almacenados. Más aún, SYBASE coloca datos modificados dentro de una tabla y entonces permite a los triggers acceder la tabla.

Los procedimientos almacenados son escritos en Transact-SQL (una extensión de SQL), los cuales proveen parametrización, control lógico de flujo, llamadas anidadas y remotas, variables temporales y mensajería. El dominio de la integridad es forzado al ligar una regla para la definición del dominio usando la cláusula "where" de SQL. La integridad de las tablas o entidades es manejada por la definición de un índice Btree único.

III.1.1.3 Funcionalidad Distribuida

✦ ✦ ORACLE

ORACLE siempre ha provisto una Arquitectura Distribuida peer-to-peer basada en una configuración de "central maestra", en la cual están basadas todas las mejoras realizadas a la versión 7, y las cuales corresponden a la *opción Distribuida*.

El aspecto más importante es la implementación de las transacciones distribuidas. Presenta un comportamiento bifásico (two-phase commit) transparente, el cual aparece fácil de implementar y no requiere de programación especial. El two-phase commit de ORACLE es inusual, en el sentido de que permite declarar uno y sólo un nodo como un nodo no bloqueado. Este nodo necesita no ser una base de datos ORACLE, pero podrá ser cualquier base de datos externa accesada vía una compuerta ORACLE.

ORACLE7 está basado en la especificación XA X/Open para procesamiento de transacciones distribuidas, las que hacen compatible a este con varios tipos de monitores TP, tales como USL Tuxedo y NCR's Top-End. Esto también significa que el two-phase commit puede ser soportado entre ORACLE y otros productos XA acomodables, tal como es el caso de INFORMIX.

Otra parte importante de la opción distribuida es la capacidad para duplicar tablas asíncronamente desde una tabla maestra que se puede actualizar. Esto es, como deberá ser, para crear y actualizar automáticamente copias de sólo lectura (read-only). El intervalo de actualización puede ser definido por el usuario.

En general, un alto grado de transparencia de localización es provisto y las llamadas remotas a procedimientos entre nodos de la red son soportadas. ORACLE también incluye otros aspectos dentro de los cuales se encuentran la administración remota de la base de datos, un nivel de optimización distribuida y soporte para nombres de la base de datos global. La compuerta toolkit está disponible para aquellos usuarios que desean proporcionar acceso SQL transparente a otras bases de datos.

† † INGRES

El uso de las herramientas de INGRES, tal como el servidor Ingres Star, Ingres Gateways e Ingres/Net, permitirán a los usuarios intercambiar datos no sólo entre las bases de datos distribuidas de INGRES, sino también entre bases de datos de otros vendedores en una gran variedad de plataformas de hardware usando las aplicaciones INGRES. Las compuertas (Gateways) ofrecen acceso transparente a otras bases de datos, tales como Rdb, Db2, IMS y archivos RMS. El acceso y la manipulación de datos son manejados por una variedad de SQL que INGRES provee. Al usar OPEN/SQL en aplicaciones, el programador no tiene que preocuparse de las varias modalidades de SQL que son usadas o de los diferentes mensajes de error. Estos aspectos son manejados por la misma compuerta. Así, de esta forma al usuario siempre se le presentará una vista consistente de los datos y si es posible de los errores asociados.

El servidor Ingres Star controla los accesos tanto a las bases de datos distribuidas de INGRES como a las otras bases de datos a través de las compuertas. De esta forma, todas las bases de datos en el sistema parecerán una sola base de datos lógica para el usuario. El acceso a las tablas distribuidas es manejado por un *Diccionario de Datos*

Central que determina dónde pueden ser localizadas las diferentes tablas a través de la red y además compila las aplicaciones. Así, el servidor Ingres Star actúa como un manejador de datos distribuido.

La capacidad proporcionada de optimización distribuida está bastante aventajada y en el ambiente distribuido la capacidad de control de recursos de la base de datos es doblemente útil. Todo esto permite al DBA asignar límites en la utilización de los recursos, en términos de CPU y de Entradas/Salidas a Disco para los usuarios.

La parte débil de la arquitectura distribuida de INGRES es la localización centralizada del diccionario de datos, ya que todos los sistemas participan en el sistema distribuido necesitando acceso al diccionario. Si el sistema responsable de la información falla, entonces la capacidad distribuida del sistema también falla. Entonces, esto se puede mejorar si el diccionario es localizado en una disposición alta de la configuración.

✦ ✦ SYBASE

SYBASE está comprometido con la arquitectura Cliente/Servidor. Este con la comunicación mejorada del two-phase commit Inter-Servidor, indica que SYBASE es uno de los vendedores que puede honestamente ofrecer una facilidad de actualización transparente y distribuida.

Una de las últimas cualidades que han sido añadidas a SYBASE es la excelente conectividad con mainframes IBM, la cual explota LU6.2 y puede proveer acceso a cualquier dato que pueda ser accesado por CICS.

Sin embargo, en el área general de la capacidad distribuida, SYBASE tiene aún que dirigirse a la replicación, fragmentación y la necesidad de un optimizador distribuido, pero se prevé que todo esto ha sido planeado para la siguiente actualización.

La metodología por default para el two-phase commit, es que un servidor será designado para manejar cualquier depósito particular. Después checando que todos los demás servidores estén listos para proceder, la transacción es depositada. Pero, si uno o más de los servidores no están disponibles, entonces la transacción es automáticamente deshechada. Una vez que el servidor maestro ha depositado la transacción, esta permanecerá depositada aún cuando uno de los otros servidores falle. El DBA no tiene elección en este procedimiento si las bases de datos están en el mismo servidor, pero fuera de eso el control programático está disponible.

La integridad de los datos está asegurada por la asociación de un identificador con cada transacción, el cual es generado por el servidor maestro. Este identificador es retenido por todos los servidores secundarios en un archivo de registro (log file). Si alguno de los servidores fallará después de que el maestro ha depositado, entonces la transacción será recuperada en una reinicialización. Similarmente, si tanto el servidor maestro como uno de los servidores secundarios fallan, con una inconsistencia en las bases de datos, entonces la situación será rectificada una vez que el servidor maestro sea reinicializado; el otro servidor entonces checará el estatus de la transacción sobresaliente y la recuperará.

Todas las actualizaciones distribuidas son manejadas por los procedimientos almacenados. El conocimiento de otras bases de datos es mantenido en la base de datos local dentro del procedimiento almacenado. Por otro lado, mientras no sean

elegantes algunas soluciones (Empress' Global Data Dictionary), este nunca proporcionará una localización transparente, en el sentido de que los usuarios no necesitan saber donde es realizada la actualización de los datos.

III.1.1.4 Performance

■ ORACLE

La arquitectura de ORACLE proporciona buena escalabilidad y con la adición de un optimizador bastante mejorado, todos los inconvenientes obvios del desempeño del producto han sido eliminados. Esto no hace que se alcance la velocidad de los productos de base de datos para red más rápidos, tales como el Modelo 204 o Adabas, pero lo hace altamente competitivo con varios de sus primos relacionales. Su escalabilidad e implementación en SMP y plataformas de hardware paralelo ahora hacen a ORACLE situable para muchos sistemas grandes.

■ INGRES

INGRES es rico en estructuras de datos y opciones de puesta a tono que estimulan un buen desempeño (Performance), especialmente para sistemas de consultas intensivas. El optimizador es uno de los más sofisticados y además proporciona una lectura adelantada, escritura diferida, depósitos agrupados y almacenamiento de múltiples archivos, la cual es una forma eficiente de partición horizontal para tablas de gran tamaño. La única debilidad marcada en el área del desempeño, es el bloqueo a nivel de registro, el cual necesitará ajustes de diseño para los sistemas de actualización intensiva.

■ SYBASE

En cuanto al funcionamiento de SYBASE bajo ambientes grandes OLTP se tienen algunas dudas, ya que las mejoras han sido realizadas pero la mayor parte de estas no están disponibles para los programadores o estas son menos refinadas que las de otros productos.

Los servidores pueden ser programados para realizar transacciones repetitivas de una forma muy eficiente y el tiempo de estampado (time-stamping) puede usarse para el bloqueo óptimo debido a la falta de un bloqueo a nivel de registro. El Servidor Abierto accesa a otros sistemas que también proveen una solución elegante y flexible para problemas de procesamiento distribuido a través de las redes con ambientes mixtos.

III.1.1.5 CPU

ORACLE

La arquitectura de ORACLE lo hace particularmente situable para máquinas con múltiples CPU's donde este es capaz de balancear la carga a través de los procesadores, pero las ventajas de esta arquitectura no serán tan notorias en máquinas con un sólo CPU. Sin embargo los servidores son multi-hebra y el SQL compartido es un aspecto de eficiencia para el CPU. En sistemas donde las operaciones aritméticas son comunes y complejas, esto puede significar un uso extra del CPU, pero para la mayoría de las aplicaciones comerciales este efecto no será tan serio.

INGRES

La arquitectura de multiservidores multi-hebra de INGRES permite que este sea apropiado para sistema de múltiples CPU's. Los servidores múltiples pueden ser creados para correr concurrentemente y en paralelo tantos como CPU's estén disponibles. Alternativamente, en sistemas de acoplamiento de pérdida, tal como el VAXcluster, el cual puede tener su propio servidor para acceder a la base de datos o un sistema que está diseñado como un nodo servidor, eliminan el bloqueo del tráfico de la red. En esencia, INGRES proporciona al DBA la flexibilidad para determinar dónde, cómo y en qué prioridad los servidores pueden correr.

INGRES también soporta procedimientos almacenados compilados. Estos son importantes para la actualización de sistemas intensivos donde el overhead de la ejecución interpretativa no es aceptable. El código para una transacción es compilado, almacenado en la base de datos y entonces es cargado a la memoria cuando es requerido. Si es frecuentemente usado, este permanecerá en la memoria. Los procedimientos almacenados también apuntan a un plano de SQL compartido. De esta forma, el uso de la memoria es reducido debido a que el procedimiento y el plano son compartidos por todos los usuarios. El overhead de una red es también reducido debido a que las llamadas son hechas directamente a procedimientos que han tenido bastantes consultas SQL pasando a través de la red.

Por otro lado, la capacidad de INGRES para manejar tipos de datos abstractos permite que este utilice tipos de datos nativos en todas las máquinas de una red (Algunos tipos de datos tal como el "float" varían dependiendo de la plataforma de hardware). El conjunto de productos INGRES usa por sí mismo esta capacidad y contribuye a la obtención de una mayor eficiencia del CPU en una red o hardware mezclado.

SYBASE

SYBASE es muy eficiente en cuanto a recursos se refiere, especialmente con sistemas capaces de atender a 50 o 60 usuarios. Los procedimientos almacenados compilados hacen el uso del CPU y la memoria más rápido y eficiente. En otras palabras, las transacciones frecuentemente usadas serán continuamente interpretadas, optimizadas y compiladas, de esta forma, el procedimiento almacenado puede ser cargado en la

memoria, en donde será optimizado y entonces estará disponible para todos los usuarios. Un procedimiento almacenado sólo necesitará ser recompilado si alguno de los objetos a los cuales hace referencia cambia.

Las capacidades del multiservidor multi-hebra fueron introducidas con las versiones 4.8 y 4.9 de SYBASE, además de que este es capaz de explotar máquinas con múltiples CPUs. SYBASE se refiere a esto como una Arquitectura de Servidor Virtual, el cual utiliza la memoria compartida. Esta arquitectura muy probablemente será extendida en la próxima versión.

La falta de tolerancia (Fault-tolerant) en la operación es soportada al configurar un segundo servidor en un CPU alterno. Este servidor no usa ninguno de los recursos del CPU. Si el procesador principal en el cual está corriendo el servidor de la base de datos falla, el servidor secundario automáticamente conecta a las aplicaciones existentes, recupera la base de datos y continua procesando.

III.1.1.6 Memoria

ORACLE

La clave del desempeño de ORACLE está en el uso efectivo de la memoria, la cual idealmente deberá ser proporcionada con recursos de memoria. La técnica de ORACLE para poseer versiones de datos en la memoria puede hacer que esta se convierta en una memoria hambrienta en los sistemas con cargas de trabajo mezcladas, y si la memoria es insuficiente, es probable que el disco sea castigado.

Otros aspectos cruciales que impactan en el desempeño de ORACLE, son la sintonización de la SGA (Area Global del Sistema) y el funcionamiento eficiente de los *buffers*. De esta forma, dependiendo de la aplicación, esto puede ser necesario para agrupar tablas relacionadas y así alcanzar un desempeño óptimo.

Una característica interesante es la habilidad para compartir SQL compilado entre los usuarios, disminuyendo de esta forma el trabajo de la memoria y el del CPU.

INGRES

Los servidores Multi-Hebra permiten que la memoria sea compartida entre los usuarios, Por lo tanto, la información bloqueada, datos, procedimientos almacenados e índices son todos compartidos por los usuarios de la base de datos en el mismo sistema. El equivalente de memoria usado por el servidor es controlable ajustando el número y el

tamaño de los buffers de la base de datos. Las páginas en la memoria y por supuesto su uso, son controlados por ciertos parámetros del sistema.

SYBASE

Los procedimientos almacenados bloquean información y datos compartidos, mientras que el servidor Multi-Hebra permite que la memoria sea compartida entre todos los usuarios de la base de datos. La cantidad de memoria requerida por usuario es menor que la requerida por los productos competidores. Típicamente el Servidor SYBASE requiere 8MB, más 40K de memoria extra por usuario, lo cual es recomendable para ser altamente eficiente en la mayoría de los eventos.

La única variable para el ajuste de la base de datos disponible para el DBA es el poder variar el número de buffers a un nivel de la base de datos. En realidad es peor que esto, ya que sólo se pueden variar por servidor. Los buffers no pueden ser ajustados a un nivel de tabla y es probable que esto cause problemas, especialmente en sistemas híbridos de consulta OLTP. Las tablas que son frecuentemente pegadas por transacciones OLTP necesitan un poco más de buffers, mientras que las consultas tendrán algunos buffers grandes.

SYBASE usa un algoritmo sofisticado para retener tablas frecuentemente usadas e índices en el cache. Este incrementa la memoria, desplazando substancialmente las desventajas mencionadas anteriormente, pero no en su totalidad.

Poniendo la solución fuera de las manos del DBA al crear un algoritmo reservado, ciertamente hará más fácil la vida de este, siendo una gran estrategia para sistemas low-end. Sin embargo, el DBA no está en la posición de ajustar el performance y por el otro lado el podrá acabar sin obtener el performance deseado.

III.1.1.7 Entradas/Salida Disco.

ORACLE

ORACLE usa procesos de escritura diferida (Asíncrona) para realizar eficientemente E/S a disco (Disk I/O) y manejar los accesos de los usuarios al sistema (login). Esta disponible el particionamiento horizontal para la distribución de tablas activas muy grandes sobre múltiples discos, además de que los índices pueden ser colocados en discos diferentes a los que contienen la mayoría de las bases de datos. Por otro lado, la compresión de índices también está disponible, mientras que las compresión de datos no.

Sin duda la mayor mejora en ORACLE7, es el optimizador estadístico/costo el cual significará una considerable reducción en las operaciones de E/S a disco. La idea detrás del optimizador es que la base de datos pueda conocer más acerca de su propio estado y de las transacciones que un programador siempre puede manejar. De esta forma, podrá encontrar eficientemente la ruta de los datos sin la asistencia de un programador. La versión 7 provee un mejor optimizador y una opción de migración entre dos optimizadores

INGRES

INGRES ofrece soporte para las escrituras diferidas, depósitos en grupo, llaves mezcladas, agrupación de datos y partición horizontal sofisticada para datos e índices. Si la base de datos es diseñada e implementada correctamente, los cuellos de botella son improbables de ser un problema al momento de consultar o actualizar sistemas intensivos.

La facilidad para ajustar tamaños de páginas y el tamaño de los buffers en una tabla base no es vital, pero es importante para los sistemas que contienen el mismo número de actualizaciones y consultas.

SYBASE

Aunque SYBASE soporta escritura diferida y depósitos en grupo, la base de datos por si misma está esparcida para proveer el tipo de estructuras de datos y parámetros que son típicamente requeridos para un buen desempeño en los ambientes OLTP. Algunas mejoras han sido implementadas en el servidor, tales como las llaves mezcladas y las listas invertidas, pero estas no están disponibles para el DBA. Algunos datos agrupados son provistos y el particionamiento horizontal ha sido implementado, aunque esto quedará deshabilitado para los índices. El tamaño de las páginas de la base de datos permanecerá invariante aunque esta sea una página de asignación optimizada para la localización en disco.

El único agrupamiento de datos soportado por SYBASE es la habilidad para almacenar datos de tablas ordenados por índice, lo cual SYBASE asegura que reducirá los puntos calientes. Sin embargo, con sólo esta capacidad no se podrá reducir la Entrada/Salida cuando se requieran hacer uniones de tablas con relaciones "uno a muchos" (1 - n) en el disco, lo cual es el principal beneficio del agrupamiento. Ligeramente más facilidades son habilitadas con los índices, ya que estos pueden ser agrupados en una llave primaria y de esta forma, podrán tener mayor prioridad en el cache.

III.1.1.8 Ambiente de Desarrollo

ORACLE

ORACLE proporciona un ambiente de desarrollo muy extenso, sus componentes más importantes son SQL*Forms y SQL*Plus. Estos comprenden las principales herramientas de desarrollo, pero otros componentes tratan con la generación de reportes, manejo de redes y facilidades de usuario final (end-user), tales como las hojas de cálculo y los gráficos.

Adicionalmente, ORACLE es comúnmente usado en conjunción con 4GLs (Lenguajes de 4a Generación) y es probablemente la base de datos más interoperable bajo revisión junto con DB2.

INGRES

El conjunto de herramientas de INGRES ofrece un ambiente integrado de desarrollo bastante completo, el cual consiste de Ingres/4GL, VIFRED (Visual Forms Editor), Ingres/Vision, Ingres/Forms y otros componentes. INGRES también provee Windows 4GL, que es un ambiente de desarrollo Orientado a Objetos. Todos estos productos están estrechamente asociados con la base de datos INGRES y pueden tomar ventaja del manejador de objetos y conocimiento.

SYBASE

El ambiente de desarrollo de SYBASE consiste de herramientas de ciclo de vida SQL, las cuales están divididas dentro de cuatro categorías, y de las cuales una tratará con los encabezados de las utilidades inferiores.

El conjunto de herramientas CASE (CASE Toolset) es hábil. Entre otras facilidades, tiene la habilidad de comprender y generar esquemas para múltiples DBMS y de esta forma asistir la portabilidad entre SYBASE y otros productos tal como Ingres, Oracle y Rdb.

El conjunto de herramientas SQL (SQL Toolset) consiste de los siguientes módulos: APT Workbench, Data Workbench, Report Workbench, Embedded SQL y SQR. Estos son usados para desarrollar aplicaciones basadas en formas, elaboración y formateo de reportes, insertar sentencias SQL en lenguajes de tercera generación (3GLs) y realizar una escritura avanzada de reportes respectivamente.

Finalmente, el conjunto de herramientas de prueba SYBASE (Sybase Testing Toolset) comprende una ventana en SQL, la cual permite editar y checar código con facilidades de ayuda en el contexto sensitivo. El depurador SQL (SQL Debug) opera interactivamente a nivel fuente. En adición al APT, SYBASE tiene la habilidad para trabajar aproximadamente con 140 lenguajes de cuarta generación (4GLs) y generadores de reportes. Un número de estos pueden ser herramientas para PC con interfase al Servidor SQL Microsoft.

III.1.1.9 Utilerías

ORACLE

ORACLE7 provee un menú para el manejo de las herramientas del DBA, el cual permite un manejo mucho más fácil de las tareas de rutina, constituyendo prácticamente una aplicación front-end para el SQL*DBA. Además, este permite al administrador de la base de datos añadir usuarios y/o archivos, monitoriar el desempeño y el uso y realizar respaldos de la información.

El DBA puede monitoriar quién está usando la base de datos, que tablas se están accedando, cómo los bloqueos han sido generados y el nivel de la actividad de E/S a disco. Otras utilerías soportan el respaldo de la base de datos, recuperación, exportación e importación hacia y desde las estructuras de archivos del sistema operativo.

INGRES

La más importante de las utilerías de INGRES es el Monitor de Desempeño Interactivo (IPM, Interactive Performance Monitor). Este es básicamente un menú para el manejo de utilerías que permite estadísticas interactivas en línea las cuales serán desplegadas en la pantalla para el análisis del desempeño. El acceso es proporcionado para la utilización de los recursos, el archivo log y la información del sistema de reloj.

La utilería Starview es provista con el servidor Star para permitir al DBA observar la base de datos distribuida y conocer de esta forma que tablas están presentes sin tener acceso a la información del diccionario de datos central. Por último, también se proporcionan las utilerías para respaldos y recuperación de información (Backup y Restore), incluyendo soporte de respaldo mientras el sistemas está en línea.

SYBASE

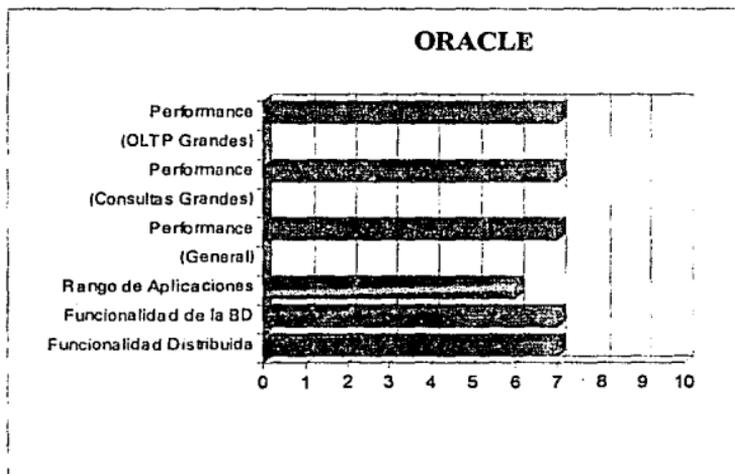
La cuarta parte de las herramientas de ciclo de vida SQL está constituida por el conjunto de Herramientas de Operación, el cual incluye el Sistema de Administración (SA Companion), que es usado por el DBA para instalar los servidores, manejar el espacio en disco y establecer las cuentas de los usuarios de la red. Por otro lado, el Monitor SQL provee información para la puesta a tono de la base de datos, descargar cuentas y tener la capacidad de planeación.

Data Workbench ofrece una transferencia masiva de datos y una utilería de conversión, así como también la habilidad para acceder el catálogo del sistema en donde el

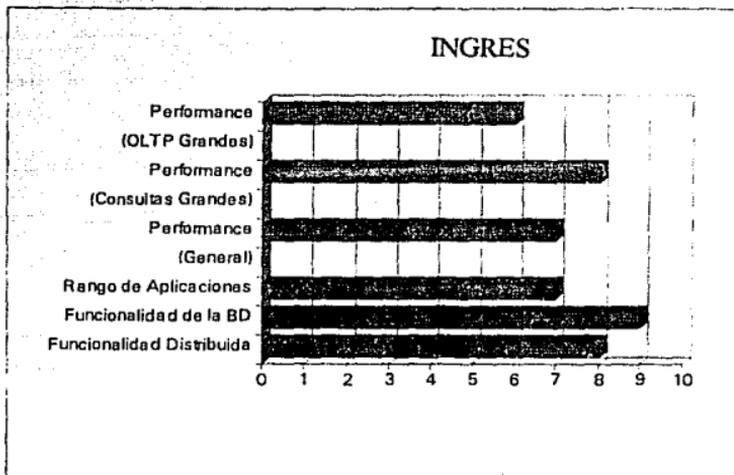
diccionario de datos es almacenado como una serie de tablas. Finalmente, está la utilidad que chequea las estructuras de la base de datos (encabezados de página, ligas, etc.) y asegura que todas estén funcionando correctamente.

III.1.2 Elección de la herramienta

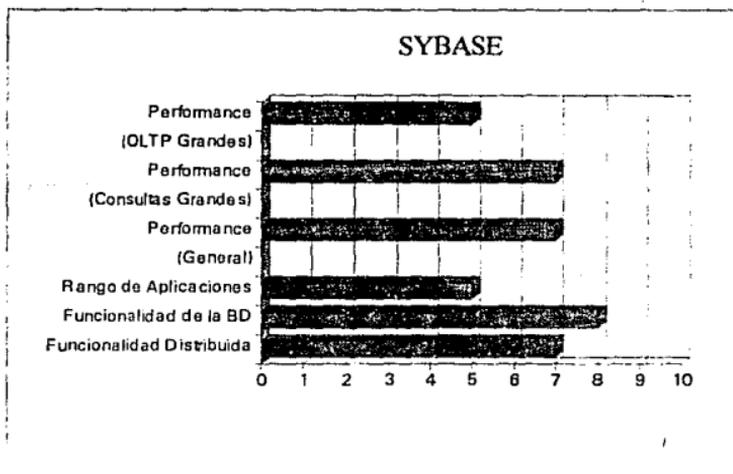
Para la elección de la mejor alternativa de solución, se tomaron en cuenta los criterios mencionados anteriormente considerando para cada uno sus ventajas y desventajas. Los resultados obtenidos se muestran en las siguientes gráficas y tablas comparativas:



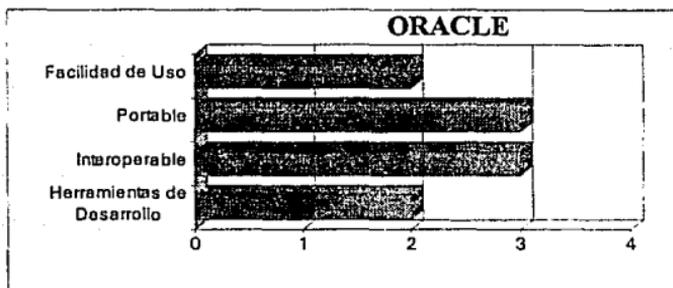
Gráfica III.1.2.1 Características de Performance y Funcionalidad de ORACLE



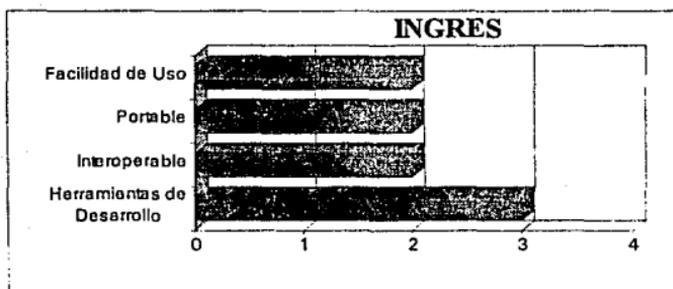
Gráfica III.1.2.2 Características de Performance y Funcionalidad de INGRES



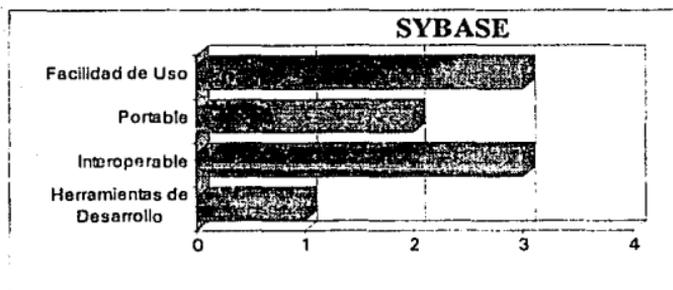
Gráfica III.1.2.3. Características de Performance y Funcionalidad de SYBASE



Gráfica III.1.2.4. Características de manejo de ORACLE



Gráfica III.1.2.4. Características de manejo de INGRES



Gráfica III.1.2.4. Características de manejo de SYBASE

FUNCIONALIDAD DISTRIBUIDA	ORACLE	INGRES	SYBASE
Acceso de lectura remota	SI	SI	SI
Acceso de Actualización remota	SI	SI	SI
Protocolo Two-Phase Commit	SI	SI	SI
Uniones sobre múltiples BD's (un mismo nodo)	SI	SI	SI
Uniones sobre múltiples BD's (dos o más nodos)	SI	SI	SI
Vistas sobre múltiples BD's	SI	SI	SI
Actualización de vistas sobre múltiples BD's	NO	NO	SI
Acceso transparente a la BD	SI	SI	SI
Definición de múltiples BD's físicas como una sola BD lógica en el Diccionario de datos	SÓLO EN EL DICCIONARIO DEL CASE	SI	NO
Referencia remota a BD's físicas a través de triggers	SI	NO	SI
Replicación automática de datos	SI	SI	NO
Transparencia en la replicación de datos	SI	SI	NO
Arquitectura distribuida de igual a igual ó maestro/ esclavo	IGUAL A IGUAL	MAESTRO/ ESCLAVO	IGUAL A IGUAL

Tabla III.1.2.1. Funcionalidad distribuida

CPU Y MEMORIA	ORACLE	INGRES	SYBASE
Uso de tipos de datos nativos del hardware	NO	SI	SI
Soporte de SQL dinámico	SI	SI	SI
Compilación de procedimientos almacenados.	SI	SI	SI
Configuración de múltiples procesos del servidor.	SI	SI	SI
Asignación específica de usuarios a los procesos del servidor.	SI	SI	SI
Soporte de multiprocesamiento simétrico.	SI	SI (Vía una arquitectura de multiservidores multihebra)	SI Arquitectura de servidor virtual
Compresión de datos.	NO	SI	SI
Compresión de índices	SI	SI	SI
Grupo de Commits.	SI	SI	SI
Otros.	Commits Rápidos (Asíncrono)	Commits Rápidos, Caché del servidor compartido.	-

Tabla III.1.2.2. CPU y Memoria

E/S A DISCO	ORACLE	INGRES	SYBASE
Liaves mezcladas	SI	SI	SI
Indices multinivel	SI	SI	SI
Listas invertidas	NO	NO	SI
Datos agrupados en base a las llaves	SI	SI	SI
Diseño de tablas para compartir páginas de datos	SI	NO	NO
Colocación de tablas en discos específicos	SI	SI	SI
Partición horizontal de tabla	SI	SI	SI
Partición vertical de tabla	NO	SI	SI
Variación del tamaño de la página de una tabla	NO	NO	NO
Variación de la página de un índice	NO	NO	NO
Número variable de buffers por tabla	SI	SI	NO
Número variable de buffers por base de datos	SI	SI	NO

Tabla III.1.2.3. Entrada/Salida a Disco

FUNCIONALIDAD	ORACLE	INGRES	SYBASE
TRIGGERS			
Soporte de Triggers	SI	SI	SI
De Inserción	SI	SI	SI
De Borrado	SI	SI	SI
De actualización	SI	SI	SI
Actualización de una columna específica	SI	SI	SI
Commit Trigger	SI	NO	NO
En Cascada	SI	SI	SI
Disparo de sentencias SQL (DML)	SI	SI	SI
Disparo de sentencias SQL (DDL)	SI	NO	SI
Disparo de múltiples sentencias SQL	SI	SI	SI
Invocar procedimientos almacenados	SI	SI	SI
Inclusión de lógica de procedimiento	SI	SI	SI
Disparo de procedimientos externos	SI	SI	SI

Tabla III.1.2.4. Funcionalidad Triggers

FUNCIONALIDAD	ORACLE	INGRES	SYBASE
PROCEDIMIENTOS ALMACENADOS			
Soporte de procedimientos almacenados	SI	SI	SI
Almacenados como objetos de la base de datos	SI	SI	SI
Almacenados como archivos ejecutables externos a la base de datos	NO	NO	SI
Traídos y editados por los usuarios de la base de datos	SI	SI	SI
Ejecutados vía trigger o llamada a un programa	SI	SI	SI
Ejecución directa del trigger por el usuario	SI	SI	SI
Capacidades del lenguaje procedural	SI	SI	SI
Nivel máximo de anidamiento	SI	SI	16
Soporte para arreglos de procesamiento	SI	SI	SI
Soporte para SQL DDL	NO	NO	SI
Soporte para SQL DML	SI	SI	SI
Soporte de cursores SQL	SI	SI	NO

Tabla III.1.2.5. Funcionalidad Procedimientos Almacenados

SEGURIDAD DE USUARIO EN LA BASE DE DATOS	ORACLE	INGRES	SYBASE
Nivel base de datos	SI	SI	SI
Nivel tabla	SI	SI	SI
Nivel de campo	SI	SI	SI
Nivel campo por valor	SI	SI	SI
Seguridad por grupo de usuarios	SI	SI	SI
Nivel de vista	SI	SI	SI
Privilegios para crear procedimientos almacenados	SI	SI	SI
Privilegios para ejecutar procedimientos existentes	SI	SI	SI
Privilegios para alterar y desechar procedimientos existentes	SI	SI	SI

Tabla III.1.2.6. Seguridad de usuario en la BD

RECUPERACION	ORACLE	INGRES	SYBASE
Recuperación hacia adelante	SI	SI	SI
Recuperación hacia atrás	SI	SI	SI
Configuración de auto commit para SQL	SI	SI	Default
Deshabilitar registro de transacciones	NO	SI	NO
Respaldo de la base de datos y registro de transacciones simultáneamente	No aplicable		SI
Resplado de la base de datos y registro de transacciones por separado	SI	SI	SI
Resplado en línea	No aplicable	SI	SI
Recuperación de una porción seleccionada del respaldo	SI	SI	NO

Tabla III.1.2.7. Recuperación

UTILERIAS	ORACLE	INGRES	SYBASE
Monitoreo en línea de los bloqueos en la base de datos	SI	SI	SI
Monitoreo en línea de las Entradas/Salidas a disco	SI	SI	SI
Monitoreo en línea de verbos	SI	NO	NO
Monitoreo en línea de la memoria	SI	SI	No aplica
Reorganización de los datos en línea	SI	SI	SI

Tabla III.1.2.8. Utilerías

Como se puede apreciar en las comparaciones anteriores, la opción que mejor cumplió con los criterios especificados es el Rdbms INGRES, pero debido al deficiente soporte técnico y comercial ofrecido en el momento por la empresa que lo representa, fue descartado y entonces se analizaron las dos opciones restantes.

De esta forma, ORACLE7 y SYBASE se analizaron y se encontró que ambos DBMSs estaban muy parejos en cuanto a sus características y facilidades ofrecidas, por lo que fue difícil la elección; debido a esto se recurrió al factor Costo-Beneficio en donde ORACLE7 resulto ser más costoso que SYBASE ofreciendo los mismos beneficios, motivo por el cual este último fue seleccionado como la herramienta estándar para desarrollo de las aplicaciones para la automatización de los procesos administrativos del Grupo Peñoles.

III.2 APLICACIONES DE DESARROLLO EN SYBASE

III.2.1 ISQL (Inter Active SQL)

Es una utilidad invocada a nivel de línea de comando del sistema operativo. Sus principales funciones son las siguientes:

- Manejo de las conexiones al servidor.
- Envío de sentencias SQL al servidor.
- Despliegue de resultado de consultas y mensajes en la pantalla.
- Interfase con el editor de texto del sistema operativo.

En la figura III.2.1.1 se muestra la conexión entre el SQL Interactivo y los Servidores SQL.

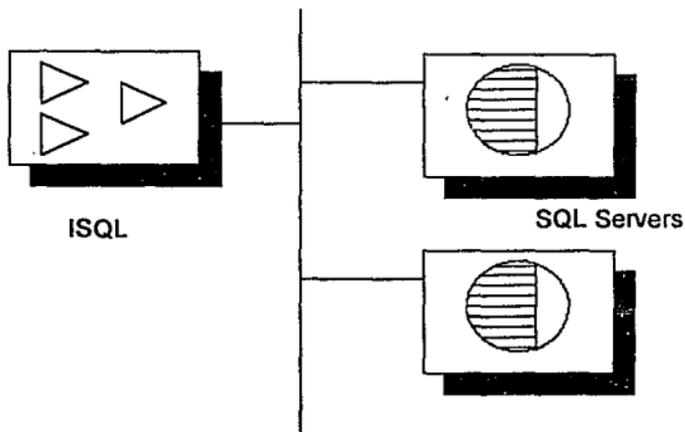


Figura III.2.1.1 Conexión entre el ISQL y los Servidores SQL

Comandos de ISQL.

- **Comandos de Uso Típico.**

```
isql -Uusername -Ppassword
```

```
> (sentencias sql)
```

```
go
```

```
> exit ó quit
```

En VMS, substituir "/U=" por "-U", ejemplo

```
isql /U=username
```

- **Comandos Comúnes.**

Para terminar un proceso batch y enviarlo al servidor se ejecuta el siguiente comando:

```
go
```

Para limpiar un comando y empezar de nuevo:

`reset`

Para llamar al editor y editar el siguiente conjunto de comandos o instrucciones:

`vi` (o `edit` para VMS)

Para terminar una sesión y desconectarse desde el servidor:

`quit`

ISQL y Archivos Script

- Los archivos Script son archivos de texto que contienen sentencias SQL que pueden ser sometidas al servidor usando ISQL.

Estos son usados principalmente para almacenar comandos usados para crear bases de datos, crear objetos y añadir cuentas de usuarios.

- Los archivos script pueden contener múltiples batches (secuencias de sentencias SQL).

Cada batch debe de ser terminado con la sentencia `go`.

Ejemplo:

Archivo Script OS

Primer

Batch

Segundo

Batch

```
if existe (select * from master.dbo.sysdatabases
           where name = "pubs")
begin
           drop databases pubs
end
go
print 'Creando la base de datos "pubs"'
create database pubs
go
```

- Como correr un Script.

UNIX:

isql -Uname -Ppwd < nombre de archivo

VMS:

isql /user=name /password=pwd /input=nombre de archivo

Lo anterior puede abreviarse como sigue:

isql /u=name /p=pwd /inp=nombre de archivo

Errores en los Archivos Script

- Si un batch en un archivo script tiene un error, dicho batch es abortado. Los batches subsecuentes en el script no son abortados al menos que éstos también tengan errores.

Ejemplo:

Archivo Script OS	
Primer Batch	<pre>select titulo_id, precio from titulos select avg(precio) form titulos go</pre>
Segundo Batch	<pre>update titulos set precio=precio*1.1 where precio < 30 select avg(precio) from titulos go</pre>

En este script, el primer batch tiene un error de sintaxis por lo que ninguno de los comandos en el batch será ejecutado.

Los comandos en el segundo batch, sin embargo, serán ejecutados.

- Cada procedimiento almacenado debe ser precedido por la sentencia *execute*.

III.2.2 DATA WORKBENCH

Data Workbench es un conjunto de herramientas basadas en ventanas que ofrecen una interfase con el servidor SQL, entre las cuales se incluyen las siguientes:

SQL Editor: Es un método simple para someter sentencias SQL al servidor y proporciona la habilidad para editar fácilmente dichas sentencias.

VQL Editor: Es una interfase visual para la generación de comandos SQL

Results Editor: Se usa para formatear los resultados después de que son obtenidos.

Report Workbench: Se usan para la generación y formateo de reportes.

Data Entry: Es usado para la creación simple, modificación y borrado de registros en una tabla.

Data Dictionary: Es usado para obtener información acerca de los objetos de la base de datos.

History List: Se usa para la recuperación de comandos SQL desde una sesión.

Help Facility: Contexto sensitivo de ayuda en línea.

Copy Table: Es un método para ejecutar Bulk Copy (Copias Masivas).

¿Cuándo debe de usarse Data Workbench?

- Cuando los resultados de las consultas necesitan formatearse.
- Cuando se requiere la historia de lo que se ha realizado.
- Cuando se requiere hacer entradas simples de datos.
- Cuando se requiere usar Report Workbench.
- Cuando se requiere usar VQL como ayuda en la estructura de las consultas.
- Cuando se requiere observar información del diccionario de datos.

En la figura III.2.2.1 se observa a Data Workbench como un cliente dentro de la arquitectura Cliente/Servidor.

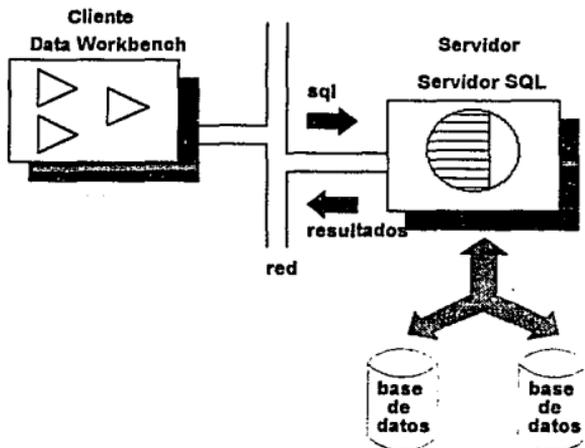


Figura III.2.2.1 Data Workbench como cliente

III.2.3 REPORT WORKBENCH

- Report Workbench es una herramienta poderosa para crear reportes empleando información de la base de datos.
- Características:
 - Puede crear reportes complejos, usando renglones ordenados, datos procesados y más.
 - Se pueden personalizar reportes con los parámetros de usuario proporcionados.
 - Incluye herramientas poderosas y dinámicas para el formateo de la presentación.
 - Permite múltiples consultas y consultas anidadas.
 - Se puede utilizar cualquier sentencia Transact-SQL.

Ejemplo de un reporte simple:

Aug 7 1993				Start of report			
título_id	título	tipo	pub_id	precio	Anticipos	real	Ventas
BU1032	The Busy Executive's Database Guide	business	1389	19.99	5,000.00	18	4095
BU1111	Cooking with Computers: Surreptitious Balance Sheets	business	1389	11.95	5,000.00	10	3876
BU2075	You Can Combat Computer Stress	business	0736	2.99	10,125.00	24	18722
BU7832	Straight Talk About Computers	business	1389	19.99	5,000.00	10	4095

III.2.4 BULLK COPY

Bulk Copy (Copia Masiva) es una copia de alta velocidad de datos entre una tabla de una base de datos de un archivo ASCII o binario.

Esto es una alternativa de alto rendimiento para usar Inserts y Selects SQL cuando se está tratando en grandes volúmenes de datos.

También es usado para intercambiar datos con aplicaciones que no son de la base de datos.

Son tres las formas para acceder el Bulk Copy:

bcp	programa único (stand-alone)
DWB/Copy Table	Corre bcp con una interfase visual
BCP-Library	Librería de subrutinas para programadores

Las tres tienen el mismo performance.

Bulk Copy Performance.

- Es equivalente a la velocidad de la sentencia SQL *'select into nombre_tabla...'* si el host (Computadora Anfitriona) puede proveer la capacidad de rapidez en el manejo de datos.
- La copia es muy rápida dentro de una tabla indexada si los datos ya están ordenados en base al índice.
- Si la tabla no está indexada de ninguna forma, el Bulk Copy estará en un modo de alta velocidad.

Los cambios no son accesados si las tablas no contienen índices.

La recuperación es firmemente garantizada, todo o nada.

El dbo deberá estar en master y ejecutar

```
sp_dboption database_name, 'select into/bulkcopy', true.
```

y entonces correr un checkpoint en la base de datos antes de que se pueda correr el bulk copy en el modo de alta velocidad.

- Los defaults son siempre ejecutados.
- Las reglas y los triggers no son ejecutados en un bulk copy
Checa la consistencia de los datos después de una copia.
- Se deberá descargar la base de datos después de un bulk copy, debido a que el sistema no podrá recobrar transacciones.

III.2.5 APT WORKBENCH

Podemos considerar a APT Workbench como el componente principal de Sybase, ya que es un conjunto de herramientas para el desarrollo rápido de aplicaciones basadas en formas, sin la necesidad de usar código de lenguajes de tercera generación (3GL). Esto permite crear aplicaciones en una de dos formas:

- Se podrá construir una aplicación completa en 4GL presionando un botón, y entonces modificar el código 4GL resultante si así se desea.
- Se podrá escribir todo el código 4GL desde el editor.

Se puede diseñar todas las formas, escribiendo y depurando todo el código 4GL, modificar el código y probar la aplicación, sin la necesidad de dejar el editor de formas.

Son 4 las herramientas que APT Workbench ofrece y que a continuación se describen:

- **APT-Edit:** Este módulo corresponde al editor de formas, el cual es usado para todas las aplicaciones basadas en estas formas. Este permite diseñar las formas, asignar procesos y probar la corrida de una aplicación.
- **APT-SQL:** Es un lenguaje de 4ta. generación especialmente diseñado para trabajar con APT-Edit.
- **APT-Build:** Es una herramienta que genera automáticamente una aplicación APT-SQL desde una forma creada en el APT-Edit.
- **APT-Execute:** Es un programa que corre una aplicación creada con APT Workbench. Este consiste en un programa principal y todas las rutinas de soporte necesarias para ejecutar dicha aplicación.

La integración del conjunto de herramientas de Sybase con el servidor SQL se observa en la figura III.2.5.1.

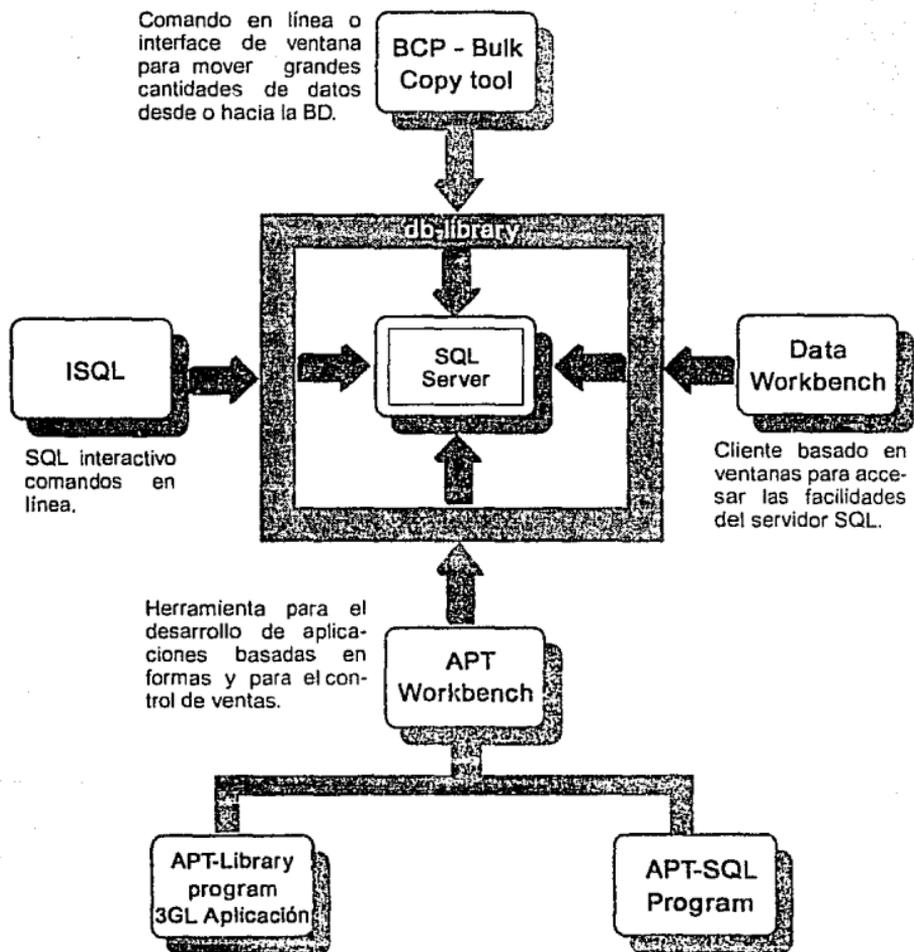


Figura III.2.5.1. Conjunto de herramientas SYBASE

APT Workbench, APT-Library y DB-Library

APT-Library: es una librería de llamadas al lenguaje anfitrión (3GL) que manejan las formas construidas en APT-Edit.

DB-Library: es una librería de llamadas al lenguaje anfitrión por el Servidor SQL.

APT Library y DB-Library no son parte de APT Workbench. Se puede escribir la mayoría de las aplicaciones sin la necesidad de usar estos, aunque si ya se tienen aplicaciones escritas con APT-Library y DB-Library que se quieran mejorar usando APT SQL, o si se prefiere escribir una aplicación en modo mixto(3GL y 4GL), se pueden usar APT-Library y DB-Library junto con APT-SQL en la misma aplicación.

APT-SQL está construido en base a APT-Library y DB-Library, por lo que el uso de las librerías es invisible para el desarrollador de aplicaciones en APT Workbench.

Entre otras cosas, APT Workbench ofrece los siguientes programas de uso individual (Stand-Alone):

- **aptcompile:** Es un compilador de código APT-SQL.
- **aptexec:** Es un programa que corre aplicaciones de APT-SQL.
- **convert:** Es un programa de conversión de formas en un estilo viejo a formas en el nuevo estilo APT-SQL.

A manera de describir de una forma rápida como trabajan todos los componentes de APT Workbench, podemos decir que lo primero que se tiene que hacer es el diseño y el desarrollo de las formas usando APT-Edit, una vez creada las formas se escriben los procesos que estarán asignadas a cada una en APT SQL. Los procesos son creados y editados en un editor del sistema operativo, que para nuestro caso fue vi, uno de los editores de UNIX. Estos procesos son compilados usando APT Compile, y son corridos desde el APT-Edit o desde el aptexec. Por último el APT-Execute controla la comunicación hacia la forma y la comunicación hacia el servidor usando las librerías de Sybase, APT-Library y DB-Library.

Lo anterior se observa gráficamente en la figura III.2.5.2

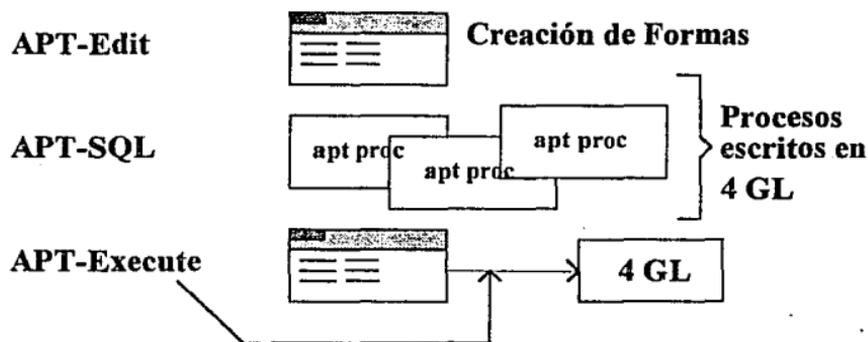


Figura III.2.5.2 Trabajo en conjunto de los componentes de APT Workbench

III.3 DISEÑO DEL SISTEMA

III.3.1 Módulos Principales (Diagramas de Jerarquía)

Un sistema está formado por varias actividades o procesos. En la programación de computadoras, los programadores con frecuencia desarrollan el software como una colección de módulos independientes pero que interactúan entre sí. Estos módulos se muestran en los diagramas de jerarquía.

Los diagramas de jerarquía de procesos continúan hasta los niveles que sean necesarios para identificar las actividades que forman parte del sistema. En general, deben de incluirse en el diagrama de jerarquía cualquier actividad que genere, modifique o utilice información. Lo común es que se necesiten, de acuerdo con la naturaleza del sistema, entre tres y siete niveles.

Antes de continuar, diremos que de acuerdo a lo que se mencionó en el Apartado III.1.1.8 referente al ambiente de desarrollo de Sybase, este puede disponer de un conjunto de herramientas Case. De esta forma para el diseño del sistema la herramienta Case utilizada fue IEW (Information Engineering Work Bench) V5.01, en la cual se elaboraron los diagramas que más adelante se presentan.

De esta forma, se tiene que el Sistema Integral de Abastecimientos está compuesto por los siguientes módulos principales.

Sistema Integral de Abastecimientos (Diagrama III.3.1.1)

- **Infraestructura (Diagrama III.3.1.2)**
 - Actualización de Catálogos

- **Requisiciones (Diagrama III.3.1.3)**
 - Validación y Relación de la Requisición
 - Asignación de la requisición por área
 - Actualización de Requisiciones
 - Actualiza Requisición
 - Registro Automático
 - Reportes y Consulta de requisiciones
 - Requisiciones recibidas
 - Situación de Requisiciones

- **Cotizaciones (Diagrama III.3.1.4)**
 - Análisis de mercado
 - Elaboración de solicitudes de cotización
 - Solicitud de cotización
 - Actualización de cotizaciones
 - Tabla comparativa
 - Consultas y reportes
 - Situación partidas por cotizar
 - Situación de la Solicitud de Cotización
 - Seguimiento de la Solicitud de Cotización

Solicitud permiso de la cotización

- **Pedidos (Diagrama III.3.1.5)**

 - Colocación de Pedidos**

 - Selección del Proveedor

 - Actualización de Pedidos

 - Impresión de Pedidos

 - Acuse de recibo

 - Impresión de la modificación

 - Acuse de recibo de la modificación

 - Confirmación de la Orden

 - Autorizaciones Pendientes**

 - Seguimiento a pedidos**

 - Seguimiento a Proveedores

 - Consultas por producto

 - Pedidos por proveedor

 - Costo por proyecto**

 - Solicitud de Transportación**

- **Movimientos de Material (Diagrama III.3.1.6)**

 - Actualización de movimientos de material

 - Consulta de pedidos y sus movimientos

 - Actualización de modalidades

 - Actualización de Tráfico

 - Actualización de facturas

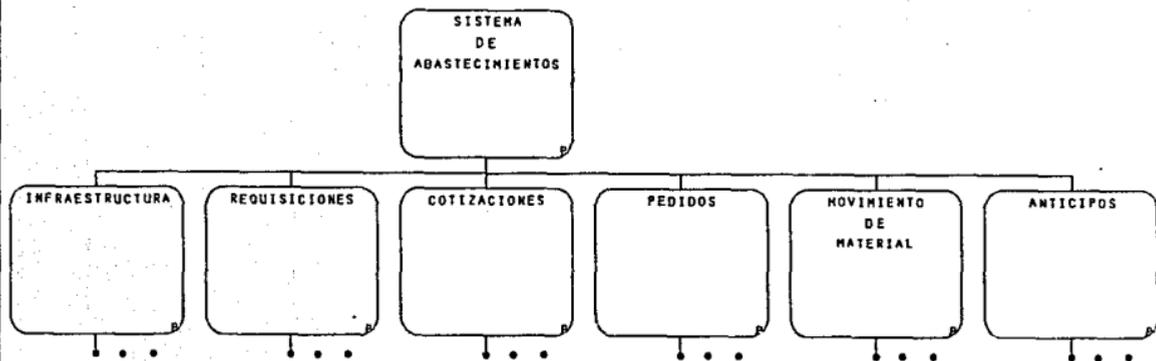


Diagrama III.3.1.1 Sistema Integral de Abastecimientos

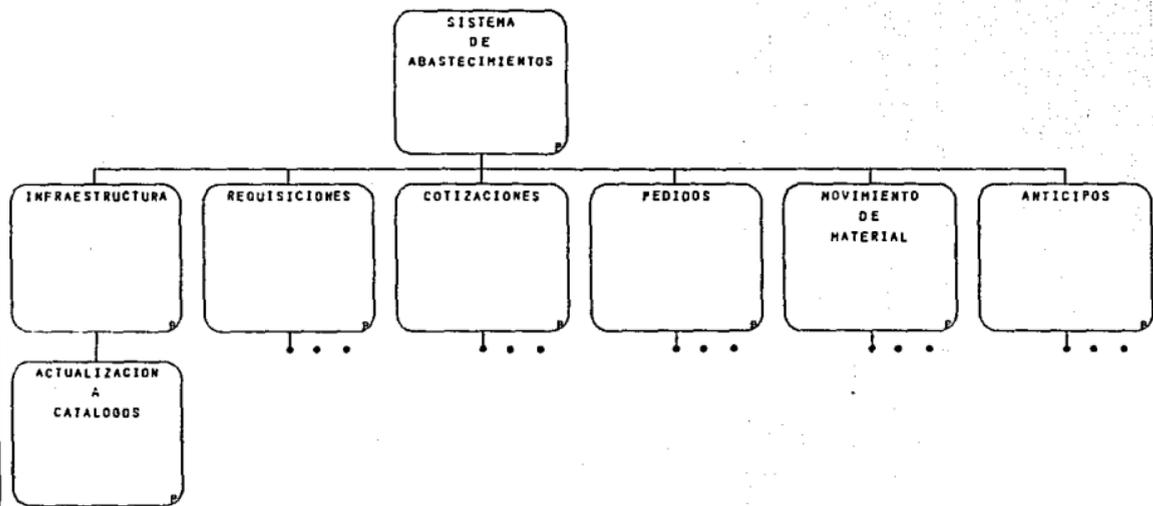


Diagrama III.3.1.2 Infraestructura

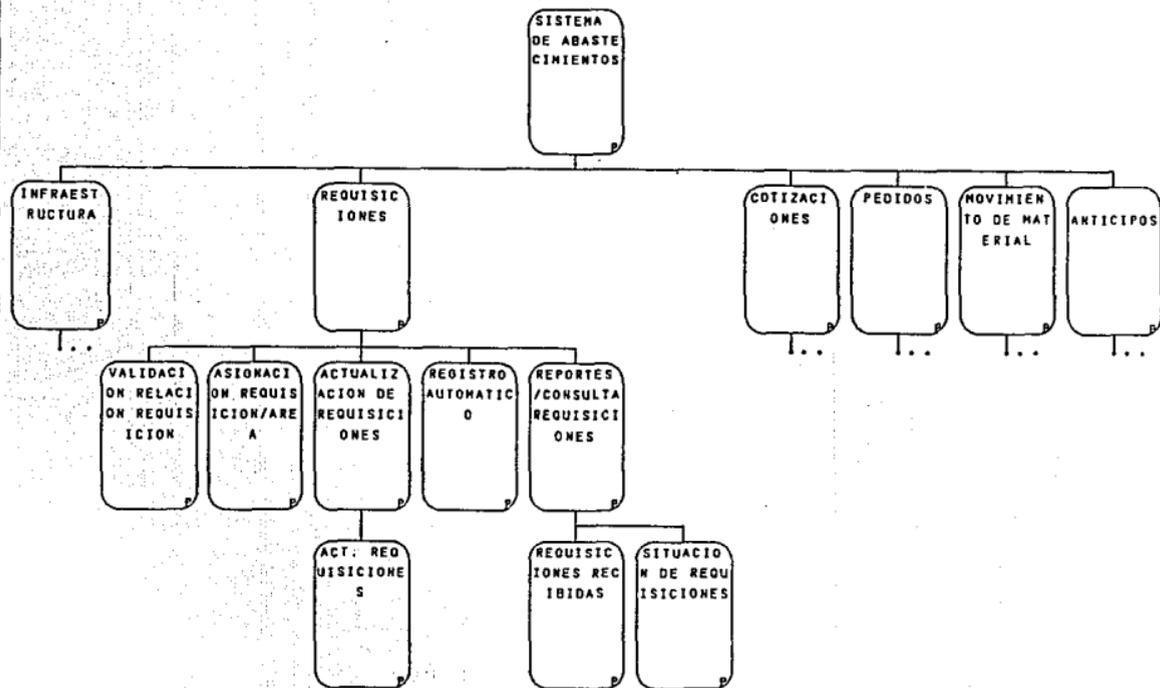


Diagrama III.3.1.3. Requisiciones

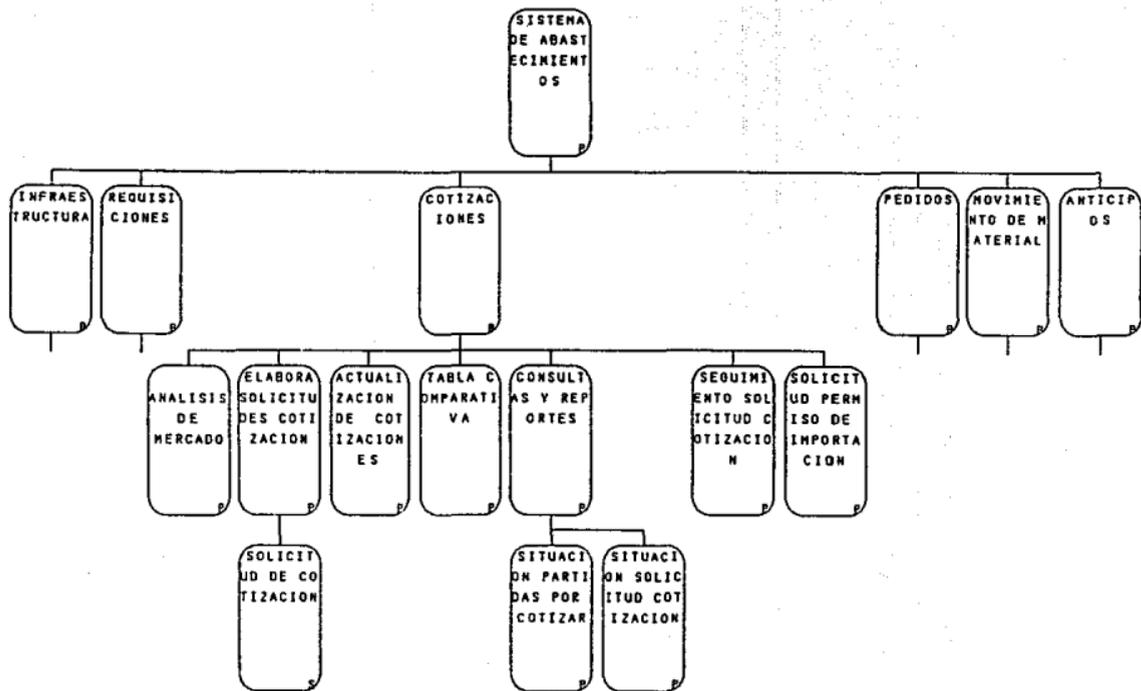


Diagrama III.3.1.4 Cotizaciones

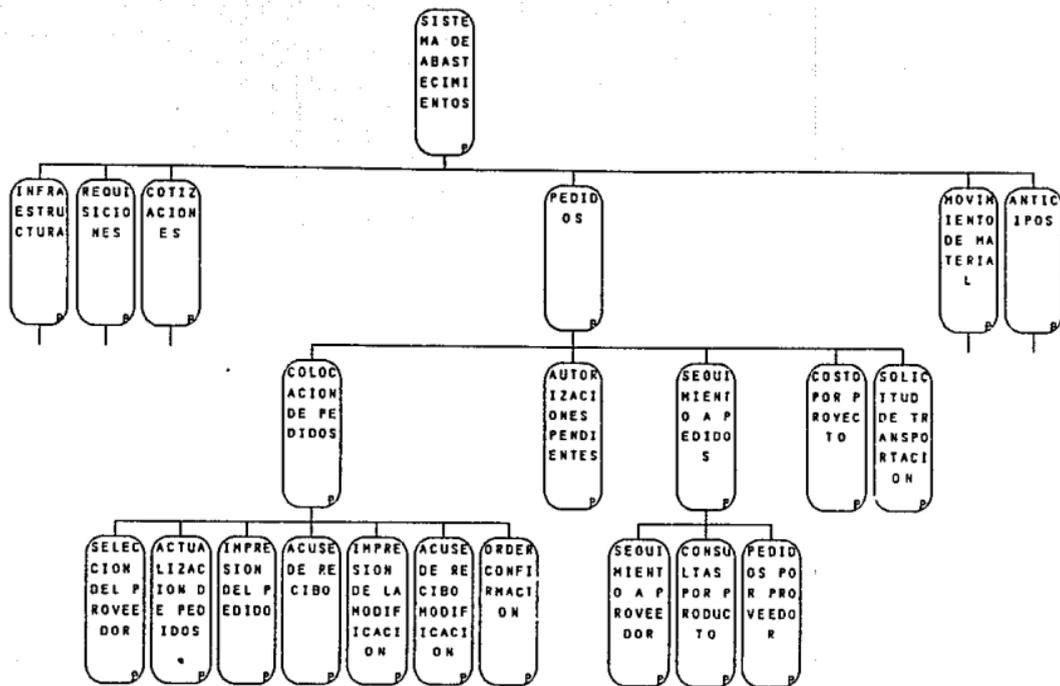


Diagrama III.3.1.5 Pedidos

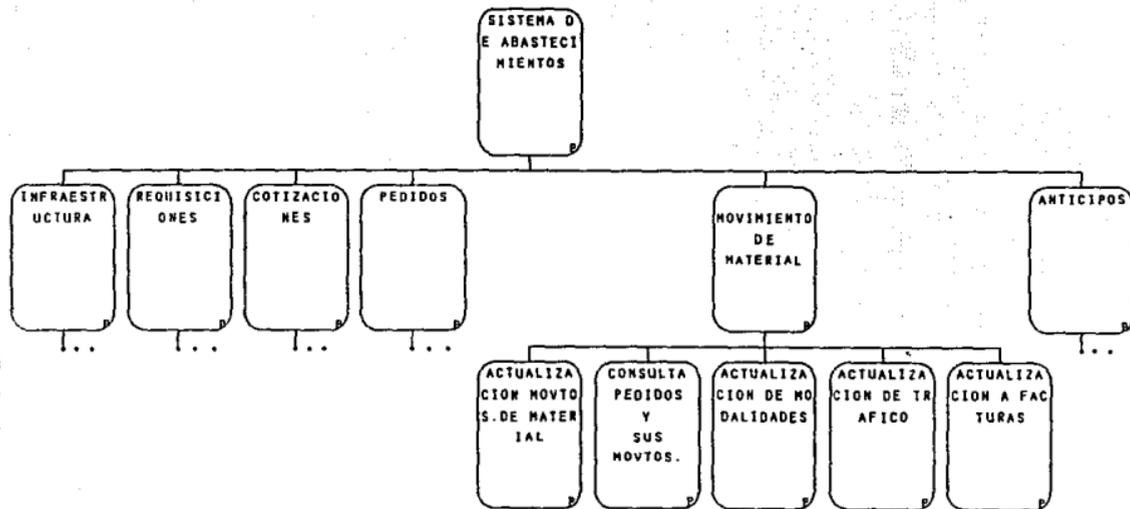


Diagrama III.3.1.6 Movimientos de Material

III.3.2 Diagrama de flujo de datos

El análisis del flujo de datos examina el empleo de los datos para llevar a cabo procesos específicos de la empresa dentro del ámbito de una investigación de sistemas. El análisis puede pensarse de tal manera que se estudien actividades del sistema desde el punto de vista de los datos: donde se originan, como se utilizan o cambian, hacia donde van, incluyendo las paradas a lo largo del camino que siguen desde su origen hasta su destino. De esta forma una de las herramientas más importantes en el análisis del flujo de datos son los diagramas de flujo de datos.

El diagrama de flujo de datos es una herramienta gráfica que se emplea para describir y analizar el movimiento de los datos a través de un sistema, ya sea que este fuera manual o automatizado incluyendo procesos, lugares para almacenar datos y retrasos en el sistema. La transformación de datos de entrada en salida por medio de procesos puede describirse en forma lógica e independiente de los componentes físicos asociados con el sistema.

Los diagramas de flujo de datos correspondientes al sistema de abastecimientos se muestran a continuación.

Sistema de Abastecimientos	Diagrama III.3.2.1
• Requisiciones	Diagrama III.3.2.2
• Cotizaciones	Diagrama III.3.2.3
• Pedidos	Diagrama III.3.2.4
• Movimientos de Material	Diagrama III.3.2.5

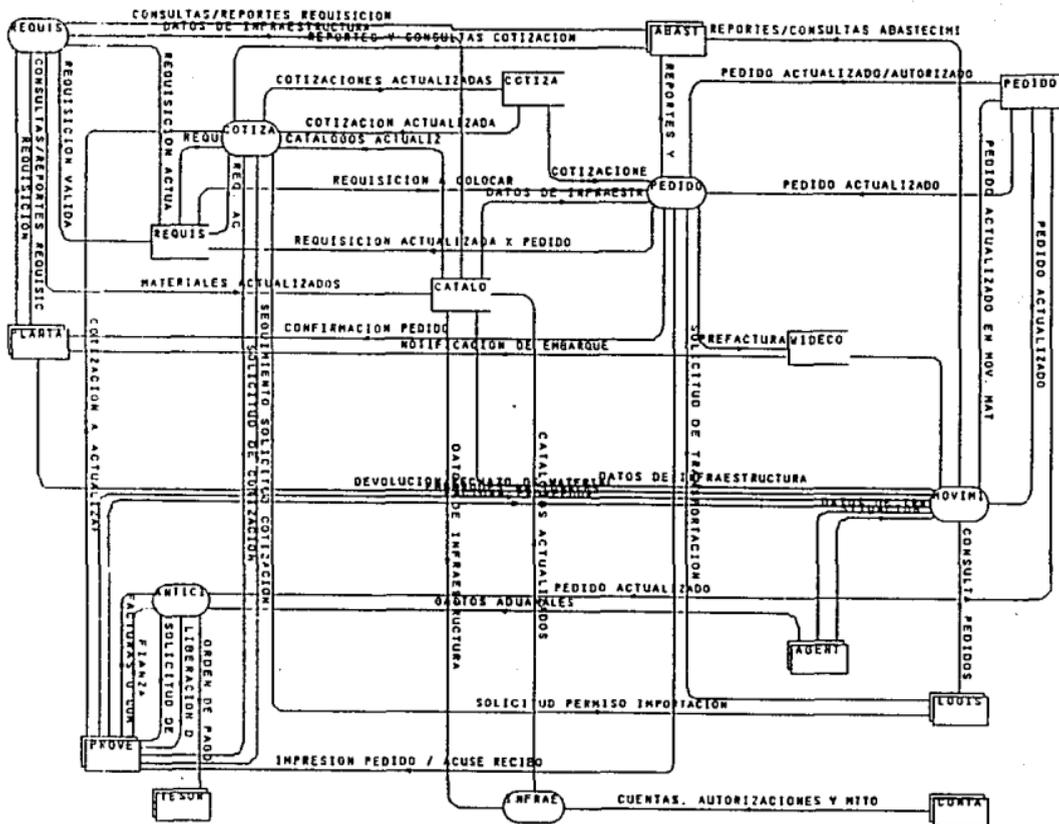


Diagrama III.3.2.1 Flujo de Datos del Sistema de Abastecimientos

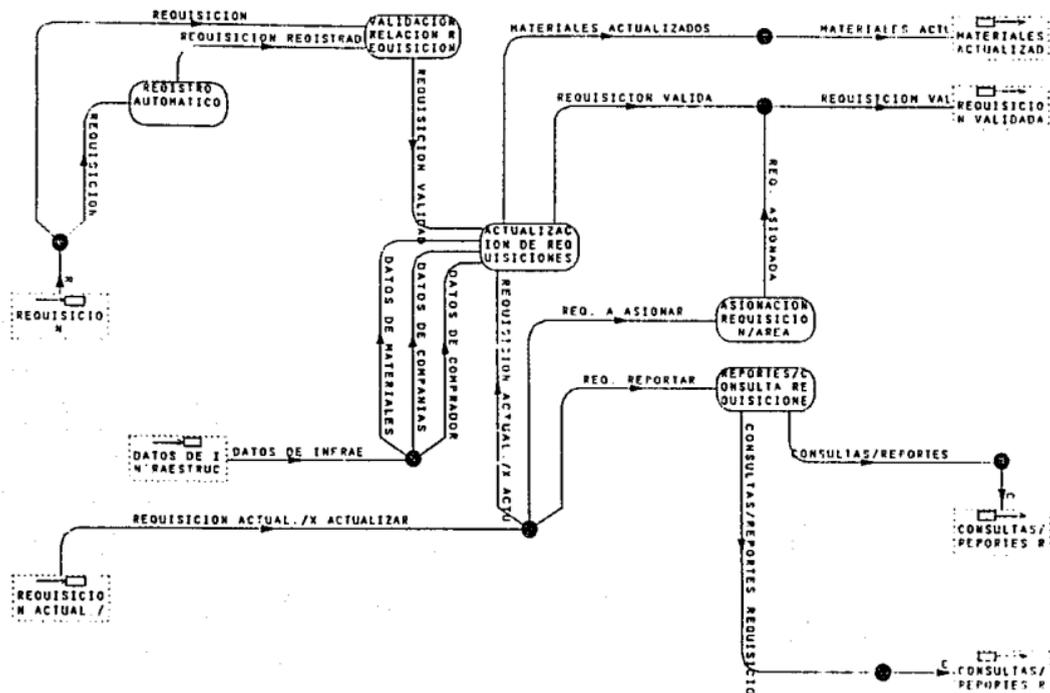


Diagrama III.3.2 Flujo de Datos de Requisiciones

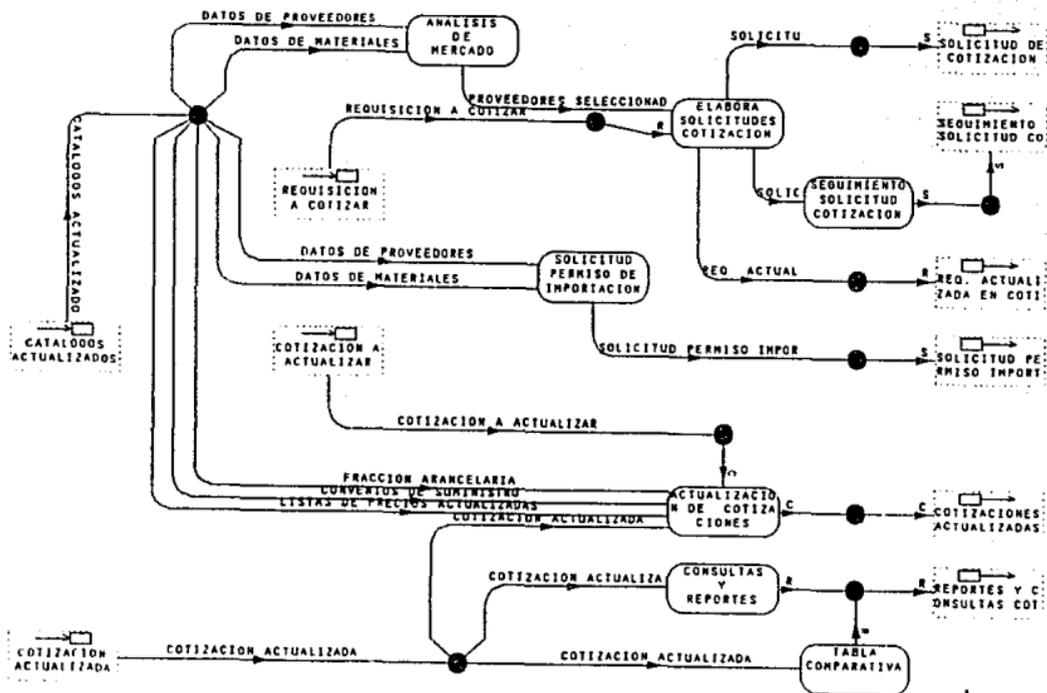


Diagrama III.3.2.3 Flujo de Datos de Cotizaciones

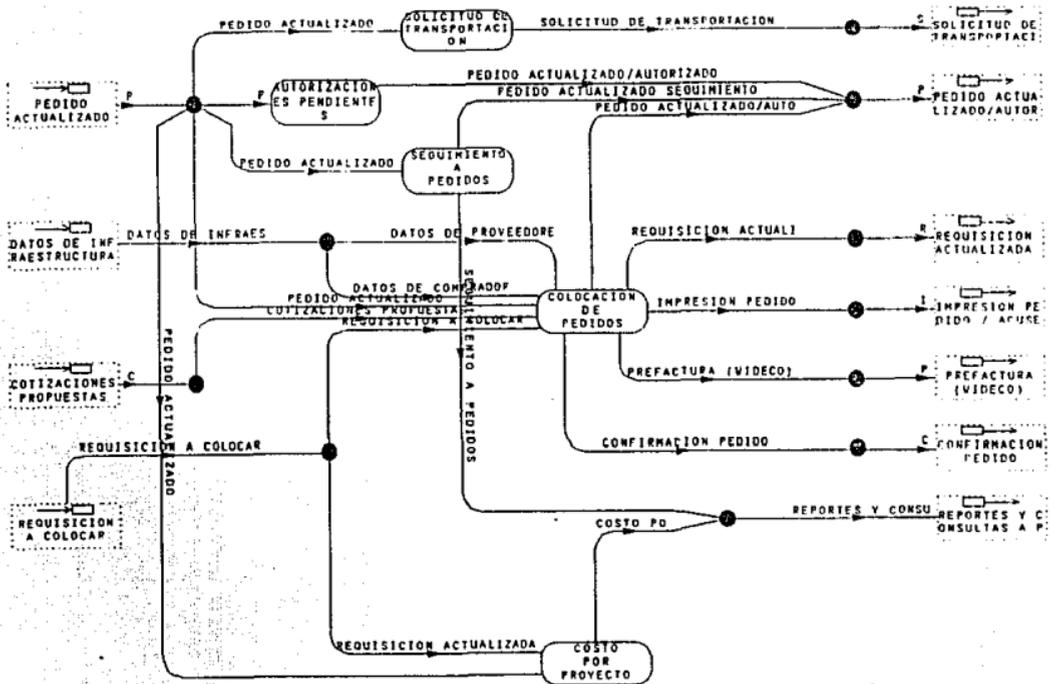


Diagrama III.3.2.4 Flujo de Datos de Pedidos

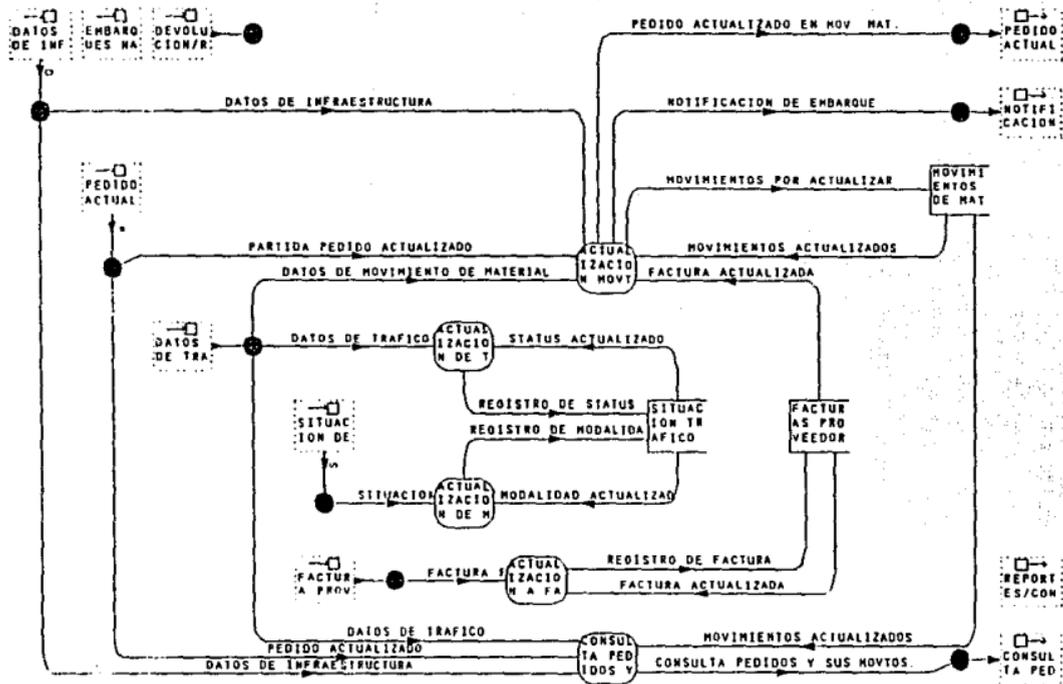


Diagrama III.3.2.5 Movimientos de Material

III.3.3 Diagrama Entidad - Relación

Los diagramas de entidad-relación son una descripción de la relación entre entidades de un sistema y el conjunto de información relacionado con la entidad. No consideran el almacenamiento físico de los datos.

Los diagramas más importantes de entidad-relación son los siguientes:

- **Requisiciones** Diagrama III.3.3.1
- **Cotizaciones** Diagrama III.3.3.2
- **Pedidos** Diagrama III.3.3.3
- **Movimientos de Material** Diagrama III.3.3.4

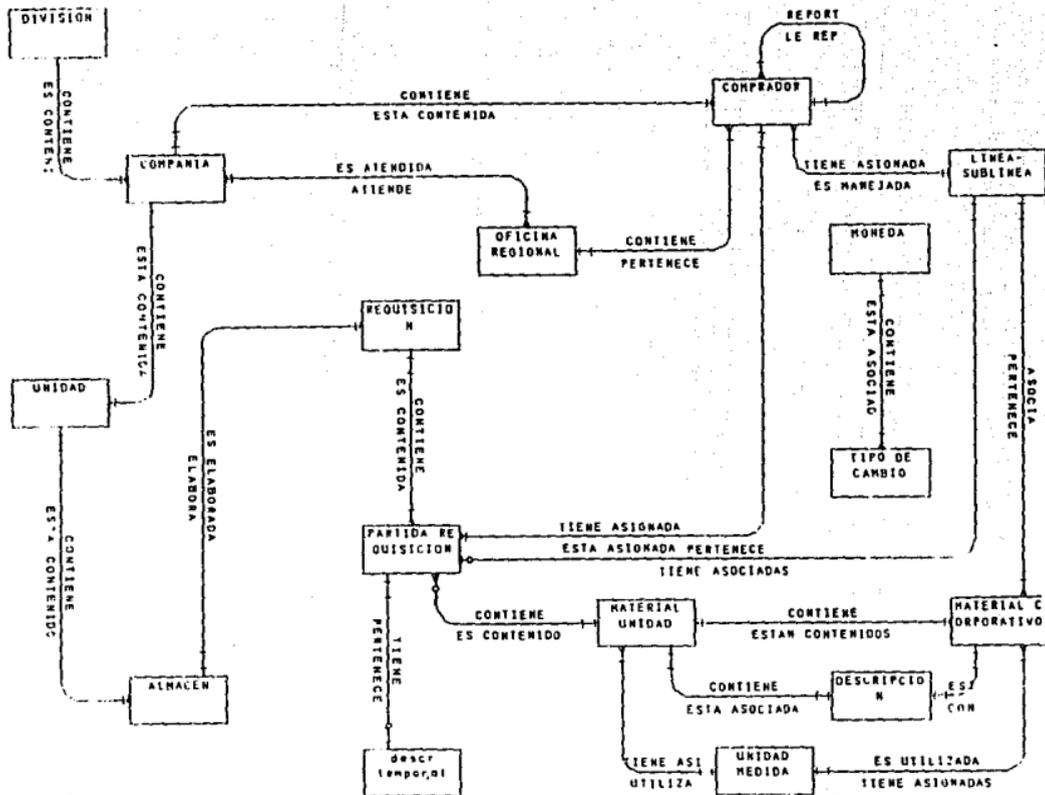


Diagrama III.3.3.1 Entidad-Relación Requisiciones

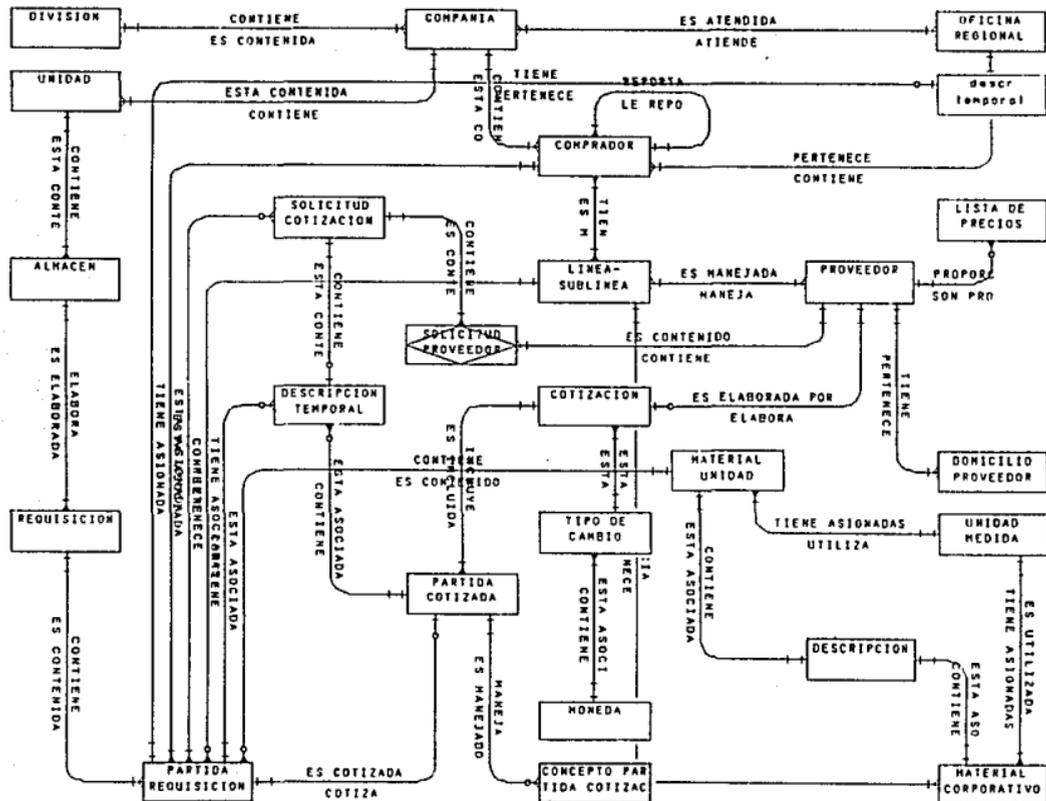


Diagrama III.3.2 Entidad-Relación Cotizaciones

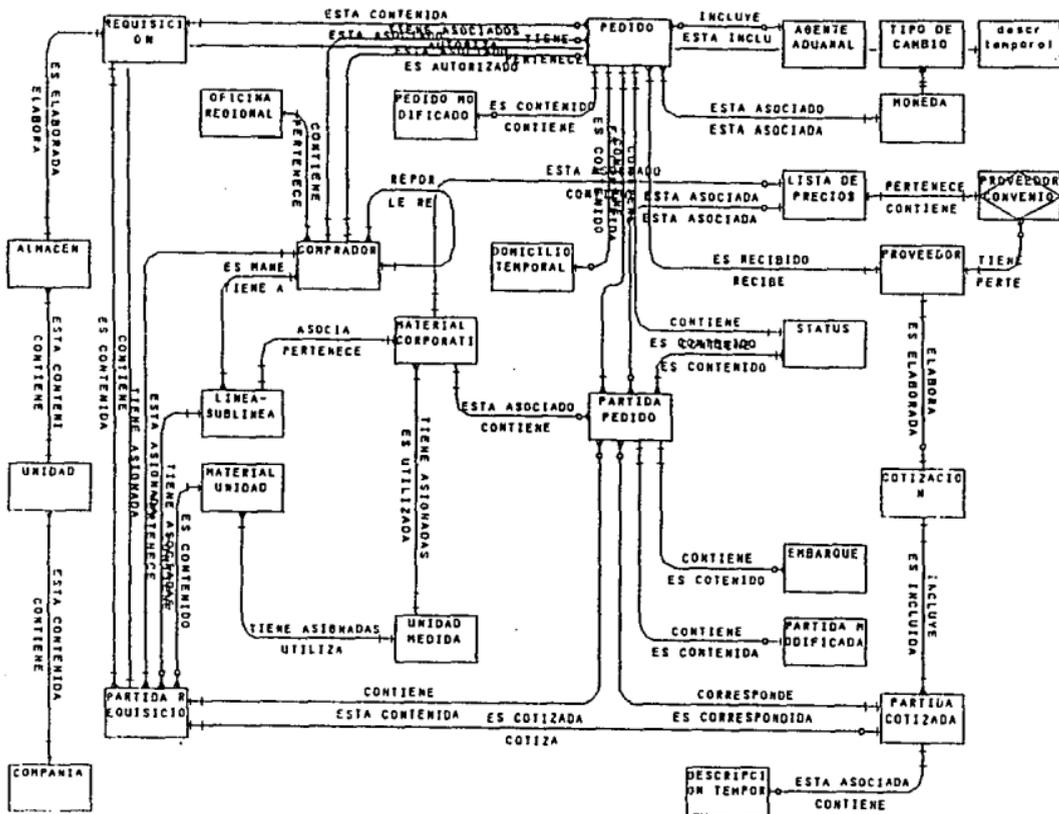


Diagrama III.3.3 Entidad-Relación Pedidos

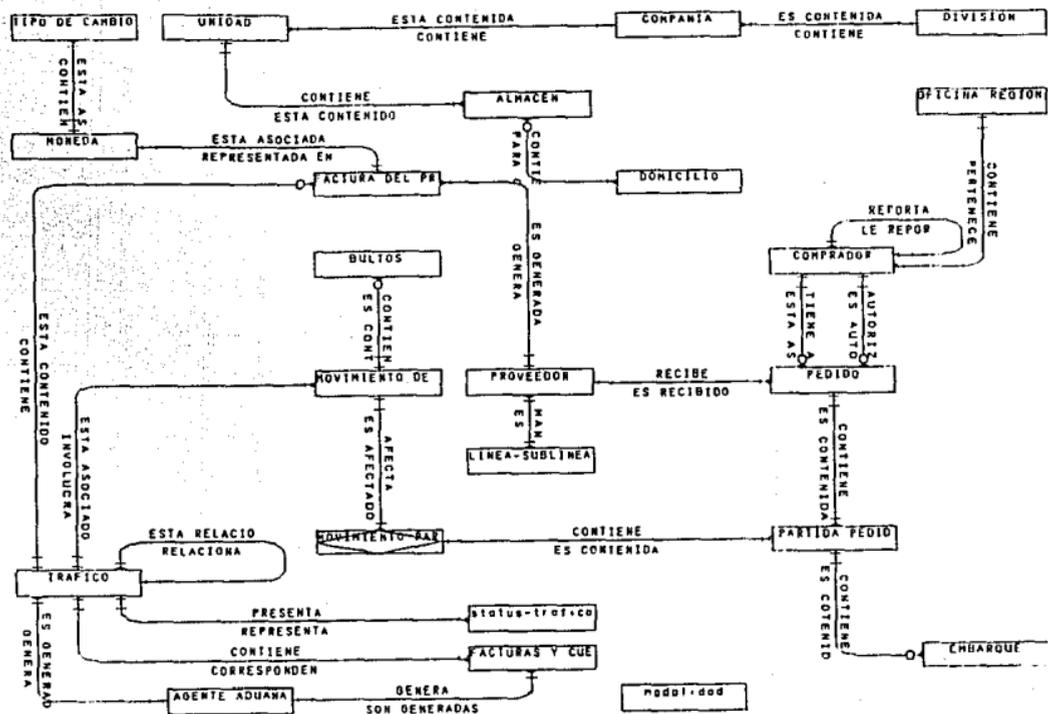


Diagrama III.3.4 Entidad-Relación Movimientos de Material

III.3.4 Diccionario de Datos

Tomando en cuenta los diagramas de entidades presentados en la sección anterior se definieron las entidades (tablas) que conformarían el diccionario de datos del sistema, las cuales se presentan en el Apéndice A.

III.3.5 Niveles de acceso a usuarios

La finalidad de definir diferentes niveles de acceso a los diferentes usuarios del sistema es tener un control sobre las personas que pueden usarlo y las funciones que estos pueden realizar dentro de este. Considerando que cierta información manejada por el sistema no puede ser alterada (por ejemplo los montos de pedidos) es necesario definir quien puede hacerlo, y que funciones le son permitidas a cada usuario (Directores, Gerentes y Compradores).

Para lograr este objetivo, es necesario definir los siguientes parámetros:

- Nombre de usuario
- Clave de acceso (password)
- Derechos de acceso (leer, modificar, borrar, agregar)

Estos parámetros deberán de ser definidos por el administrador del sistema, quien será el encargado de proporcionar a los demás usuarios los niveles de acceso que le correspondan. El administrador será el responsable de la actualización de los datos

relativos a seguridad y deberá hacer las modificaciones cada vez que este sea necesario.

Los permisos asignados a cada usuario del sistema son especificados por medio de los *grants*, los cuales son instrucciones especiales que determinan cuales usuarios pueden hacer modificaciones (insertar, modificar y borrar) sobre tablas específicas.

Los *grants* que establecen los niveles de acceso al sistema de abastecimientos se muestran en el Apéndice B.

III.4 DESARROLLO DEL SISTEMA

III.4.1 Programación

Como es bien sabido, la programación se refiere al diseño del software de los diferentes módulos del sistema. Para esto se tomaron en cuenta los principios de diseño que se muestran en la Tabla III.4.1.1.

PRINCIPIO	DESCRIPCION	OBJETIVOS
Modularidad y fragmentación	Diseño de un sistema como una jerarquía de módulos	Diseñar la estructura en forma descendente con módulos que realicen funciones específicas
Acoplamiento	La fuerza de las relaciones entre módulos	Maximizar la independencia entre los módulos minimizando el acoplamiento
Cohesión	La fuerza de las relaciones dentro del módulo	Maximizar la cohesión; los elementos altamente relacionados deben estar en el mismo módulo
Extensión de control	Número de módulos subordinados al módulo que hace la llamada	Limitar la extensión de control de cinco a siete módulos
Tamaño	Número de instrucciones que componen a un módulo	Limitar el tamaño de forma que la función de todo el módulo se centre en un solo propósito
Uso compartido	Uso de los módulos por otros módulos	Evitar la duplicación permitiendo que los módulos sean llamados por otros que necesitan la función de cada uno

Tabla III.4.1.1. Principios del diseño del Software

Por otra parte, de acuerdo a las necesidades de programación de los diferentes módulos del sistema se emplearon básicamente tres lenguajes diferentes para desarrollar los programas (ver Apéndice C), así los lenguajes utilizados fueron los siguientes:

- **APT Workbench (Transact-SQL)**

APT-Workbench se utilizó para desarrollar los programas basados en ventanas (Formas) y menús pull-down, los cuales constituyen la mayor parte de los programas del sistema. Las formas se encuentran por todo el sistema, desde la forma que ofrece acceso al sistema hasta la forma de impresión de reportes y actualización de catálogos.

En el Apéndice D se presentan las definiciones de las formas principales del Sistema de Abastecimientos (Menú Principal, Requisiciones, Cotizaciones, Pedidos y Movimientos de Material).

El estilo de las formas que se emplean en el sistema de abastecimientos son como las que se muestran en las figuras III.4.1.1 a la III.4.1.7.

- **SQR (Structured Query Report)**

SQR fue utilizado para el desarrollo de los programas que generan la mayoría de los reportes de los diferentes módulos (Requisiciones, Cotizaciones, Pedidos, Movimientos de Material.).

Algunos ejemplares de dichos reportes se encuentran en el Apéndice E.

```

+-----LOGISTICA Y ABASTECIMIENTO-----+
| /1 Compras Trafico Infraestructura / 0 |
+-----+
| Act./Cons.          -> || Requisicion      193 |
| Consulta            -> || Sol:cotizacion |
| Catálogos           -> || Cotizacion    |
| Reportes de Pedidos -> || Pedido:Colocar |
| Reportes de Requisiciones -> || Pedido:Modificar |
+-----+
|                               | Movimiento de Material |
+-----+

```



Figura III.4.1.1 Formato de Pantalla Menú Principal

```

+-----REQUISICIONES-----+
| / Actualiza Consulta Ptd! Notas! Menu E/Col! 0 |
+-----+
| Centro Requisicion |
| Usuario RB Raul Becerril H |
| Cta C. |
| Aut.D | Comprador |
| Monto | Fechas de: |
| Depto | Recb 931103 Partidas |
| Solic | Elab Colocadas |
| O.T. | Req Programadas |
| Equipo | E/Coloc Dias Notas |
| Folio | Tipo A (A,<A>,S) Estatus |
| | Prio C (A,B,<C>) Fecha |
| | A.Fijo N (S,<N>) |
| Linea | Descripcion Generica |
| Sublinea |
+-----+

```



Figura III.4.1.2 Formato de Pantalla Requisiciones

```

+-----SOLICITUD DE COTIZACION-----+
| / Act Movtos.  Prov. Req. Ant. Req. Sig. Sol Asoc. |
+-----+
|                                     Fecha de Solicitud 93110331 |
| Solicitacion No. _____ Requisicion |
| Fec. Lta. Cot. _____ Centro |
| Notas |
|                                     |
|                                     Partidas en solicitud |
| Descripción |
+-----+

```

Actu
lizarMo
ntosProve
dorReq
Ant.Req
Sig.Sol
Asoc.SALIDA
FB

Figura III.4.1.3 Formato de Pantalla Solicitud de Cotización

```

+-----Cotizaciones-----+
| / Act. Cons. Sol. Prou. Cant. C Sig. Sol Asoc. 0 |
+-----+
| Cve. Cotizacion _____ Fecha Cotizacion 931103 |
| Proveedor/RFC _____ |
| No. Solicitud _____ Requisicion _____ |
| Fecha Vigencia _____ Centro _____ |
| Moneda _____ Cond. Pago _____ |
| _____ Entrega _____ Descuento _____ |
| Cond. General % |
| Lugar Por Partida % a Dias |
| Tiempo Per Pto. Pago |
+-----+
    
```



Figura III.4.1.4 Formato de Pantalla Cotizaciones

```

-----Selección para orden de compra-----
/ Actualiza Consulta Posic. V/ /A/ Menu 0
-----
Comprodor RB Reul Bacerril H
Centro
Requis. De n
Cotizacion % Desc. Pedido Num.

Requis. St Cotizacion Vigen. Ven Proveedor

Selec. 0 / 0
Proveedor
----->

```



Figura III.4.1.5 Formato de Pantalla Pedidos Colocación

```

=====ACTUALIZACION DE PEDIDOS=====
/ Actualizo Consultas Ptd! Notas! D.C. Impresión Menu 0
-----
| Usuario
| Pedido          ----          SI
| Autoriza        Modif.          F. Elab.
|                                     Partidas
|                                     Notas
| Cotizacion
| Proveedor
| Attn.
| Forma de Pago   Días F. Pago   Desc. x P.P.   Días P.P.
| Con. Entrega   Lugar de Ent.
| Via de consig.  Don. consignar  Don. Fac.   C.S.   Agt.du.
| F. Emb.         Monedo        T.C.
|                                     Valor Bruto
|                                     Descuento
|                                     % IVA
|                                     Valor Neto
|                                     Imp. Pagado
=====
    
```

Actualiz F1	Consulta F2	Partidas F3	Notas F4	D.C. F5	Imprimir F6	Menu F7	SALIDA F8
----------------	----------------	----------------	-------------	------------	----------------	------------	--------------

Figura III.4.1.6 Formato de Pantalla Pedidos Modificación

-----MOVIMIENTOS DE MATERIAL-----

/	Movto	Partida	Consultas	Bultos	Facturas	0	
Tipo	Movimiento			Fecha		Cond. Flete	
Transporte				Guía			
Remitente				Compañía			
Factura	Fecha	Importe	Mon	Bul	Cant.	Tp	Peso U.H.
Ato							
Proveedor							
Part	Descripción			Cant.	Ev Ant.	Acumu	Cant. ST

----->

MOVIDOS

PARTIDAS

CONSULTA

BULTOS

FACTURAS

A_Descr

D_Descr

SALIDA

FB

Figura III.4.1.7 Formato de Pantalla Movimientos de Material

- Lenguaje "C"

El lenguaje C fue utilizado básicamente para el desarrollo de los programas que generan la impresión de las requisiciones, las solicitudes de cotización y los pedidos, además de algunos reportes, los cuales se muestran en el Apéndice E.

III.4.2 Pruebas y Ajustes

Al llegar a esta etapa en el desarrollo del sistema, se hizo una revisión del sistema conducida igualmente por los usuarios y los analistas. Esto no fue tan solo una práctica normal, sino un proceso formal que sirvió para determinar qué tan bien está funcionando el sistema, cómo ha sido aceptado y cuales ajustes son necesarios.

El interés fundamental durante la revisión fue determinar si el sistema cumplió su objetivo, es decir, que nosotros deseábamos saber si el nivel de desempeño de los usuarios había mejorado y si el sistema estaba produciendo el resultado deseado.

De esta forma, la prueba del sistema se realizó con la participación directa de los usuarios quienes debieron ejecutar procesos formales correspondientes al trabajo de cada uno, esto con la finalidad de probar todos los módulos y funciones del sistema. Al momento de estar probando se tomaron en cuenta aspectos como la calidad de la salida del sistema, la facilidad de uso y la confianza por parte del usuario, entre los principales.

Con lo que respecta a la calidad que fue la que mereció la mayor atención, se calificó la precisión de la información, lo oportuno de su presentación y lo apropiado del formato de las pantallas y reportes. Así en este punto se encontró durante la prueba que los usuarios en algún momento no sabían a que tipo de movimiento correspondía la pantalla en la que se encontraban debido a que faltaba que se les presentara datos informativos acerca de dicho movimiento; este problema se fue solucionando paulatinamente conforme los usuarios iban detectando las ausencias de información. También se encontró en un principio que el despliegue de las pantallas resultaba

relativamente lento, detectándose que la causa era el rendimiento de la máquina que se estaba utilizando; para solucionar esto se tubo que adelantar el cambio del modelo de la máquina que ya se tenía previsto. Por el lado de las pantallas y reportes, se encontró que los ajustes fueron mínimos, sin embargo conforme los usuarios se adentraban al sistema fueron detectando nuevas necesidades.

La facilidad de uso fue el siguiente punto a evaluar. Aquí se cuestionó si el sistema era fácil de usar, si las validaciones eran adecuadas para prevenir los errores de captura, detectarlos, reportarlos y corregirlos cuando estos ocurrieran. Así se observó que los usuarios se encontraron un poco desconcertados debido al cambio del sistema y por consiguiente a la navegación en comparación con el sistema anterior, lo cual es comprensible debido a que llevaban trabajando mucho tiempo con el sistema de SISINF. No obstante los usuarios hicieron comentarios acerca de que la navegación pudiera ser mejorada y es en lo que actualmente se está trabajando. Pero finalmente se puede decir que el sistema tiene una aceptable facilidad de uso ya que los usuarios lo aceptaron y no tuvieron mayor problema para adaptarse.

La confianza por parte del usuario también es un indicador de la calidad del sistema, ya que esta afecta su longevidad y uso. Para este punto se encontró en un principio que los usuarios no confiaban mucho en el sistema pero esto no impacto demasiado debido a que es una reacción natural en la implantación de cualquier nuevo sistema hecho en casa.

La revisión también fue muy importante para recabar información para el mantenimiento del sistema. Puesto que ningún sistema es en realidad totalmente

completo, el sistema permanecerá mientras no se requieran cambios debido a desarrollos internos, como nuevos usuarios o actividades de la empresa; o bien, situaciones externas. La revisión después de la implantación vino a ser nuestra primera fuente de información de los requisitos futuros de mantenimiento.

Finalmente, una vez realizadas las pruebas y hecho la mayoría de los ajustes, el sistema fue aprobado por el Subdirector y el Director de la Dirección de Logística y Abastecimientos. De esta forma, se procedió a la implantación formal del sistema.

III.5 IMPLANTACION Y EVALUACION

III.5.1 Capacitación

En esta etapa del Sistema, se realizó la presentación al usuario y el plan de capacitación.

Desde la fase del diseño, se fueron presentando los avances del Sistema al Subdirector del área de Abastecimientos, por lo que este no le resultó totalmente desconocido. La fase de presentación se centró en la parte operativa del sistema, donde los puntos que se tocaron fueron:

- Entrada al Sistema
- Menú Principal
- Infraestructura del Sistema
- Módulo de Compras
 - Requisiciones
 - Solicitud de Cotización
 - Cotización
 - Pedidos
 - Pedidos Colocación
 - Pedidos Modificación
- Generación de Reportes

Al terminar la presentación, se determinó el plan a seguir para la capacitación de los usuarios. Se llegó a la conclusión de que el nuevo sistema debía de operar en forma paralela con el anterior, esto se hizo con la finalidad de que el cambio para el usuario fuera pausado mientras se familiarizaba con el nuevo sistema y que pudiera evaluar las características de operación y manejo de el sistema anterior contra el nuevo.

Se les dio un entrenamiento básico, principalmente con la finalidad de que obtuvieran los conocimientos y habilidades necesarios para poder operar el nuevo sistema. Se les capacitó principalmente en los siguientes puntos:

-  Conocimiento de las partes del nuevo equipo y breve descripción de sus funciones
-  Forma de iniciar y entrar al Sistema
-  Menú Principal del sistema y navegación a través de sus opciones
-  Infraestructura del Sistema
-  Módulo de Compras
-  Elaboración, cambios y consultas de pedidos, cotizaciones, requisiciones y pagos.
-  Manejo de catálogos
-  Generación de Reportes

Para reforzar los conocimientos, o bien, para tener un medio de apoyo para cualquier duda, se llevó a cabo la elaboración del Manual de Usuario. En la construcción de este manual se consideró al usuario del sistema y se incluyeron aquéllos procedimientos

que son necesarios para la operación del mismo, pero que deban de ser realizados por el usuario.

El Manual de Usuario contiene los siguientes puntos:

-  Carátula
-  Introducción
-  Panorama global del sistema
-  Módulo de Compras
 - Requisiciones
 - Solicitud de Cotización
 - Cotización
 - Pedidos
 - Pedidos Colocación
 - Pedidos Modificación
-  Apéndice
-  Glosario

El Manual de Usuario se encuentra en el Apéndice F.

III.5.2 Evaluación de Resultados

De igual manera en que los componentes de hardware decreentan sus costos y la tecnología continua avanzando a pasos agigantados, el software requiere optimizarse para un mejor aprovechamiento. en secciones anteriores se describieron los procesos anteriores de pruebas y ajustes para la detección de problemas de funcionamiento y eficiencia. En esta última sección nos dedicaremos a la evaluación del sistema y al manual de usuario del mismo.

La evaluación de la calidad de software debe tomar como punto inicial los procedimientos, estándares y metodologías existentes, y tiene como objetivo principal proveer la información cuantitativa en relación a la calidad, de manera que se establezca un mejor control de la misma. En suma, el proceso de evaluación es el vehículo a través del cual es posible mejorar la calidad de los sistemas. De acuerdo a esto, es necesario definir la calidad en función de varios elementos que denominaremos factores y que denotan facetas diferentes en la evaluación de resultados.

De acuerdo con los trabajos de William Perry existe una gama muy amplia de factores de calidad de sistemas. Sin embargo su instrumentación práctica no es inmediata, por lo que ha sido necesario seleccionar los factores más importantes para ser considerados en la evaluación. Para este sistema los factores que se consideraron fueron:

- Funcionalidad
- Eficiencia
- Facilidad de mantenimiento

Entenderemos por funcionalidad como la capacidad con la que un programa/aplicación ejecuta correctamente sus funciones desde un punto de vista técnico, mismas que fueron planteadas por el usuario e interpretadas por sus diseñadores.

Entenderemos por eficiencia como la manera con la cual un programa/aplicación ejecuta sus funciones sin desperdiciar recursos de máquina como memoria o espacio en disco y con un adecuado comportamiento.

Entenderemos por facilidad de mantenimiento como la simplicidad con la cual un programa/aplicación (en operación) puede ser modificado, probado y puesto de nuevo en operación.

Criterios

Asociados a los factores de calidad de sistemas definidos, tenemos para cada uno "elementos" de menor jerarquía, que denotan las características prácticas de calidad y por lo tanto resultan más fáciles de medir, mediante el establecimiento de métricas concretas para cada uno de los criterios identificados.

En la figura III.5.2.1 se muestra la relación entre los factores y criterios de calidad.

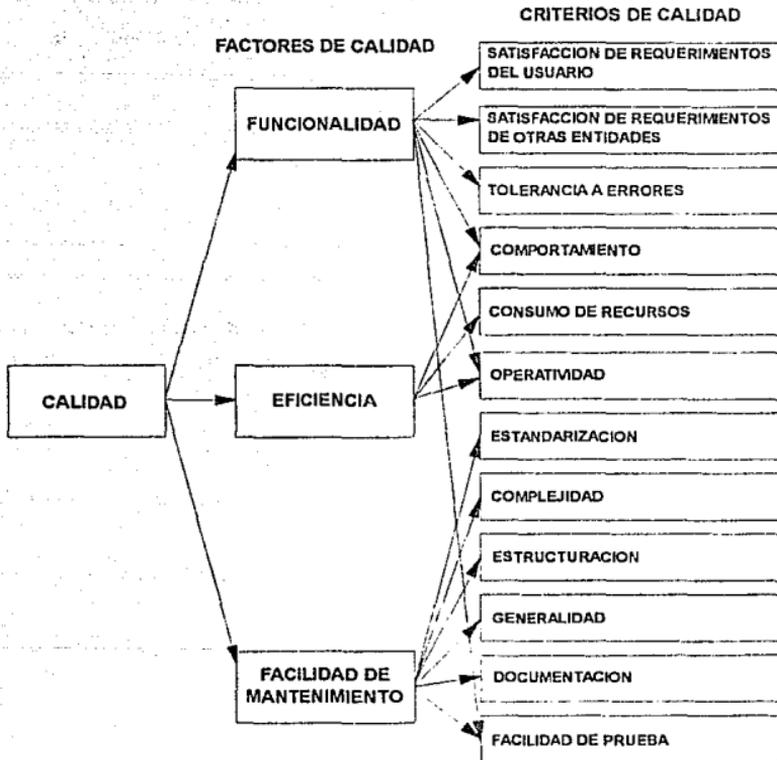


Figura III.5.2.1 Relación entre factores y criterios

Aunque algunos criterios son compartidos por más de un factor (como se aprecia en la figura III.5.2.1), en la práctica para propósitos de evaluación, cada criterio depende solamente de un factor, respetando la relación más frecuentemente identificada.

Criterios de calidad de evaluación de funcionalidad

1. Satisfacción de requerimientos del usuario.

Son los atributos del software que proporciona una instrumentación completa de las funciones requeridas por el usuario desde un punto de vista técnico.

2. Satisfacción de requerimientos de otras entidades.

Son los atributos del software que proporciona una instrumentación completa de las funciones requeridas por otras entidades.

3. Tolerancia a errores.

Son los atributos del software que proporcionan continuidad de operación bajo condiciones adversas (anormales).

Criterios de evaluación de calidad de eficiencia

1. Consumo de recursos.

Atributos del software relativos a sus requerimientos de procesador, entradas/salidas, memoria, disco e impresión al ser ejecutado en el equipo en el que va a utilizarse.

2. Comportamiento

Son los atributos del software relativos a sus características de respuesta para los volúmenes de procesos esperados.

3. Operatividad.

Son los atributos del software que determinan la facilidad o dificultad de operación del sistema.

Criterios de evaluación de calidad para facilidad de mantenimiento

1. Estandarización.

Son los atributos del software que proporcionan técnicas de análisis, diseño, construcción y documentación uniformes.

2. Complejidad.

Son los atributos que determinan la facilidad o dificultad de entender el sistema programa o módulo en lo que a su funcionamiento se refiere.

3. Estructuración.

Son los atributos que determinan su modalidad, facilidad de expansión y ejecución ordenada de funciones.

4. Documentación.

Es el complemento textual del software que proporciona información para su operación

5. Generalidad.

Atributos del software que determinan el nivel de utilización de técnicas de programación, funciones o bases de datos para uso compartido.

6. Facilidad de prueba.

Son los atributos del software que determinan la facilidad o dificultad de ejecución del sistema en un medio ambiente de prueba con la finalidad de verificar su funcionamiento.

Evaluación de la calidad

Para llevar a cabo la evaluación de la calidad de software que determinan la facilidad o dificultad de ejecución del sistema en un medio ambiente de prueba con la finalidad de verificar su funcionamiento.

- Se asignan "pesos" a cada uno de los criterios de evaluación.

Conforme al sistema que deba ser evaluado, este valor puede ser modificado si es necesario.

- Se asigna una calificación acorde al nivel de satisfacción del criterio de evaluación.

- Se multiplica esta calificación por el peso asignado al criterio de calidad correspondiente.

Los "pesos" asignados para cada uno de los criterios de evaluación de nuestro sistema fueron los siguientes:

Funcionalidad.

- Satisfacción de los requerimientos del usuario	9
- Satisfacción de los requerimientos de otras entidades	10
- Tolerancia a errores:	6
- Recuperación de errores de usuario	9
- Recuperación de errores de hardware	10
- Recuperación de errores de software	9
- Recuperación de errores de operación	9
Promedio de funcionalidad:	8.86

Eficiencia.

- Comportamiento:	8
- Consumo de recursos:	8
- Operatividad	9
Total de eficiencia:	8.33

Facilidad de mantenimiento.

- Estandarización	9
- Complejidad	4

- Estructuración	9
- Generalidad	9
- Documentación	7
- Facilidad de prueba	8
- Total de facilidad de mantenimiento:	7.66
PROMEDIO TOTAL:	<u>8.28</u>

Calificaciones:

10.0 Excelente

8.0 Bueno

6.0 Satisfactorio

4.0 Inaceptable

2.0 No se aplica

De acuerdo a la evaluación anterior se puede constatar que el Sistema Integral de Abastecimientos en términos generales es BUENO, considerando que satisface las necesidades para el que fue concebido.

CONCLUSIONES

CONCLUSIONES

Para nosotros, la realización del presente trabajo significó la oportunidad de llevar a cabo a la práctica, en un caso real, los conocimientos adquiridos durante nuestra formación profesional, refiriéndonos específicamente a las Bases de Datos conjuntamente con el desarrollo de sistemas de información.

En lo referente a las bases de datos, nos encontramos que de acuerdo al estudio que hicimos para seleccionar el tipo de base de datos a utilizar, que la tendencia en el mercado eran las Bases de Datos Relacionales, las cuales ofrecen muchas facilidades para su diseño e implementación. Gracias a sus características fueron las que se eligieron como la base de datos para llevar a cabo nuestro propósito. Con esto, también pudimos reforzar y ampliar nuestros conocimientos en cuanto a la forma de diseñar, manejar e implementar una base de datos, así, como saber todo en cuanto a la definición, manipulación, y las reglas de seguridad e integridad de los datos que en estas se almacenan.

Al tratar con bases de datos relacionales necesariamente tuvimos que trabajar con Manejadores de Bases de Datos Relacionales (RDBMS). Esto nos ayudo bastante, ya que nos pudimos actualizar en cuanto a los diferentes DBMS que están disponibles en el mercado, específicamente con los que tienen la capacidad de manejar grandes volúmenes de información. De esta forma, pudimos constatar que los RDBMS nos

ofrecen un conjunto poderoso de herramientas y lenguajes de cuarta generación (4GL), entre otras facilidades para una manipulación eficiente de los datos.

Con lo que respecta al RDBMS que seleccionamos: Sybase, podemos decir que aunque no es la más completa, si es de los que más y mejores alternativas ofrece para facilitar la implementación de cualquier sistema. Con la ayuda de sus diferentes herramientas y su lenguaje de cuarta generación (Transact-SQL), pudimos llevar a cabo la implementación de los diferentes módulos del sistema de abastecimientos con el menor esfuerzo posible; sin dejar de mencionar que su 4GL nos ofreció un enfoque completamente distinto en cuanto a la forma de programar se refiere, por lo que en un principio nos costo un poco de tiempo y trabajo acostumbrarnos a él.

Trabajando con este tipo de plataformas también tuvimos la necesidad de conocer la parte complementaria por el lado del desarrollo de sistemas: la Metodología Entidad-Relación (ER), la cual es básicamente una técnica de diagramas en donde se presentan las diferentes entidades que están involucradas en un sistema con sus respectivas relaciones. Pudimos visualizar también que fue relativamente simple y fácil de comprender, además que constituyó una herramienta de comunicación entre los analistas y los programadores del sistema y los usuarios finales, ya que sólo se basa en dos fases principales: la definición del esquema empresarial y la traducción de dicho esquema al esquema de usuario.

Por otra parte, el trabajar en un ambiente Multiusuario bajo el sistema operativo UNIX nos permitió conocer el potencial de las minicomputadoras y de los sistemas operativos multiusuario que en conjunto permiten a los sistemas manejadores de bases de datos atender a un gran número de usuarios concurrentemente y por consiguiente manejar grandes volúmenes de información. Esto también nos permitió establecer una

diferencia entre lo que es un sistema basado en computadoras personales y un sistema de misión crítica corriendo en un sistema multiusuario.

El desarrollo de lo que fue el sistema en sí, nos permitió profundizar sobre las diferentes etapas implicadas en el proceso de desarrollo del sistema, ya que de una u otra manera vivimos como nació el sistema, desde su planeación y análisis pasando por su diseño hasta su implementación. El convivir directamente con los diferentes usuarios y con nuestros compañeros colaboradores fue una experiencia más que debemos considerar para nuestros futuros proyectos.

Por último, diremos que el desarrollo del Sistema de Abastecimientos contribuirá a mejorar significativamente las funciones de la Subdirección de Abastecimientos y las diferentes Oficinas Regionales del Grupo Peñoles, lo cual nos llena de orgullo y nos sirve de aliciente para seguir adelante en nuestra vida profesional.

BIBLIOGRAFIA

BIBLIOGRAFIA

Bloor, Robin and Howard, Philip **DATABASE AN EVALUATION AND COMPARISON.**
Edit Tom Jowit. United Kindom. 1992

Cárdenas, Alfonso F. **DATA BASE MANAGEMENT SYSTEMS.** 2nd. edition. Boston.
Allyn and Bacon. 1979

Date, C.J. **INTRODUCCON A LOS SISTEMAS DE BASES DE DATOS. VOL I.** 5a
edición. México. Addison-Wesley Iberoamericana. 1993

Date, C.J. **AN INTRODUCTION TO DATABASE SYSTEMS. VOL I.** 5th edition.
Addison-Wesley Systems Programming Series 1990.
Addison-Wesley publishing Company, Inc.

Davis, Gordon B. **COMPUTER DATA PROCESSING.** 2th edition. International Student
Edition. 1976.

DBMS, DBMS BUYERS GUIDE ISSUE 1993. Septiembre 1993. (revista). Pag 10-88

Finkestein, Rich. **MULTIUSER DATABASES. BYTE.** Product Focus. Mayo 1990
(revista). Pag 136-148.

James A. Senn. ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION. McGraw-Hill
2a. edición. 1993.

James R. Groff y PAul N. Weinberg. APLIQUE SQL . Mc Graw-Hill. 2a. edición. 1992.

Korth, Henry and F. Silberschatz, Abraham. DATABASE SYSTEM CONCEPTS.
McGraw-Hill Advanced Computer Science Series. 1986.

MANUAL DE SYBASE APT WORKBENCH

RELASE 4.0

REVISED MAY., 1989

APLICACION TOOLS

RELASE 4.0

REVISED MAY., 1989

MANUAL DE SYBASE SQR

MANUAL DE SYBASE TRANSACT SQL

Pressman. Roger S. SOFTWARE ENGINEERING: A PRACTITIONERS APPROACH.
3a edición. México. McGraw-Hill. 1984.

GLOSARIO

GLOSARIO

3GL

Se refiere a los lenguajes de computadora de tercera generación, tales como: Cobol, Pascal, Fortran entre otros.

4GL

Lenguaje de desarrollo de instrucciones sencillas que incrementan la productividad.

Actualización distribuida

Una actualización se puede realizar a través de un nodo o nodos remotos, se requiere del two-phase-commit para garantizar la integridad de la transacción

Aplicación

Pieza de software de uso sencillo que realiza una serie de tareas para el usuario final.

Apuntador

Parte de información la cual contiene una dirección física (en memoria o disco).

Arbol binario

Estructura de árbol caracterizada por tener por lo menos un nodo y dos ramas.

Archivo espejo

Copia del archivo maestro, si este llega a fallar el archivo duplicado puede utilizarse para continuar la operación o recuperar la información.

Asíncrono

Proceso o acción la cual puede ocurrir sin necesidad de esperar que otro proceso o acción sea completado.

Atributo

Conjunto de propiedades o características de una entidad. Por ejemplo la edad de un empleado.

Backup

Utilería usada para tener una copia de la base de datos o parte de esta usualmente se utiliza la cinta, esta copia puede ser usada para restaurar la base de datos en caso de una falla seria .

Base de datos distribuida

Es una base de datos lógica la cual puede ser utilizada a través de un sistema de cómputo desde varias localidades conectadas a través de una red.

Base de datos relacional

Colección de tablas de dos dimensiones de una base de datos que tienen relación entre si.

Batch

Proceso no interactivo el cual puede correr cuando el sistema es cargado.

Block

Cantidad de datos que pueden ser movidos o direccionados como una sola unidad.

Bloqueo de actualización

Proceso en el cual los usuarios solo pueden hacer acceso de lectura cuando alguien más está haciendo uso del archivo y se esta realizando alguna actualización

Buffer

Unidad de memoria utilizada para recibir o enviar bloques de registros a otro dispositivo. Un buffer en términos de base de datos es una unidad de I/O.

Búsqueda distribuida (Query)

Búsqueda que puede correr a través de una base de datos distribuida involucrando uno o varios nodos de la red.

Campo

Conjunto de caracteres que representa un tipo de información.

Campo Indexado

Columnas de una tabla usadas para construir estructuras de índices.

CASE

Computer Aided Software Engineering. Método para el análisis y diseño de sistemas el cual provee una representación gráfica y textual de los requerimientos de un sistema, de su estructura y funcionalidad que permiten reducir el tiempo en el ciclo de desarrollo de un sistema.

Commit

Proceso que completa una transacción lógica y deja disponible los resultados de la transacción a otros usuarios de la base de datos.

Compilador

Parte de software que acepta un lenguaje de programación y genera un código de máquina.

Concurrente

Se refiere a cuando los usuarios o procesos pueden acceder al mismo tiempo un recurso.

Cursor

En SQL es el apuntador de una colección de renglones que son regresados después de un query.

Daemon (Demon)

Un proceso del sistema UNIX el cual puede ejecutarse sin la petición del usuario.

DBMS (Data Base Management System)

El Sistema Manejador de Base de Datos es un conjunto de programas que tienen la función principal de administrar una base de datos y son la interface entre el sistema y los usuarios.

Deadlock

Estado en el cual dos o más usuarios están en espera de un recurso ocupado por otro u otros usuarios.

Diccionario

Usualmente almacena los meta-datos (los datos acerca de los datos) y cataloga la información, puede ser usada para el almacenamiento de formas, programas, entre otros.

Entidad

Una persona, lugar o concepto acerca del cual una organización desea almacenar datos.

Escalabilidad

Propiedad de una base de datos para que un gran número de procesos pueda ser utilizada eficientemente.

Estabilidad del cursor

Garantiza que ningún otro usuario pueda acceder los datos donde el cursor está posicionado.

Estructura de liga

Estructura de datos que utiliza apuntadores para conectar registros en una secuencia lógica.

Indice

Estructura de datos adicional que permite realizar búsquedas por renglones de manera más eficiente, cuando un campo índice es usado como parte de un query.

Integridad

Se refiere a la precisión y validación de los datos dentro de una base de datos. Existen mecanismos de control para la integridad como triggers y reglas.

Interoperable

Capacidad de un producto de tener interfase con muchos otros productos.

Intérprete

Traductor de instrucciones.

Llave

Atributo de una tabla cuyo valor es requerido para realizar búsquedas en otra tabla.

Llave primaria

Atributo o dato único que identifica el renglón de una tabla.

Local

Sistema en el cual los usuarios están conectados directamente.

Log

Registro de todos los cambios que se han efectuado dentro de una base de datos.

Maestro/Esclavo

Cuando un proceso o sistema está designado a controlar a otro proceso o sistema, si ocurre una falla en el maestro el esclavo no puede continuar.

Manejo de memoria

Manejador de los recursos de memoria que garantiza la mayor capacidad de uso.

Meta-data

Información o datos acerca de los datos.

MIS

Management Information Systems: sistemas que pueden manejar los datos y generar búsquedas y reportes.

Modelo de Dato

Conjunto de entidades reglas y relaciones que pueden ser usadas para el modelado de un sistema.

Modelo Entidad/Relación

Relación entre entidades creado por Peter Chen. Un modelo de datos puede incluir información acerca de las relaciones entre las entidades (u objetos).

Nodo

Es un índice de página dentro de la estructura de un árbol binario.

Nodo de la red

Sistema de cómputo que es parte de una red.

Nodo raíz

Dentro de una estructura de árbol es el índice de mayor nivel.

Normalización

Proceso para el alojamiento correcto de los atributos de una entidad.

Objeto

Combinación de estados o conjunto de métodos.

OLTP (On Line Transaction Processing)

Procesamiento de transacciones en línea

Optimizar

Administración de los componentes de un sistema de base de datos que determinan el mejor camino para satisfacer las necesidades.

Partición

División de memoria en áreas lógicas.

Precompilador (SQL)

Pieza de software que convierte las instrucciones SQL en un lenguaje de manipulación nativo que el compilador pueda reconocer.

Procedimiento Almacenado

Es una colección de sentencias SQL almacenadas en la base de datos, las cuales pueden ser ejecutadas por nombre.

Los procedimientos almacenados pueden aceptar y retornar parámetros, retornar valores y llamar a otros procedimientos. Además de que pueden ejecutarse más rápido que los mismos comandos ejecutados interactivamente o desde un batch.

Proceso

Unidad de programa que es ejecutado por el CPU, un programa puede ejecutar más de un proceso.

Query

Búsqueda y selección de un conjunto de datos usualmente efectuada en una o más tablas de la base de datos.

Rdbms (Relational Data Base Management System)

Son sistemas manejadores de bases de datos con la capacidad de manejar bases de datos relacionales.

Red

Sistemas de cómputo conectados (nodos) por canales de comunicación y el software apropiado.

Registro

Conjunto de campos, un conjunto de registros hacen un archivo, su equivalente en una tabla son los renglones.

Relación muchos a muchos

Relación entre dos entidades la cual puede tener relación con muchas ocurrencias con otro tipo de entidad.

Relación uno a muchos

Relación entre dos entidades en la cual la ocurrencia de la primera entidad está relacionada con muchas ocurrencias del segundo tipo de entidad.

Relación uno a uno

Relación entre dos entidades en la cual la primera ocurrencia está relacionada sólo con otra ocurrencia del segundo tipo de entidad.

Remoto

Sistema que puede ser accesado a través de una red.

Replicación

Copia de los datos usados en un sistema distribuido para proveer mayor velocidad local en los datos usados con mayor frecuencia.

Report Writer

Pieza de software utilizada para generar reportes de manera rápida y eficiente.

Run-Time

Cuando un programa se ejecuta.

Select (SQL)

Comando para extraer los datos seleccionados de una tabla en base a un criterio de búsqueda.

Servidor

Proceso que provee un servicio usualmente a la base de datos.

Síncrono

Proceso o acción que debe esperar que otro proceso se termine antes que este pueda ejecutarse.

SQL

Structure Query Language: Estándar no procedural para la manipulación de un lenguaje en sistemas de bases de datos relacionales.

Tabla

Representación de los datos, donde éstos son acomodados en un arreglo de dos dimensiones (renglones y columnas).

Tablespace

Es usado para describir áreas físicas para el almacenamiento de tablas.

Timeout

Período de tiempo especificado en la cual una acción particular puede ocurrir.

Tipo de Dato

Tipos de datos específicos soportados por un sistema; por ejemplo: texto, numérico, fecha, etc.

Tipo de registro

Es un conjunto de campos predeterminados dentro de un archivo.

Transacción

Es una serie de sentencias SQL las cuales son tratadas como un sólo evento.

Tree

Estructura de árbol consistente en nodos los cuales están conectados por ramas, el nodo raíz se encuentra en el nivel más alto y de el se derivan otros nodos.

Trigger

Es un tipo especial de procedimiento almacenado, el cual especifica una acción la cual debe ser tomada cuando una condición particular ocurra, específicamente cuando los datos en una tabla son insertados, actualizados o borrados.

Los triggers permiten proporcionar una consistencia más estrecha en la base de datos y por lo consiguiente en cada aplicación.

Vista

Tabla virtual la cual no es almacenada en forma física en la base de datos.

APENDICE A.

DICCIONARIO DE DATOS

NOTA: En este apéndice sólo se incluye una parte de la información referente al mismo, debido a lo extenso del sistema.

Tabla : agente_aduanal.abct

APENDICE A

```
create table agente_aduanal /*Datos generales del agente aduanal */
( agt_clave char(02) not null, /*Clave del Agente Aduanal PKEY */
  agt_nombre char(40) not null, /*Nombre del Agente Aduanal */
  agt_pais char(04) not null, /*País */
  agt_estado char(04) not null, /*Estado */
  agt_cod_posta int not null, /*Codigo Postal */
  agt_colonia char(40) not null, /*Colonia */
  agt_calle char(40) not null, /*Calle */
  agt_fax char(14) null, /*Número de fax */
  agt_telefono char(14) not null, /*Teléfono */
  agt_cve_repre char(02) not null, /*Representante Extranjero */
  agt_atencion char(40) null) /*Representante Extranjero */
```

```
go
create unique clustered index xp_age_aduan on agente_aduanal
( agt_clave )
go
exec sp_primarykey agente_aduanal, agt_clave
go
```

Tabla : atencion_proveedor.abct

```
create table atencion_proveedor /*Domicilio del prov. y sucur. */
( prv_rfc char(20) not null, /*R.F.C. del proveedor FKEY P */
  atn_num_conse tinyint not null, /*Número Consecutivo de la dir.*/
  atn_atencion char(40) null, /*Persona que atiende al comp.*/
  atn_puesto char(40) null) /*Puesto de la persona que at. */
go
create unique nonclustered index xp_atencion_proveedor on atencion_proveedor
( prv_rfc , atn_num_conse )
go
exec sp_primarykey atencion_proveedor ,
prv_rfc , atn_num_conse
go
exec sp_foreignkey atencion_proveedor , proveedor ,
prv_rfc
go
```

Tabla : banco_proveedor.abct

```
create table banco_proveedor /*Banco del prov. y sucur. */
( prv_rfc char(20) not null, /*R.F.C. del proveedor FKEY P */
  bco_num_conse tinyint not null, /*Número Consecutivo del Banco.*/
  bco_cuenta char(30) null, /*Cuenta Bancaria */
  bco_nombre char(40) null, /*Nombre del Banco */
  bco_pais char(4) null, /*País */
  bco_estado char(4) null, /*Estado */
  bco_colonia char(40) null, /*Colonia */
  bco_calle char(40) null, /*Calle */
  bco_dig_munic char(40) null, /*Delegación */
  bco_cod_posta char(8) null) /*Codigo Postal */
go
create unique nonclustered index xp_banco_proveedor on banco_proveedor
( prv_rfc , bco_num_conse )
go
exec sp_primarykey banco_proveedor ,
prv_rfc , bco_num_conse
go
exec sp_foreignkey banco_proveedor , proveedor ,
prv_rfc
go
```

```

create table bulto
(agt_clave char(2) not null, /*Datos generales del bulto */
tra_clave char(14) not null, /*Clave del agente aduanal PKEY FKEY */
/*mov_num_conse int not null, /*Numero de tr fico PKEY FKEY */
/*mov_tipo char(2) not null, Consecutivo del movimiento PKEY FKEY */
bul_num_conse char(2) not null, Tipo de movimiento PKEY FKEY */
bul_num_conse smallint not null, /*Numero consecutivo del bulto PKEY */
bul_cantidad float not null, /*Cantidad recibida */
bul_descripcion char(40) null, /*Descripcion de la bulto */
bul_tipo char(2) not null, /*Tipo de bulto */
bul_peso float not null, /*Peso del bulto */
ume_clave char(4) not null) /*Unidad de medida */
go

```

Tabla : calendario.abct

```

create table calendario
( cal_ano_mes char(06) not null, /* Año Mes al que se refiere */
cal_dia char(02) not null) /* Dia que no se labora */
go

```

Tabla : cotizacion.abct

```

create table cotizacion
(prv_rfc char(20) not null,
cot_clave char(12) not null,
cot_fec_cotiz fecha_hoy not null,
cot_fec_limit_vigen datetime not null,
mon_clave char(2) not null,
prv_cnd_pago char(3) null,
prv_dia_condi_pago tinyint null,
prv_dto_gener float null,
prv_tpo_entra smallint not null,
prv_lug_entra char(2) not null,
prv_cnd_entra char(3) null,
prv_dia_pront_pago tinyint null,
prv_dto_pront_pago float null,
prv_rfc_repra char(20) not null,
pdo_num_conse tinyint not null,
cot_des_lug char(35) null)
go
exec sp_foreignkey cotizacion , proveedor ,
prv_rfc
go
exec sp_foreignkey cotizacion , moneda ,
mon_clave
go

```

```

create table descripcion
( des_clave      int          not null, /* Clave de descrip. cons. PKEY */
  des_num_conse  tinyint     not null, /* consecut. de descrip PKEY */
  des_nombre     varchar(255) not null) /* Descripcion de matrl */
go
create unique nonclustered index xp_descripcion on descripcion
( des_clave ,des_num_conse )
go
exec sp_primarykey descripcion,
     des_clave , des_num_conse
go
drop trigger abindme01_idu
go
/* TRIGGER      INS/DEL/UPD DESCRIPCION */
create trigger abindme01_idu on descripcion for insert, delete, update
as
begin
  if @@rowcount = 0
    return
  exec abvaluu_pro /* verifica actualizacion desde aplicacion */
  return
end
go

```

Tabla : descripcion_ingles.abct

```

create table descripcion_ingles
( des_clave      int          not null, /* Clave de descrip. cons. PKEY */
  des_num_conse  tinyint     not null, /* consecut. de descrip PKEY */
  des_nombre     varchar(255) not null) /* Descripcion de matrl */
go
create unique nonclustered index xp_descripcion on descripcion_ingles
( des_clave ,des_num_conse )
go
exec sp_primarykey descripcion_ingles,
     des_clave , des_num_conse
go
drop trigger abindmi01_idu
go
/* TRIGGER      INS/DEL/UPD DESCRIPCION_INGLES */
create trigger abindmi01_idu on descripcion_ingles for insert, delete, update
as
begin
  if @@rowcount = 0
    return
  exec abvaluu_pro /* verifica actualizacion desde aplicacion */
  return
end
go

```

Tabla : desc_generica_transf.abct

```

create table desc_generica transf
( alm_clave     char(4)      null, /* Clave almacen FKEY PKEY */
  req_num_conse int          null, /* Consec. requis FKEY PKEY */
  req_consec    int          null, /* Consec. sisinf */
  req_descrip   char(40)     null) /* Descrip. sisinf */
go

```

Tabla : desc_partida_requi_transf

APENDICE A

```
create table desc_partida_requi_transf
( alm_clave char(4) null, /* Clave almacen FKEY PKEY */
  req_num_conse int null, /* Consec. requis FKEY PKEY */
  prq_num_conse smallint null, /* Partida requis PKEY */
  prq_consec int null, /* Consec. sisinf */
  prq_descrip char(36) null) /* Descrip. sisinf */
go
```

Tabla : descr_temporal.abct

```
create table descr_temporal
( des_clave int not null, /* Clave de descrip cons PKEY */
  des_num_conse tinyint not null, /* consecut. de descrip PKEY */
  des_nombre varchar(255) not null) /* Descripcion del material */
go
create unique nonclustered index xp_des_tem on descr_temporal
( des_clave , des_num_conse )
go
exec sp_primarykey descr_temporal ,
  des_clave , des_num_conse
go
drop trigger abindte01_idu
go
/* TRIGGER INSERT/DEL/UPD DESCR_TEMPORAL */
create trigger abindte01_idu on descr_temporal_ingles for insert, delete,
update
as
begin
  if @@rowcount = 0
    return
  exec abvalusu_pro /* verifica actualizacion desde aplicacion */
  return
end
go
```

Tabla : descr_temporal_modifica.a

```
create table descr_temporal_modifica /* descripcion de partidas */
( des_clave int not null, /* Clave de descrip cons PKEY */
  des_num_conse tinyint not null, /* consecut. de descrip PKEY */
  des_nombre varchar(255) not null) /* Descripcion del material */
go
create unique nonclustered index xp_descr_temporal_modifica
on descr_temporal_modifica
( des_clave , des_num_conse )
go
exec sp_primarykey descr_temporal_modifica ,
  des_clave , des_num_conse
go
```

Tabla : descr_temporal_ingles.abc

APENDICE A

```
create table descr_temporal_ingles
( des_clave int not null, /* Clave de descrip cons PKEY */
  des_num_conse tinyint not null, /* consecut. de descrip PKEY */
  des_nombre varchar(255) not null) /* Descripcion del material */
go
create unique nonclustered index xp_des_tem_ing on descr_temporal_ingles
( des_clave , des_num_conse )
go
exec sp_primarykey descr_temporal_ingles ,
      des_clave , des_num_conse
go

drop trigger abindti01_idu
go

/* TRIGGER INSERT/DEL/UPD DESCR_TEMPORAL */
create trigger abindti01_idu on descr_temporal_ingles for insert, delete,
update
as
begin
  if @@rowcount = 0
    return

  exec abvaluu_pro /* verifica actualizacion desde aplicacion */

  return
end
go
```

Tabla : descr_temporal_transf.abc

```
create table descr_temporal_transf
( des_clave int null, /* Clave de descrip cons PKEY */
  des_num_conse tinyint null, /* consecut. de descrip PKEY */
  des_nombre varchar(255) null) /* Descripcion del material */
go
```

Tabla : gastos_reales.abct

```
create table gastos_reales
( agt_clave char(2) not null, /* agente aduanal */
  tra_clave char(14) not null, /* numero de trafico */
  gto_clave char(4) not null, /* clave del gto. aduanal */
  gto_fec_cuota datetime not null, /* fecha de actualizacion */
  gto_importe money not null, /* monto del gasto */
  gto_iva_importe money null, /* I.V.A. del importe */
  mon_clave char(2) not null) /* tipo de moneda */
go
```

```

create table domicilio_proveedor
(prv_rfc          Char(20)      not null, /*Domicilio del prov. y sucur. */
pdo_num_conse   tinyint      not null, /*R.F.C. del proveedor FKEY P */
pdo_atencion    Char(40)      null, /*Numero Consecutivo de la dir.*/
pdo_pais        Char(4)       not null, /*Persona que atiende al comp.*/
pdo_estado     Char(4)       not null, /*Pais */
pdo_colonia     Char(40)      not null, /*Estado */
pdo_calle       Char(40)      not null, /*Colonias */
pdo_dig_munic   Char(40)      not null, /*Calle */
pdo_cod_posta   Char(8)       not null, /*Delegacion */
pdo_telefono    Char(14)      null, /*Codigo Postal */
pdo_fax         Char(14)      null, /*Telefono */
pdo_banco       Char(4)       null, /*Fax */
pdo_cta_banca   Char(20)      null) /*Banco */
/*Cuenta del banco */

go
create unique nonclustered index xp_domicilio_proveedor on domicilio_proveedor
( prv_rfc , pdo_num_conse )
go
exec sp_primarykey domicilio_proveedor ,
      prv_rfc , pdo_num_conse
go
exec sp_foreignkey domicilio_proveedor , proveedor ,
      prv_rfc
go

```

Tabla : domicilio.abct

```

create table domicilio
( dom_clave      tinyint      not null, /* Clave domic. Consec. PKEY */
  dom_nombre     varchar(255) not null, /* Domicilio 3 x 36 */
  dom_tel        telefono     null, /* Telefono de domicilio */
  dom_fax        telefono     null) /* Fax de domicilio */

go
create unique clustered index xp_domicilio on domicilio
( dom_clave )
go
exec sp_primarykey domicilio ,
      dom_clave
go

```

Tabla : ofnareg_at_compania.abct

```

create table ofnareg_at_compania /* tabla asociativa oficina atiende cias */
( ofi_clave     Char(4)       not null, /* Clave de oficina PKEY FKEY */
  cia_clave     smallint      not null) /* Clave de compania PKEY FKEY */

go
create unique clustered index xp_ofnareg_at_compania on ofnareg_at_compania
( ofi_clave , cia_clave )
go
exec sp_primarykey ofnareg_at_compania ,
      ofi_clave , cia_clave
go
exec sp_foreignkey ofnareg_at_compania , ofna_regional,
      ofi_clave
go
exec sp_foreignkey ofnareg_at_compania , compania ,
      ofi_clave
go

```

```

create table embarque
(alm_clave      char(4)      not null, /* Cve. de almacen      */
 ped_clave      int          not null, /* Num. consec. reuis.  */
 ped_num_conse  tinyint     not null, /* Num. consec. Pedido PKEY */
 ppe_num_conse  smallint    not null, /* Num. consec. partida  */
 emb_num_conse  smallint    not null, /* Num. consec. de embarque */
 emb_cantidad   float       null, /* Cantidad requerida    */
 emb_pre_unita  money       null, /* Precio unitario      */
 emb_cve_statu char(2)     null, /* Status de la partida  */
 emb_fec_statu  datetime    null, /* Fecha de status      */
 emb_fec_entre  datetime    null) /* Fecha de entrega de mat. */

go
create unique nonclustered index xp_embarque on embarque
( alm_clave, ped_clave, ped_num_conse, ppe_num_conse, emb_num_conse)
go
exec sp_primarykey embarque,
alm_clave, ped_clave, ped_num_conse, ppe_num_conse, emb_num_conse
go
exec sp_foreignkey embarque , partida_pedido,
alm_clave, ped_clave, ped_num_conse, ppe_num_conse
go

/* asociacion de RULES */

sp_bindrule 'gtr_0' , 'embarque.emb_num_conse'
go
sp_bindrule 'gte_0' , 'embarque.emb_cantidad'
go

```

Tabla : embarque_modifica.abct

```

create table embarque_modifica
(alm_clave      char(4)      not null, /* Cve. de almacen      */
 ped_clave      int          not null, /* Num. consec. reuis.  */
 ped_num_conse  tinyint     not null, /* Num. consec. Pedido PKEY */
 ppe_num_conse  smallint    not null, /* Num. consec. partida  */
 emb_num_conse  smallint    not null, /* Num. consec. de embarque */
 emb_cantidad   float       not null, /* Cantidad requerida    */
 emb_pre_unita  money       null, /* Precio Unitario      */
 emb_cve_statu  char(2)     null, /* Status de la partida  */
 emb_fec_statu  datetime    null, /* Fecha de estatus     */
 emb_fec_entre  datetime    null) /* Fecha de entrega de mat. */

go
create unique nonclustered index xp_embarque_modifica
on embarque_modifica
( alm_clave, ped_clave, ped_num_conse, ppe_num_conse, emb_num_conse)
go
exec sp_primarykey embarque_modifica,
alm_clave, ped_clave, ped_num_conse, ppe_num_conse, emb_num_conse
go

```

Tabla : detalle_resumen.abct

```

create table detalle_resumen
(com_clave      char(2)      not null, /* Cve. del comprador   */
 alm_clave      char(4)     not null, /* Cve de almacen PKEY  */
 res_periodo    int         not null, /* Periodo de proceso   PKEY  */
 res_semana     tinyint    not null, /* Semana de proceso    */
 res_cantida    int         not null, /* Cantidad en proceso  */

go
create unique nonclustered index xp_detalle_resumen on detalle_resumen
( com_clave, alm_clave, res_periOdo, res_semana )
go
/* llaves involucradas */
exec sp_primarykey detalle_resumen ,
com_clave, alm_clave, res_periOdo, res_semana
go

```

```

create table factura_pedido /*Tabla que relaciona proveedor con sus facturas*/
(agt_clave char(2) not null, /*Clave del agente aduanal FKEY */
tra_clave char(14) not null, /*Número de tráfico FKEY */
alm_clave char(4) not null, /*Cve de almacen FKEY */
ped_clave int not null, /*Numero consec. requis. FKEY */
ped_num_conse tinyint not null, /*Num. consec (anexo) FKEY */
pfa_num_factu char(14) not null, /*Numero de la factura PKEY */
pfa_importe money not null, /*Importe de la factura */
mon_clave char(2) not null, /*Clave de la Moneda */
pfa_fec_factu datetime not null) /*Fecha de la factura */
go

```

Tabla : factura trafico.abct

```

create table factura trafico /*Tabla que relaciona Trafico con sus facturas*/
(agt_clave char(2) not null, /*Clave del agente aduanal FKEY */
tra_clave char(14) not null, /*Número de tráfico FKEY */
ftr_num_factu char(14) not null, /*Numero de la factura PKEY */
ftr_imp_factu money not null, /*Importe de la factura */
ftr_fec_factu datetime not null, /*Fecha de la factura */
mon_clave char(2) not null) /*Clave de la Moneda */
go
create unique clustered index xp_fac_tra on factura trafico
( agt_clave, tra_clave, ftr_num_factu )
go
exec sp_primarykey factura trafico
agt_clave, tra_clave, ftr_num_factu
go

```

Tabla : factura trafico_pedido.ab

```

create table factura trafico_pedido
(agt_clave char(2) not null, /*Clave del agente aduanal FKEY */
tra_clave char(14) not null, /*Número de tráfico FKEY */
ftr_num_factu char(14) not null, /*Numero de la factura PKEY */
alm_clave char(4) not null, /*Cve de almacen FKEY */
ped_clave int not null, /*Numero consec. requis. FKEY */
ped_num_conse tinyint not null) /*Num. consec (anexo) FKEY */
go
create unique clustered index xp_fac_tra_ped on factura trafico_pedido
(agt_clave, tra_clave, ftr_num_factu, alm_clave, ped_clave, ped_num_conse )
go
exec sp_primarykey factura trafico_pedido
agt_clave, tra_clave, ftr_num_factu, alm_clave, ped_clave, ped_num_conse
go

```

Tabla : lista_precio.abct

```

create table lista_precio
(lis_convenio char(4) not null,
lin_clave tinyint not null,
lin_sub_linea char(6) not null,
mco_clave char(14) not null,
lis_des_gener char(40) not null,
lis_pre_unita money not null,
lis_pco_unita float null,
lis_descuento float null,
lis_tpo_entre smallint null,
lis_fec_alta fecha_hoy)

```

```

create table impresora
(imp_clave char(20) not null, /* Impresoras */
 tim_clave tinyint not null, /* Clave */
 imp_ubicacion char(40) not null, /* Ubicacion Fisica */
 imp_pap_usado char(60) not null, /* Tipo de Papel usado */
 imp_cable char(15) null, /* Numero de Cable al sistema */
 imp_sop_grafi char(01) not null) /* Si soporta graficos = S */
go
create unique nonclustered index xp_imp on impresora
(imp_clave)
go
exec sp_primarykey impresora, imp_clave
go

```

Tabla : modulo.abct

```

create table modulo
(mod_clave char(20) not null, /* modulo */
 mod_descripcion char(60) not null) /* Descripcion */
go
create unique nonclustered index xp_mod on modulo
(mod_clave)
go
exec sp_primarykey modulo, mod_clave
go

```

Tabla : modificacion.abct

```

create table modificacion
(alm_clave char(4) not null, /* Cve de almacen FREY PKEY */
 ped_clave int not null, /* Num consec.rsquis. PKEY */
 ped_num_conse tinyint not null, /* Num. consec (anexo) PKEY */
 ped_num_modif tinyint not null, /* Numero de modificacion */
 ped_fec_modif datetime null, /* Fecha ultima modificacion */
 ped_fec_autor_modif datetime null, /* Fecha de autorizacion de Modif */
 ped_imp_modif money null, /* Monto Modificado */
 mod_num_parti smallint null, /* Numero de Partidas */
 com_clave char(2) null) /* Calve del comprador */
go
create unique nonclustered index xp_modificacion on modificacion
(alm_clave, ped_clave, ped_num_conse, ped_num_modif )
go
/* llaves involucradas */
exec sp_primarykey modificacion ,
alm_clave, ped_clave, ped_num_conse, ped_num_modif
go

```

Tabla : movimiento_material.abct

```

create table movimiento_material /*Tabla que registra los movimientos de material*/
(agt_clave char(2) not null, /*Clave del agente aduanal PKEY FREY */
 tra_clave char(14) not null, /*Numero de tr fico PKEY FREY */
 mov_num_conse int not null, /*Consecutivo del movimiento PKEY */
 mov_tip char(2) not null, /*Tipo de movimiento PKEY */
 mov_fecha datetime not null, /*Fecha del movimiento de material */
 mov_cnd_flete char(4) null, /*Condiciones de flete */
 mov_remittente char(40) null, /*Nombre del remitente */
 mov_gui_talon char(14) not null, /*Guia o tal'n */
 mov_tip_trans char(20) null, /*Transporte utilizado */
 mov_fec_alta fecha_hoy not null) /*Fecha en que se da de alta el Mov. */
go

```

Tabla : movimiento_partida_pedido

APENDICE A

```

create table movimiento_partida_pedido /*Tabla asociativa de movimientos con partid
(agt_clave char(2) not null, /*Clave del agente aduanal PKEY */
tra_clave char(14) not null, /*Número de tráfico PKEY */
mov_num_conse int not null, /*Consecutivo del movimiento PKEY */
mov_tipo char(2) not null, /*Tipo de movimiento PKEY */
alm_clave char(4) not null, /*Clave del almacén PKEY */
ped_clave int not null, /*Clave del pedido PKEY */
psd_num_conse tinyint not null, /*Consecutivo del pedido PKEY */
ppe_num_conse smallint not null, /*Consecutivo partida pedido PKEY */
emb_num_conse smallint not null, /*Consecutivo partida embarque PKEY */
mov_cantidad float not null) /*Cantidad en movimiento */
go

```

Tabla : modalidad_status.abct

```

create table modalidad_status /*Modalidades de los estatus tráfico*/
(agt_clave char(2) not null, /*Clave del agente aduanal PKEY */
tra_clave char(14) not null, /*Número de tráfico PKEY */
str_cve_statu char(02) not null, /*Status del tráfico PKEY */
mos_cve_statu char(02) not null, /*Modalidad PKEY */
mos_nivel tinyint not null, /*Nivel Gerarquico de la modalidad */
mos_fec_statu datetime not null) /*Fecha de la modalidad */
go

```

```

create unique nonclustered index xp_mod_statu on modalidad_status
(agt_clave, tra_clave, str_cve_statu, mos_cve_statu)
go
exec sp_primarykey modalidad_status, agt_clave, tra_clave, str_cve_statu, mos_cve_statu
go

```

Tabla : proveedor_transf.abct

```

create table proveedor_transf /*Tabla principal de proveedores*/
(prv_rzo_socia char(40) null, /*Razon social PKEY */
prv_rfc char(20) null, /*R.F.C del proveedor PKEY */
prv_num_pedid char(10) null, /*Num pedido MET-MEX */
prv_fec_pedid datetime null, /*Fecha pedido MET-MEX */
prv_fec_recib datetime null, /*Fecha recibo MET-MEX */
prv_pre_unita money null, /*Precio unitario */
prv_tpo_moned smallint null) /*Tiempo de moneda */
go

```

Tabla : proveedor_convenio.abct

```

create table proveedor_convenio
(prv_rfc char(20) not null,
lis_convenio char(4) not null,
prc_descripcion char(40) null)

```

```

create table partida_cotizacion
  (prv_rfc          char(20)          not null,
   cot_clave       char(12)          not null,
   alm_clave       char(4)           not null,
   req_num_conse   int                not null,
   prq_num_conse   smallint          not null,
   pco_pre_unita   float              not null,
   pco_cantidad    float              not null,
   ume_clave       char(5)           not null,
   pco_pso_unita   float              null,
   pco_ptj_dcto    float              null,
   pco_tpo_entre   smallint          null,
   des_clave       int                null)
go
create unique nonclustered index xp_partida_cotizacion on partida_cotizacion
  ( prv_rfc , cot_clave , alm_clave , req_num_conse , prq_num_conse )
go
exec sp_primarykey partida_cotizacion ,
  prv_rfc , cot_clave , alm_clave , req_num_conse , prq_num_conse
go
exec sp_foreignkey partida_cotizacion , cotizacion ,
  prv_rfc , cot_clave
go
exec sp_foreignkey partida_cotizacion , partida_requisicion ,
  alm_clave , req_num_conse , prq_num_conse
go

```

Tabla : partida_pedido_modifica.a

```

create table partida_pedido_modifica
  (alm_clave       char(4)          not null, /* Cve. de almacen */
   ped_clave       int              not null, /* Num. Requis/ Pedido PKEY */
   ped_num_conse   tinyint         not null, /* Numero de anexo */
   ppe_num_conse   smallint        not null, /* Num. consec. partida/pedido */
   req_num_conse   int              not null, /* Num. consec. requisicion */
   prq_num_conse   smallint        not null, /* Num. consec. partida req. */
   ppe_cantidad    float            not null, /* Cantidad requerida */
   ppe_pre_unita   money            not null, /* Precio unitario */
   ppe_ptj_dcto    float            null, /* Porcentaje de descuento */
   /* ppe_fec_entre  datetime        null, /* Fecha entrega de partida */
   des_clave       int              null, /* Descripcion partida */
   ppe_cotizada    tinyint         null, /* partida cotizada */
   ppe_cve_progr   char(1)         null, /* Indica partida programada */
   ppe_cve_statu   char(2)         not null, /* Status de la partida */
   ppe_fec_statu   datetime        not null) /* Fecha de estatus */
go
create unique nonclustered index xp_partida_pedido_modifica on
  partida_pedido_modifica
  ( alm_clave , ped_clave , ped_num_conse , ppe_num_conse )
go
sp_primarykey partida_pedido_modifica
  alm_clave , ped_clave , ped_num_conse , ppe_num_conse
go
sp_foreignkey partida_pedido_modifica , modifica_pedido ,
  alm_clave , ped_clave , ped_num_conse
go

```

```

create table plano_pedido
(alm_clave      char(4)      not null, /* Cve. de almacen */
 ped_clave      int          not null, /* Num. Requis/ Pedido PKEY */
 ped_num_conse  tinyint     not null, /* Numero de anexo */
 pla_num_conse  smallint    not null, /* Num. consec. plano */
 pla_status    char(4)      not null, /* Status del plano */
 pla_fec_statu datetime     not null) /* Fecha del status */
go
create unique nonclustered index xp_plano_pedido on plano_pedido
( alm_clave , ped_clave , ped_num_conse , pla_num_conse )
go
sp_primarykey plano_pedido
alm_clave , ped_clave , ped_num_conse , pla_num_conse
go
sp_foreignkey plano_pedido , pedido ,
alm_clave , ped_clave , ped_num_conse
go

```

Tabla : partida_requisicion_trans

```

create table partida_requisicion_transf
( alm_clave      char(4)      null, /* Clave almacen FKEY PKEY */
 req_num_conse  int          null, /* Consec. requis FKEY PKEY */
 prq_num_conse  smallint    null, /* Partida requis FKEY */
 prq_can_reque  float       null, /* Cantidad solicitada */
 prq_can_coloc  float       null, /* Cantidad colocada */
 prq_des_indic  tinyint     null, /* Indica si unid, corpo o n */
 prq_cod_estat  char(2)     null, /* Clave de estatus actual */
 prq_fec_estat  datetime    null, /* Fecha de estatus actual */
 prq_fec_alta  fecha hoy,   null, /* Fecha de alta de registro */
 prq_num_descr  tinyint     null, /* Indica partidas programa */
 mun_clave     char(12)     null, /* Codigo matr unidad */
 com_clave     char(2)      null, /* Comprador */
 umc_clave     char(5)      null, /* Unidad de medida */
 lin_clave     tinyint     null, /* Clave de linea */
 lin_sub_linea char(6)      null, /* Clave de la Sublinea */
 mco_clave     char(14)    null, /* Codigo del material Corp */
 des_clave     int         null, /* FKEY a descripcion matr */
 prq_can_cance float       null) /* Cantidad cancelada */
go

```

Tabla : proveedor_linea_sublinea.

```

create table proveedor_linea_sublinea
( prv_rfc        char(20)    not null,
 lin_clave      tinyint     not null,
 lin_sub_linea  char(06)    null)
go
create unique nonclustered index sp_proveedor_linea_sublinea on
proveedor_linea_sublinea
( prv_rfc , lin_clave , lin_sub_linea )
go
exec sp_primarykey proveedor_linea_sublinea ,
pFv_rfc , lin_clave , lin_sub_linea
go
exec sp_foreignkey proveedor_linea_sublinea , proveedor ,
pFv_rfc , lin_clave , lin_sub_linea
go
exec sp_foreignkey proveedor_linea_sublinea , linea_sublinea ,
pFv_rfc , lin_clave , lin_sub_linea
go

```

```

create table reporte
(rep_clave char(20) not null, /* Reportes */
 rep_codigo char(10) null, /* Clave */
 rep_descripcion char(60) null, /* Tipo */
 rep_des_papel char(60) null, /* Descripcion del Papel que usa */
 rep_copias tinyint not null, /* Numero de copias de default */
 imp_clave char(20) not null, /* Impresora de default */
 rep_usa_grafi char(01) not null, /* Si usa caracteres graficos */
 mod_clave char(20) null, /* Modulo al que pertenece */
 rep_ruta char(50) null, /* Path donde se encuentra si es sqw */
 rep_opcion char(50) null) /* Opciones de imprecion lp -o"x" */
go
create unique nonclustered index xp_rep on reporte
(rep_clave)
go
exec sp_primarykey reporte, rep_clave
go

```

Tabla : resumen.abct

```

create table resumen
(com_clave char(2) not null, /* Cve. del comprador */
 alm_clave char(4) not null, /* Cve de almacen FKEY PKEY */
 res_periodo int not null, /* Periodo de proceso PKEY */
 res_requis int not null, /* Total de requisiciones */
 res_part_req int not null, /* Total de partidas de requis. */
 res_req_cancel int not null, /* Total de requis. canceladas. */
 res_req_periodo int not null, /* Total de requis. del periodo */
 res_ped_nacio int not null, /* Total de pedidos nacion. */
 res_part_ped_nacio int not null, /* Total de part. de pedidos nac */
 res_ped_nacio_imp int not null, /* Total de ped. nacio. importe */
 res_ped_import int not null, /* Total de ped. de import. */
 res_ped_import_part int not null, /* Total de part. de ped. imp. */
 res_ped_import_imp int not null) /* Total de ped. imprt. importe */
go
create unique nonclustered index xp_resumen on resumen
( com_clave, alm_clave, res_periOdo )
go
/* llaves involucradas */
exec sp_primarykey resumen,
com_clave, alm_clave, res_periOdo
go

```

Tabla : solicitud_cotizacion.abct

```

create table solicitud_cotizacion
(sco_num_conse int not null,
 sco_fec_sol datetime not null,
 sco_fec_lim_cotiz datetime not null,
 des_clave int null)
go
create unique nonclustered index xp_solicitud_cotizacion on
solicitud_cotizacion
( sco_num_conse )
go
exec sp_primarykey solicitud_cotizacion,
sco_num_conse
go

```



```

drop index solicitud_transporte.xp_sol_trans
go
drop table solicitud_transporte
go
create table solicitud_transporte
(sot_num_conse int not null, /*Consecutivo de solicitud */
 agt_clave char(02) not null, /*Clave del agente aduanal */
 tra_clave char(14) not null, /*Numero de trafico */
 sot_dim_largo float not null, /*Largo del bulto */
 sot_dim_ancho float not null, /*Ancho del bulto */
 sot_dim_alto float not null, /*Alto del bulto */
 sot_ume_solic char(04) not null, /*Unidad medida de dimensiones*/
 sot_tip_mecia char(10) not null, /*Tipo de mercancia */
 sot_dom_agent varchar(255) not null, /*Lugar de entrega de mcia. */
 sot_rem_horar char(10) not null, /*Horario de carga */
 sot_des_horar char(10) not null, /*Horario de descarga */
 sot_notas varchar(255) not null, /*Observaciones de transporte */
 sot_lin_trans char(04) not null, /*Linea transportista empleada*/
 sot_gui_talon char(14) not null, /*No. talon de identificacion */
 sot_tra_placa char(10) not null, /*No. de placas de tractor */
 sot_num_econo int not null, /*No. economico de camion */
 sot_caj_numero char(10) not null, /*No. de cajas del tractor */
 sot_fec_alta datetime not null, /*Fecha de alta solicitud */
 sot_fec_asign datetime not null, /*Fecha asignación linea trans*/
 sot_fec_embar datetime not null, /*Fecha de embarque */
 sot_fec_cruce datetime not null, /*Fecha de cruce */
 sot_fec_coloc datetime not null, /*Fecha de coloc autotrans */
 sot_operador char(40) not null, /*Nombre operador de unidad */
 com_clave char(02) not null) /*Clave de usuario */
go
create unique nonclustered index xp_sol_trans on solicitud_transporte
(sot_num_conse)
go
exec sp_primarykey solicitud_transporte , sot_num_conse
go

```

Tabla : status.abct

```

create table status
(sta_tipo char(02) not null, /* Tipo status R, P, M */
 sta_clave char(02) not null, /* Clave del status */
 sta_nombre varchar(40) not null, /* Nombre del status */
 sta_nivel tinyint not null, /* Nivel de proceso */
 sta_cve_depen char(02) null) /* Status de quien depende */
go
create unique clustered index xp_sta_clave on status
(sta_tipo, sta_clave)
go
exec sp_primarykey status,
sta_tipo, sta_clave
go

```

Tabla : status_trafico.abct

```

create table status_trafico
(agt_clave char(2) not null, /*Datos generales del tr fico */
 tra_clave char(14) not null, /*Clave del agente aduanal PKEY */
 str_cve_statu char(02) not null, /*Número de tráfico PKEY */
 str_nivel1 tinyint not null, /*Status del trafico */
 str_fec_statu datetime not null, /*Nivel Gerarquico del Status */
 /*Fecha de el status */
)
go
create unique nonclustered index xp_sta_trafi on status_trafico
(agt_clave, tra_clave, str_cve_statu)
exec sp_primarykey sEatus_trafiGo, agt_clave, tra_clave, str_cve_statu

```

```

drop index solicitud_transporte_pedido.xp_sol_trans_ped
go
drop table solicitud_transporte_pedido
go
create table solicitud_transporte_pedido
(sot_num_conse tinyint not null, /*Consecutivo solicitud transporte*/
alm_clave char(4) not null, /*Clave de almacen*/
ped_clave int not null, /*No.consecutivo de requisicion*/
ped_num_conse tinyint not null) /*No.consecutivo (anexo) */
go
create unique nonclustered index xp_sol_trans_ped on solicitud_transporte_pedido
(sot_num_conse)
go
exec sp_primarykey solicitud_transporte_pedido , sot_num_conse
go

```

Tabla : solicitud_proveedor.abct

```

create table solicitud_proveedor
(sco_num_conse int not null, /* Numero consecutivo de solici. cot */
prv_rfc char(20) not null, /* RFC del proveedor */
prv_rfc_repre char(20) not null, /* RFC del proveedor solicitado */
pdo_num_conse tinyint not null, /* Numero consecutivo de la dir. */
cot_clave char(14) null) /* Numero de cotizac. de proveedor */
go
create unique nonclustered index xp_solicitud_proveedor on solicitud_proveedor
(sco_num_conse)
go
exec sp_primarykey solicitud_proveedor ,
sco_num_conse
go
exec sp_foreignkey solicitud_proveedor , proveedor ,
prv_rfc
go
exec sp_foreignkey solicitud_proveedor , domicilio_proveedor ,
prv_rfc_repre , pdo_num_conse
go

```

Tabla : solicitud_partida_requisi

```

create table solicitud_partida_requisicion
(sco_num_conse int not null,
alm_clave char(4) not null,
req_num_conse int not null,
prq_num_conse smallint not null,
des_clave int null)
go
create unique nonclustered index xp_solicitud_partida_requisicion on
solicitud_partida_requisicion
(sco_num_conse , alm_clave , req_num_conse , prq_num_conse )
go
exec sp_primarykey solicitud_partida_requisicion ,
sco_num_conse , alm_clave , req_num_conse , prq_num_conse
go
exec sp_foreignkey solicitud_partida_requisicion , solicitud ,
sco_num_conse
go
exec sp_foreignkey solicitud_partida_requisicion , partida_requisicion ,
alm_clave , req_num_conse , prq_num_conse
go

```

```

create table tabla
(tab_num_conse tinyint not null,
tab_nomBre char(40) not null)
go
create unique nonclustered index xp_tabla on tabla
(tab_num_conse)
go
exec sp_primarykey tabla ,
tab_num_conse
go

```

Tabla : trafico.abct

```

create table trafico /*Datos generales del tr fico
(agt_clave char(2) not null, /*Clave del agente aduanal PKEY */
tra_clave char(14) not null, /*Número de tráfico PKEY */
tra_tip_opera char(2) null, /*Tipo de operación aduanal */
tra_fec_llega datetime not null, /*fecha de llegada */
tra_cla_asoci char(14) not null) /*asociación con otros tráfico */
go
create unique nonclustered index xp_trafico on trafico
(agt_clave, tra_clave)
go
exec sp_primarykey trafico, agt_clave, tra_clave
go
exec sp_foreignkey trafico, agente_aduanal, agt_clave
go

```

Tabla : telefono_proveedor.abct

```

create table telefono_proveedor /*Domicilio del prov. y sucur. */
(prv_rfc char(20) not null, /*R.F.C. del proveedor PKEY P */
prv_cve_prove char(1) not null, /*B= Bancos, P= Proveedor */
prv_num_conse tinyint not null, /*Numero Consecutivo de telefo.*/
prv_telefono char(14) null, /*Telefono */
prv_fax char(14) null) /*Fax */
go
create unique nonclustered index xp_telefono_proveedor on telefono_proveedor
(prv_rfc , prv_cve_prove , prv_num_conse )
go
exec sp_primarykey telefono_proveedor ,
prv_rfc , prv_cve_prove , prv_num_conse
go
exec sp_foreignkey telefono_proveedor , proveedor ,
prv_rfc
go

```

Tabla : tipo_impresora.abct

```

create table tipo_impresora /*Tipos de Impresoras */
(tim_clave tinyint not null, /* Clave del tipo */
tim_cla_impre char(40) not null, /* Clase */
tim_mod_impre char(14) not null) /* Modelo */
go
create unique nonclustered index xp_tipo_imp on tipo_impresora
(tim_clave)
go
exec sp_primarykey tipo_impresora, tim_clave
go

```

```

create table almacen
( uni_clave      char(4)      not null, /* clave de unidad FKEY */
  alm_clave      char(4)      not null, /* calve de almacen PKEY */
  alm_nombre     varchar(40)  not null, /* Nombre del almacen */
  alm_nom_respo  varchar(40)  not null, /* Nombre de almacenista */
  alm_pto_respo  varchar(40)  not null, /* Puesto del responsable */
  alm_cve_indic  char(1)      null, /* tipo alm. p/capt. req */
  alm_cta_cuent  smallint     null, /* Mayor cuenta de cargo */
  alm_cta_subct  null, /* Subcuenta */
  alm_cta_ssubc  tinyint      null, /* Subsubcuenta */
  alm_cta_conce  smallint     null, /* Concepto */
  alm_fac_alta  fecha hoy     not null, /* Fecha de Alta */
  alm_cons_sol  smallInt      not null, /* Num. consecutivo de remicion */
  alm_remiTente varchar(40)    null)
go
create unique clustered index xp_almacen on almacen
( uni_clave, alm_clave )
go
exec sp_primarykey almacen ,
      uni_clave, alm_clave
go
exec sp_foreignkey almacen , unidad ,
      uni_clave
go
drop trigger abinalm01_ins
go
drop trigger abinalm01_del
go
drop trigger abinalm01_upd
go
/* TRIGGER ALMACEN ----INSERT ***** */
create trigger abinalm01_ins on almacen for insert
as
begin
  if @@rowcount = 0
    return
/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/
  exec abvalusu_pro
  if not exists(select * from inserted ins, unidad uni
               where ins.uni_clave = uni.uni_clave)
    begin
      raiserror 20110 'No existe unidad'
      rollback tran
      return
    end
end
go
/* TRIGGER ALMACEN---DELETE ***** */
create trigger abinalm01_del on almacen for delete
as
begin
  if @@rowcount = 0
    return
/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/
  exec abvalusu_pro
/* Si la unidad tiene almacenes vigentes, rollback tran */

```

```

if exists (select * from deleted de, requisicion req
where de.alm_clave = req.alm_clave)
begin
    raiserror 20020 'Existen Requisiciones de este Almacen'
    rollback tran
    return
end

/* elimina domicilios asociados al almacen */
delete almacen_domicilio from almacen domicilio dom,
                        deleted de
where dom.alm_clave = de.alm_clave

end
go

/* TRIGGER      ALMACEN---UPDATE      ***** */
create trigger abinalm01_upd on almacen for update
as
begin
    if @@rowcount = 0
        return

/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/

exec abvalusu_pro

/* No se permite cambio de Almacen      */
if (update (alm_clave))
begin
    raiserror 20050 'No se permite cambio de almacen.'
    rollback tran
    return
end
end
go

```

```

create table almacen_domicilio          /* Tabla asociativa almacen domicilio */
( alm_clave      char(4)      not null, /* Clave almacen FKEY PKEY */
  ald_tipo      char(1)      not null, /* Tipo dom Fac/Cons PKEY */
  ald_num_conse tinyint      not null, /* No. consecutivo del tip. */
  ald_clave     tinyint      not null) /* Clave domicilio FKEY PKEY */
go
create unique clustered index xp_almacen_domicilio on almacen_domicilio
( alm_clave , ald_tipo , ald_num_conse, ald_clave)
go
exec sp_primarykey almacen_domicilio
    alm_clave , ald_tipo , ald_num_conse, alm_clave
go
exec sp_foreignkey almacen_domicilio , almacen ,
    alm_clave
go

drop trigger abalmdo01_ins
go
drop trigger abalmdo01_del
go
drop trigger abalmdo01_upd
go

/* TRIGGER          ALMA/DOMIC ----INSERT          ***** */
create trigger abalmdo01_ins on almacen_domicilio for insert
as
begin
    declare @rowcount int
    select @rowcount = @@rowcount

    if @rowcount = 0
        return

    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    */

    exec abvalusu_pro

/* verifica existencia del almacen */
if not exists(select * from inserted ins, almacen alm
              where ins.alm_clave = alm.alm_clave)
    begin
        raiserror 20010 'No existe almacen'
        rollback tran
        return
    end
end
go

/* TRIGGER          ALMA/DOMIC---DELETE          ***** */
create trigger abalmdo01_del on almacen_domicilio for delete
as
begin
    declare @rowcount int
    select @rowcount = @@rowcount

    if @rowcount = 0
        return

    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    */

    exec abvalusu_pro

end
go

```

```

/* TRIGGER          ALMA/DOMIC---UPDATE          ***** */
create trigger abalmdo01_upd on almacen_domicilio for update
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount
  if @rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvalusu_pro

  /* No se permite cambio de Almacen          */
  if (update (alm_clave))
  begin
    raiserror 20050 'No se permite cambio de Almacen.'
    rollback tran
    return
  end
end
go

```

```

create table compania
  ( div_clave      tinyint      not null, /* Cve division llave foranea */
    cia_clave      smallint     not null, /* Clave de compania PKEY      */
    cia_nombre     varchar(40)   not null, /* Nombre de compania         */
    cia_nom_corto  varchar(14)   not null, /* Nombre abreviado           */
    cia_nom_contr  varchar(40)   not null, /* Nombre de contralor de CIA */
    cia_dom_fisca  varchar(108)  not null, /* Domicilio fiscal 3 x 36    */
    cia_rfc        char(20)      not null, /* R.F.C. de compania         */
    cia_fec_alta  fecha hoy,     /* Fecha de alta de registro  */
    cia_nom_fisca2 varchar(108)   null) /* Domicilio fiscal 3 x 36    */
go
create unique clustered index xp_compania on compania
  ( cia_clave )
go
exec sp_primarykey compania ,
  cia_clave
go
exec sp_foreignkey compania , division ,
  div_clave
go

drop trigger abincia01_ins
go
drop trigger abincia01_del
go
drop trigger abincia01_upd
go

/* TRIGGER      COMPANIA ---INSERT      ***** */
create trigger abincia01_ins on compania for insert
as
begin
  if @@rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvaluu_pro
  if @@error != 0
    rollback tran

  if not exists(select * from inserted ins, division div
    where ins.div_clave = div.div_clave)
    begin
      raiserror 20110 'No existe division'
      rollback tran
      return
    end
end
go

/* TRIGGER      COMPANIA---DELETE      ***** */
create trigger abincia01_del on compania for delete
as
begin
  if @@rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvaluu_pro

  /* Si la compania tiene unidades vigentes, rollback tran */

```

```

if exists (select * from deleted de, unidad uni
where de.cia_clave = uni.cia_clave)
begin
    raiserror 20020 'Existen Unidades de esta Compania'
    rollback tran
    return
end
end
go

/* TRIGGER          COMPANIA---UPDATE          ***** */
create trigger abincia01_upd on compania for update
as
begin
    if @@rowcount = 0
        return

    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    */

    exec abvalusu pro
    if @@error != 0
        rollback tran

    /* No se permite cambio de Compania          */
    if (update (cia_clave))
        begin
            raiserror 20050 'No se permite cambio de Compania.'
            rollback tran
            return
        end
end
end
go

```

```

create table comprador
( ofi_clave char(4) not null, /* Cve de oficina llave foranea */
  com_clave char(2) not null, /* Clave de comp(usuario) PKEY */
  com_cve_usuar char(7) null, /* Cve de usuario entrada UNIX */
  com_clave_sup char(2) not null, /* Cve de comp.a quien reporta */
  com_nombre varchar(40) not null, /* Nombre de comp(usuario) */
  com_puesto tinyint null, /* Puesto del comp(usuario) */
  com_area char(2) null, /* Area usuaria(compr. o traf) */
  com_nom corto varchar(14) null, /* Nombre abreviado de comp */
  com_vig bit not null, /* Indicador de vigencia ON-OFF */
  com_lim anti importe not null, /* Monto limite p/anticipos */
  com_lim autor importe not null, /* Monto limite autoriz. pedido */
  com_nivel tinyint null, /* Cve. niv. de auto. p/proces */
  com_fec alta fecha hoy, /* Fecha de alata de registro */
  com_active tinyint not null) /* Reservadoso futuro */

go
create unique clustered index xp_comprador on comprador
( com_clave )

go
exec sp_primarykey comprador ,
com_clave

go
exec sp_foreignkey comprador , ofna_reginal ,
ofi_clave

go

drop trigger abincom01_ins
go
drop trigger abincom01_del
go
drop trigger abincom01_upd
go

/* TRIGGER COMPRADOR ----INSERT ***** */
create trigger abincom01_ins on comprador for insert
as
begin
  if @@rowcount = 0
    return

  if @@rowcount > 1
  begin
    raiserror 20040 'No se permite transaccion multiple'
    rollback tran
    return
  end

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvaluu_pro

  if not exists(select * from inserted ins, ofna regional ofi
    where ins.ofi_clave = ofi.ofi_clave and
    ofi.ofi_ind_oper = 1)
  begin
    raiserror 20030 'No existe oficina de compras'
    rollback tran
    return
  end
end
go

/* TRIGGER COMPRADOR---DELETE ***** */
create trigger abincom01_del on comprador for delete
as
begin

```

```

if @@rowcount = 0
return

if @@rowcount > 1
begin
raiserror 20040 'No se permite transaccion multiple'
rollback tran
return
end

/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/

exec abvalusu_pro

/* NO se permite delete de compradores si tiene requisiciones */

if exists (select * from deleted del, partida_requisicion prq
where del.com_clave = prq.com_clave)
begin
raiserror 20020 'Existen requisiciones de este comprador'
rollback tran
return
end

/* elimina rows de tabla asociativa */
delete compra_linea from compra_linea col, deleted da
where col.com_clave = da.com_clave
end
go

create table #user (clave char(7) not null)
go

/* TRIGGER          COMPRADOR---UPDATE      ***** */
create trigger abincom01_upd on comprador for update
as
begin
if @@rowcount = 0
return

if @@rowcount > 1
begin
raiserror 20040 'No se permite transaccion multiple'
rollback tran
return
end

if update (com_clave) or update (ofi_clave)
begin
raiserror 20050 'No se permite cambio de comprador'
rollback tran
return
end

/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/

if not exists(select * from #user where clave = user_name(-1))
begin
raiserror 20060 'No se permite actualizacion de usuario'
rollback tran
return
end
end
go

```

```

create table compra_linea /* Tabla asociativa comprad. maneja lineas */
( com_clave char(2) not null, /* Clave de compr. PKEY FKEY */
  lin_clave tinyint not null) /* Clave de linea PKEY FKEY */
go
create unique clustered index xp_compra_linea on compra_linea
( com_clave , lin_clave )
go
exec sp_primarykey compra_linea ,
      com_clave , lin_clave
go
exec sp_foreignkey compra_linea , comprador ,
      com_clave
go

drop trigger abcolin01_ins
go
drop trigger abmcolin1_del
go
drop trigger abcolin01_upd
go

/* TRIGGER COMPRA LINEA INSERT */
create trigger abcolin01_ins on compra_linea for insert
as
begin
  if @@rowcount = 0
    return
  exec abvalusu_pro /* verifica actualizacion desde aplicacion */

  if not exists(select * from inserted ins, comprador com,
                linea_sublinea lin
                where ins.lin_clave = lin.lin_clave and
                      ins.com_clave = com.com_clave)
  begin
    raiserror 20010 'No existe linea o comprador'
    rollback tran
    return
  end
end

end
go

/* TRIGGER COMPRA LINEA --DELETE */
create trigger abcolin01_del on compra_linea for delete
as
begin
  if @@rowcount = 0
    return
  exec abvalusu_pro /* verifica actualizacion desde aplicacion */

end
go

/* TRIGGER COMPRA LINEA ---UPDATE */
create trigger abcolin01_upd on compra_linea for update
as
begin
  raiserror 20010 'No se permite cambio de compra_linea'
  rollback tran
  return

end
go

```

```

create table division
( div_clave          tinyint          not null, /* Clave de division PKEY */
  div_nombre         varchar(40)      not null, /* Nombre de division */
  div_fec_alta       fecha_hoy)
go
create unique clustered index xp_division on division
( div_clave )
go
exec sp_primarykey division ,
      div_clave
go

drop trigger abindiv01_ins
go
drop trigger abindiv01_del
go
drop trigger abindiv01_upd
go

/* TRIGGER          DIVISION ----INSERT      ***** */
create trigger abindiv01_ins on division for insert
as
begin
  if @@rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvalusu_pro
end
go

/* TRIGGER          DIVISION---DELETE      ***** */
create trigger abindiv01_del on division for delete
as
begin
  if @@rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvalusu_pro

  /* Si la requisicion tiene partidas colocadas, rollback tran */

  if exists (select * from deleted de, compania co
             where de.div_clave = co.div_clave)
    begin
      raiserror 20020 'Existen Companias de esta Division'
      rollback tran
      return
    end
end
go

/* TRIGGER          DIVISION---UPDATE      ***** */
create trigger abindiv01_upd on division for update
as
begin
  if @@rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

```

```

create table domicilio_temporal
( alm_clave      char(4)      not null, /* Cve de almacen FREY PKEY */
  ped_clave      int          not null, /* Num consec. requis. PKEY */
  ped_num_conse  tinyint      not null, /* Num. consec (anexo) PKEY */
  dom_nomBre     varchar(255) not null, /* Domicilio 3 x 36 */
  dom_tel        telefono     null, /* Telefono de domicilio */
  dom_fax        telefono     null, /* Fax de domicilio */
  dom_tipo       tinyint      null) /* Tipo de domic. 1=Fac 2=con*/
go
/* create unique clustered index xp_domicilio_temporal on domicilio_temporal
( alm_clave, ped_clave, ped_num_conse, dom_tipo )
go
exec sp_primarykey domicilio_temporal ,
alm_clave, ped_clave, ped_num_conse, dom_tipo
go */

/* TRIGGER          DOMICILIO-TEMPORAL---INSERT      ***** */
create trigger abdotin01_ins on domicilio_temporal for insert
as
begin
declare @rowcount int
select @rowcount = @@rowcount
if @rowcount = 0
return

/* verifica existencia de llaves foraneas */
if @rowcount = (select count(*) from inserted ins, pedido pe
               where ins.alm_clave = pe.alm_clave and
                     ins.ped_clave = pe.ped_clave and
                     ins.ped_num_conse = pe.ped_num_conse )
begin
raiserror 20100
'Falla de integridad en llaves foraneas '
rollback tran
return
end
end
go

/* TRIGGER          DOMICILIO-TEMPORAL---UPDATE      ***** */
create trigger abdotin01_upd on domicilio_temporal for update
as
begin
if @@rowcount = 0
return

if (update (alm_clave) or update (ped_clave) or
    update (ped_num_conse))
begin
raiserror 20050 'No se permite cambio de Centro o Numero de ped.'
rollback tran
return
end
end
go

```

```

create table domicilio_temporal_modifica
( alm_clave      char(4)      not null, /* Cve de almacen FREY PKEY */
  ped_clave      int          not null, /* Num consec. requis. PKEY */
  ped_num_conse  tinyint     not null, /* Num. consec (anexo) PKEY */
  dom_nombre     varchar(108) not null, /* Domicilio 3 x 36 */
  dom_tel        telefono    null, /* Telefono de domicilio */
  dom_fax        telefono    null) /* Fax de domicilio */

go
create unique clustered index xp domicilio_temporal_modifica
  on domicilio_temporal_modifica
  ( alm_clave, ped_clave, ped_num_conse )

go
exec sp_primarykey domicilio_temporal_modifica ,
  alm_clave, ped_clave, ped_num_conse

go

/* TRIGGER DOMICILIO-TEMPORAL-MODIFICA---INSERT ***** */
create trigger abdomin01_ins on domicilio_temporal_modifica for insert
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount
  if @rowcount = 0
    return

  /* verifica existencia de llaves foraneas */
  if @rowcount = (select count(*) from inserted ins, pedido pe,
    where ins.alm_clave = pe.alm_clave and
          ins.ped_clave = pe.ped_clave and
          ins.ped_num_conse = pe.ped_num_conse )
  begin
    raiserror 20100
    'Falla de integridad en llaves foraneas '
    rollback tran
    return
  end
end
go

/* TRIGGER DOMICILIO-TEMPORAL-MODIFICA---UPDATE ***** */
create trigger abdomin01_upd on domicilio_temporal_modifica for update
as
begin
  if @@rowcount = 0
    return
  begin
    raiserror 20050 'No se permite cambios '
    rollback tran
    return
  end
end
go

```

```

CREATE TABLE abaste.dbo.historico (
com_clave char(2) NOT NULL,
his_año smallint NOT NULL,
his_mes smallint NOT NULL,
his_tot_req_01 smallint NULL,
his_par_req_01 smallint NULL,
his_can_req_01 smallint NULL,
his_col_req_01 smallint NULL,
his_tot_req_02 smallint NULL,
his_par_req_02 smallint NULL,
his_can_req_02 smallint NULL,
his_col_req_02 smallint NULL,
his_tot_req_03 smallint NULL,
his_par_req_03 smallint NULL,
his_can_req_03 smallint NULL,
his_col_req_03 smallint NULL,
his_tot_req_04 smallint NULL,
his_par_req_04 smallint NULL,
his_can_req_04 smallint NULL,
his_col_req_04 smallint NULL,
his_tot_req_05 smallint NULL,
his_par_req_05 smallint NULL,
his_can_req_05 smallint NULL,
his_col_req_05 smallint NULL,
his_tot_req_06 smallint NULL,
his_par_req_06 smallint NULL,
his_can_req_06 smallint NULL,
his_col_req_06 smallint NULL,
his_tot_req_07 smallint NULL,
his_par_req_07 smallint NULL,
his_can_req_07 smallint NULL,
his_col_req_07 smallint NULL,
his_tot_req_08 smallint NULL,
his_par_req_08 smallint NULL,
his_can_req_08 smallint NULL,
his_col_req_08 smallint NULL,
his_tot_req_09 smallint NULL,
his_par_req_09 smallint NULL,
his_can_req_09 smallint NULL,
his_col_req_09 smallint NULL,
his_tot_req_10 smallint NULL,
his_par_req_10 smallint NULL,
his_can_req_10 smallint NULL,
his_col_req_10 smallint NULL,
his_tot_req_11 smallint NULL,
his_par_req_11 smallint NULL,
his_can_req_11 smallint NULL,
his_col_req_11 smallint NULL,
his_tot_req_12 smallint NULL,
his_par_req_12 smallint NULL,
his_can_req_12 smallint NULL,
his_col_req_12 smallint NULL,
his_tot_req_13 smallint NULL,
his_par_req_13 smallint NULL,
his_can_req_13 smallint NULL,
his_col_req_13 smallint NULL,
his_tot_req_14 smallint NULL,
his_par_req_14 smallint NULL,
his_can_req_14 smallint NULL,
his_col_req_14 smallint NULL,
his_tot_req_15 smallint NULL,
his_par_req_15 smallint NULL,
his_can_req_15 smallint NULL,
his_col_req_15 smallint NULL,
his_tot_req_16 smallint NULL,
his_par_req_16 smallint NULL,
his_can_req_16 smallint NULL,
his_col_req_16 smallint NULL,

```

his_tot_req_17	smallint	NULL,
his_par_req_17	smallint	NULL,
his_can_req_17	smallint	NULL,
his_col_req_17	smallint	NULL,
his_tot_req_18	smallint	NULL,
his_par_req_18	smallint	NULL,
his_can_req_18	smallint	NULL,
his_col_req_18	smallint	NULL,
his_tot_req_19	smallint	NULL,
his_par_req_19	smallint	NULL,
his_can_req_19	smallint	NULL,
his_col_req_19	smallint	NULL,
his_tot_req_20	smallint	NULL,
his_par_req_20	smallint	NULL,
his_can_req_20	smallint	NULL,
his_col_req_20	smallint	NULL,
his_tot_req_21	smallint	NULL,
his_par_req_21	smallint	NULL,
his_can_req_21	smallint	NULL,
his_col_req_21	smallint	NULL,
his_tot_req_22	smallint	NULL,
his_par_req_22	smallint	NULL,
his_can_req_22	smallint	NULL,
his_col_req_22	smallint	NULL,
his_tot_req_23	smallint	NULL,
his_par_req_23	smallint	NULL,
his_can_req_23	smallint	NULL,
his_col_req_23	smallint	NULL,
his_tot_req_24	smallint	NULL,
his_par_req_24	smallint	NULL,
his_can_req_24	smallint	NULL,
his_col_req_24	smallint	NULL,
his_tot_req_25	smallint	NULL,
his_par_req_25	smallint	NULL,
his_can_req_25	smallint	NULL,
his_col_req_25	smallint	NULL,
his_tot_req_26	smallint	NULL,
his_par_req_26	smallint	NULL,
his_can_req_26	smallint	NULL,
his_col_req_26	smallint	NULL,
his_tot_req_27	smallint	NULL,
his_par_req_27	smallint	NULL,
his_can_req_27	smallint	NULL,
his_col_req_27	smallint	NULL,
his_tot_req_28	smallint	NULL,
his_par_req_28	smallint	NULL,
his_can_req_28	smallint	NULL,
his_col_req_28	smallint	NULL,
his_tot_req_29	smallint	NULL,
his_par_req_29	smallint	NULL,
his_can_req_29	smallint	NULL,
his_col_req_29	smallint	NULL,
his_tot_req_30	smallint	NULL,
his_par_req_30	smallint	NULL,
his_can_req_30	smallint	NULL,
his_col_req_30	smallint	NULL,
his_tot_req_31	smallint	NULL,
his_par_req_31	smallint	NULL,
his_can_req_31	smallint	NULL,
his_col_req_31	smallint	NULL,

) on 'default'

go

create unique clustered index his_index
on historico (com_clave, his_año, his_mes)

go

```

CREATE TABLE abaste.dbo.historico_pedido (
com clave          Char(2)          NOT NULL,
his_año            smallint        NOT NULL,
his_mes            smallint        NOT NULL,
his_ped_nal_01    smallint        NULL,
his_ped_imp_01    smallint        NULL,
his_par_nal_01    smallint        NULL,
his_par_imp_01    smallint        NULL,
his_cer_ped_01    smallint        NULL,
his_ped_nal_02    smallint        NULL,
his_ped_imp_02    smallint        NULL,
his_par_nal_02    smallint        NULL,
his_par_imp_02    smallint        NULL,
his_cer_ped_02    smallint        NULL,
his_ped_nal_03    smallint        NULL,
his_ped_imp_03    smallint        NULL,
his_par_nal_03    smallint        NULL,
his_par_imp_03    smallint        NULL,
his_cer_ped_03    smallint        NULL,
his_ped_nal_04    smallint        NULL,
his_ped_imp_04    smallint        NULL,
his_par_nal_04    smallint        NULL,
his_par_imp_04    smallint        NULL,
his_cer_ped_04    smallint        NULL,
his_ped_nal_05    smallint        NULL,
his_ped_imp_05    smallint        NULL,
his_par_nal_05    smallint        NULL,
his_par_imp_05    smallint        NULL,
his_cer_ped_05    smallint        NULL,
his_ped_nal_06    smallint        NULL,
his_ped_imp_06    smallint        NULL,
his_par_nal_06    smallint        NULL,
his_par_imp_06    smallint        NULL,
his_cer_ped_06    smallint        NULL,
his_ped_nal_07    smallint        NULL,
his_ped_imp_07    smallint        NULL,
his_par_nal_07    smallint        NULL,
his_par_imp_07    smallint        NULL,
his_cer_ped_07    smallint        NULL,
his_ped_nal_08    smallint        NULL,
his_ped_imp_08    smallint        NULL,
his_par_nal_08    smallint        NULL,
his_par_imp_08    smallint        NULL,
his_cer_ped_08    smallint        NULL,
his_ped_nal_09    smallint        NULL,
his_ped_imp_09    smallint        NULL,
his_par_nal_09    smallint        NULL,
his_par_imp_09    smallint        NULL,
his_cer_ped_09    smallint        NULL,
his_ped_nal_10    smallint        NULL,
his_ped_imp_10    smallint        NULL,
his_par_nal_10    smallint        NULL,
his_par_imp_10    smallint        NULL,
his_cer_ped_10    smallint        NULL,
his_ped_nal_11    smallint        NULL,
his_ped_imp_11    smallint        NULL,
his_par_nal_11    smallint        NULL,
his_par_imp_11    smallint        NULL,
his_cer_ped_11    smallint        NULL,
his_ped_nal_12    smallint        NULL,
his_ped_imp_12    smallint        NULL,
his_par_nal_12    smallint        NULL,
his_par_imp_12    smallint        NULL,
his_cer_ped_12    smallint        NULL,
his_ped_nal_13    smallint        NULL,
his_ped_imp_13    smallint        NULL,
his_par_nal_13    smallint        NULL,
his_par_imp_13    smallint        NULL,

```

his_cer_ped 13	smallint	NULL,
his_ped_nal 14	smallint	NULL,
his_ped_imp 14	smallint	NULL,
his_par_nal 14	smallint	NULL,
his_par_imp 14	smallint	NULL,
his_cer_ped 14	smallint	NULL,
his_ped_nal 15	smallint	NULL,
his_ped_imp 15	smallint	NULL,
his_par_nal 15	smallint	NULL,
his_par_imp 15	smallint	NULL,
his_cer_ped 15	smallint	NULL,
his_ped_nal 16	smallint	NULL,
his_ped_imp 16	smallint	NULL,
his_par_nal 16	smallint	NULL,
his_par_imp 16	smallint	NULL,
his_cer_ped 16	smallint	NULL,
his_ped_nal 17	smallint	NULL,
his_ped_imp 17	smallint	NULL,
his_par_nal 17	smallint	NULL,
his_par_imp 17	smallint	NULL,
his_cer_ped 17	smallint	NULL,
his_ped_nal 18	smallint	NULL,
his_ped_imp 18	smallint	NULL,
his_par_nal 18	smallint	NULL,
his_par_imp 18	smallint	NULL,
his_cer_ped 18	smallint	NULL,
his_ped_nal 19	smallint	NULL,
his_ped_imp 19	smallint	NULL,
his_par_nal 19	smallint	NULL,
his_par_imp 19	smallint	NULL,
his_cer_ped 19	smallint	NULL,
his_ped_nal 20	smallint	NULL,
his_ped_imp 20	smallint	NULL,
his_par_nal 20	smallint	NULL,
his_par_imp 20	smallint	NULL,
his_cer_ped 20	smallint	NULL,
his_ped_nal 21	smallint	NULL,
his_ped_imp 21	smallint	NULL,
his_par_nal 21	smallint	NULL,
his_par_imp 21	smallint	NULL,
his_cer_ped 21	smallint	NULL,
his_ped_nal 22	smallint	NULL,
his_ped_imp 22	smallint	NULL,
his_par_nal 22	smallint	NULL,
his_par_imp 22	smallint	NULL,
his_cer_ped 22	smallint	NULL,
his_ped_nal 23	smallint	NULL,
his_ped_imp 23	smallint	NULL,
his_par_nal 23	smallint	NULL,
his_par_imp 23	smallint	NULL,
his_cer_ped 23	smallint	NULL,
his_ped_nal 24	smallint	NULL,
his_ped_imp 24	smallint	NULL,
his_par_nal 24	smallint	NULL,
his_par_imp 24	smallint	NULL,
his_cer_ped 24	smallint	NULL,
his_ped_nal 25	smallint	NULL,
his_ped_imp 25	smallint	NULL,
his_par_nal 25	smallint	NULL,
his_par_imp 25	smallint	NULL,
his_cer_ped 25	smallint	NULL,
his_ped_nal 26	smallint	NULL,
his_ped_imp 26	smallint	NULL,
his_par_nal 26	smallint	NULL,
his_par_imp 26	smallint	NULL,
his_cer_ped 26	smallint	NULL,
his_ped_nal 27	smallint	NULL,
his_ped_imp 27	smallint	NULL,
his_par_nal 27	smallint	NULL,
his_par_imp 27	smallint	NULL,
his_cer_ped 27	smallint	NULL,
his_ped_nal 28	smallint	NULL,
his_ped_imp 28	smallint	NULL,
his_par_nal 28	smallint	NULL,
his_par_imp 28	smallint	NULL,
his_cer_ped 28	smallint	NULL,

```

his_ped_nal_29          smallint          NULL,
his_ped_imp_29         smallint          NULL,
his_par_nal_29         smallint          NULL,
his_par_imp_29         smallint          NULL,
his_cer_ped_29         smallint          NULL,
his_ped_nal_30         smallint          NULL,
his_ped_imp_30         smallint          NULL,
his_par_nal_30         smallint          NULL,
his_par_imp_30         smallint          NULL,
his_cer_ped_30         smallint          NULL,
his_ped_nal_31         smallint          NULL,
his_ped_imp_31         smallint          NULL,
his_par_nal_31         smallint          NULL,
his_par_imp_31         smallint          NULL,
his_cer_ped_31         smallint          NULL
) on 'default'
go

create unique clustered index his_index
on historico_pedido (com_clavē, his_año, his_mes)
go

```



```

CREATE TABLE abaste.dbo.his_ped_mensual (
  com_clave          char(2)          NOT NULL,
  his_año            smallint         NOT NULL,
  his_ped_nal_01    smallint         NULL,
  his_ped_imp_01    smallint         NULL,
  his_par_nal_01    smallint         NULL,
  his_par_imp_01    smallint         NULL,
  his_cer_ped_01    smallint         NULL,
  his_ped_nal_02    smallint         NULL,
  his_ped_imp_02    smallint         NULL,
  his_par_nal_02    smallint         NULL,
  his_par_imp_02    smallint         NULL,
  his_cer_ped_02    smallint         NULL,
  his_ped_nal_03    smallint         NULL,
  his_ped_imp_03    smallint         NULL,
  his_par_nal_03    smallint         NULL,
  his_par_imp_03    smallint         NULL,
  his_cer_ped_03    smallint         NULL,
  his_ped_nal_04    smallint         NULL,
  his_ped_imp_04    smallint         NULL,
  his_par_nal_04    smallint         NULL,
  his_par_imp_04    smallint         NULL,
  his_cer_ped_04    smallint         NULL,
  his_ped_nal_05    smallint         NULL,
  his_ped_imp_05    smallint         NULL,
  his_par_nal_05    smallint         NULL,
  his_par_imp_05    smallint         NULL,
  his_cer_ped_05    smallint         NULL,
  his_ped_nal_06    smallint         NULL,
  his_ped_imp_06    smallint         NULL,
  his_par_nal_06    smallint         NULL,
  his_par_imp_06    smallint         NULL,
  his_cer_ped_06    smallint         NULL,
  his_ped_nal_07    smallint         NULL,
  his_ped_imp_07    smallint         NULL,
  his_par_nal_07    smallint         NULL,
  his_par_imp_07    smallint         NULL,
  his_cer_ped_07    smallint         NULL,
  his_ped_nal_08    smallint         NULL,
  his_ped_imp_08    smallint         NULL,
  his_par_nal_08    smallint         NULL,
  his_par_imp_08    smallint         NULL,
  his_cer_ped_08    smallint         NULL,
  his_ped_nal_09    smallint         NULL,
  his_ped_imp_09    smallint         NULL,
  his_par_nal_09    smallint         NULL,
  his_par_imp_09    smallint         NULL,
  his_cer_ped_09    smallint         NULL,
  his_ped_nal_10    smallint         NULL,
  his_ped_imp_10    smallint         NULL,
  his_par_nal_10    smallint         NULL,
  his_par_imp_10    smallint         NULL,
  his_cer_ped_10    smallint         NULL,
  his_ped_nal_11    smallint         NULL,
  his_ped_imp_11    smallint         NULL,
  his_par_nal_11    smallint         NULL,
  his_par_imp_11    smallint         NULL,
  his_cer_ped_11    smallint         NULL,
  his_ped_nal_12    smallint         NULL,
  his_ped_imp_12    smallint         NULL,
  his_par_nal_12    smallint         NULL,
  his_par_imp_12    smallint         NULL,
  his_cer_ped_12    smallint         NULL,
) on 'default'
go
create unique clustered index his_ped_mensual
on his_ped_mensual (com_clave, his_año)

```

```
CREATE TABLE abaste.dbo.historico_mensual_req (
  com_clave Char(2) NOT NULL,
  his_periodo smallint NOT NULL,
  his_req_mes smallint NULL,
  his_req_par_mes smallint NULL,
  his_req_ant smallint NULL,
  his_req_mes_12 smallint NULL,
  his_req_mes_11 smallint NULL,
  his_req_mes_10 smallint NULL,
  his_req_mes_09 smallint NULL,
  his_req_mes_08 smallint NULL,
  his_req_mes_07 smallint NULL,
  his_req_mes_06 smallint NULL,
  his_req_mes_05 smallint NULL,
  his_req_mes_04 smallint NULL,
  his_req_mes_03 smallint NULL,
  his_req_mes_02 smallint NULL,
  his_req_mes_01 smallint NULL,
  his_req_mes_00 smallint NULL,
  his_req_ven smallint NULL,
  his_req_tot smallint NULL,
  his_req_pun smallint NULL)
go
```

```
create unique clustered index historico_mensual_req
on historico_mensual_req (com_clave, his_periodo)
go
```

```

CREATE TABLE abaste.dbo.his_req_mensual (
  com_clave          char(2)          NOT NULL,
  his_año            smallint         NOT NULL,
  his_tot_req_01     smallint         NULL,
  his_par_req_01     smallint         NULL,
  his_can_req_01     smallint         NULL,
  his_col_req_01     smallint         NULL,
  his_tot_req_02     smallint         NULL,
  his_par_req_02     smallint         NULL,
  his_can_req_02     smallint         NULL,
  his_col_req_02     smallint         NULL,
  his_tot_req_03     smallint         NULL,
  his_par_req_03     smallint         NULL,
  his_can_req_03     smallint         NULL,
  his_col_req_03     smallint         NULL,
  his_tot_req_04     smallint         NULL,
  his_par_req_04     smallint         NULL,
  his_can_req_04     smallint         NULL,
  his_col_req_04     smallint         NULL,
  his_tot_req_05     smallint         NULL,
  his_par_req_05     smallint         NULL,
  his_can_req_05     smallint         NULL,
  his_col_req_05     smallint         NULL,
  his_tot_req_06     smallint         NULL,
  his_par_req_06     smallint         NULL,
  his_can_req_06     smallint         NULL,
  his_col_req_06     smallint         NULL,
  his_tot_req_07     smallint         NULL,
  his_par_req_07     smallint         NULL,
  his_can_req_07     smallint         NULL,
  his_col_req_07     smallint         NULL,
  his_tot_req_08     smallint         NULL,
  his_par_req_08     smallint         NULL,
  his_can_req_08     smallint         NULL,
  his_col_req_08     smallint         NULL,
  his_tot_req_09     smallint         NULL,
  his_par_req_09     smallint         NULL,
  his_can_req_09     smallint         NULL,
  his_col_req_09     smallint         NULL,
  his_tot_req_10     smallint         NULL,
  his_par_req_10     smallint         NULL,
  his_can_req_10     smallint         NULL,
  his_col_req_10     smallint         NULL,
  his_tot_req_11     smallint         NULL,
  his_par_req_11     smallint         NULL,
  his_can_req_11     smallint         NULL,
  his_col_req_11     smallint         NULL,
  his_tot_req_12     smallint         NULL,
  his_par_req_12     smallint         NULL,
  his_can_req_12     smallint         NULL,
  his_col_req_12     smallint         NULL,
) on 'default'
go

create unique clustered index his_req_mensual
  on his_req_mensual (com_clave, his_año)
go

```

```

create table linea_sublinea
  ( lin_clave tinyint not null, /* Numero de linea de matr.PKEY */
    lin_sub_linea char(6) null, /* Clave sublinea(marca) PKEY */
    lin_nombre varchar(60) not null) /* Nombre de Linea/Sublinea */
go
create unique clustered index xp_linea_sublinea on linea_sublinea
  ( lin_clave , lin_sub_linea )
go
exec sp_primarykey linea_sublinea ,
  lin_clave , lin_sub_linea
go
drop trigger abinlin01_ins
go
drop trigger abinlin01_del
go
drop trigger abinlin01_upd
go

/* TRIGGER LINEA_SUBLINEA ---INSERT ***** */
create trigger abinlin01_ins on linea_sublinea for insert
as
begin
  if @@rowcount = 0
    return
  if @@rowcount > 1
  begin
    raiserror 20030 'No se permite insercion multiple'
    rollback tran
    return
  end
  /* Verifica que la transaccion sea solicitada por medio del sistema
  * de Abastecimiento.
  */
  exec abvalusu_pro
end
go

/* TRIGGER LINEA_SUBLINEA---DELETE ***** */
create trigger abinlin01_del on linea_sublinea for delete
as
begin
  if @@rowcount = 0
    return
  /* NO se permite delete de linea_sublinea si tiene mat.corpor. */
  if exists(select * from deleted del, material_corporativo mco
    where del.lin_clave = mco.lin_clave and
    del.lin_sub_linea = mco.lin_sub_linea)
  begin
    raiserror 20020 'Existe material corporativo de esta linea'
    rollback tran
    return
  end
  /* elimina rows de tabla asociativa */
  delete compra_linea from compra_linea lin, deleted de
  where lin.lin_clave = de.lin_clave
end
go

/* TRIGGER LINEA_SUBLINEA---UPDATE ***** */
create trigger abinlin01_upd on linea_sublinea for update

```

```

create table material_corporativo
( lin_clave          tinyint      not null, /* Línea de matr1 FKEY */
  lin_sub_linea     char(6)       not null, /* Sublínea de matr1 FKEY */
  mco_clave         char(14)      not null, /* Código corporativo Mat. FKEY */
  ume_clave         char(5)       not null, /* Unidad de medida */
  des_clave         int           not null, /* FKEY a descripción material */
  mco_num_descr     tinyint       not null, /* Num. de regs de descripción */
  hcc_clave         tinyint       null, /* FKEY a compras anteriores */
  mco_num_hcc       tinyint       null, /* Num de registr.compr.ant. */
  muc_mat_corpo    int           not null, /* FKEY consec.asoc.matun matco */
  mco_fec_alta     fecha_hoy)    /* Fecha de alta de registro */
go
create unique nonclustered index xp_material_corporativo on
material_corporativo
( lin_clave, lin_sub_linea, mco_clave )
go
exec sp_primarykey material_corporativo ,
lin_clave, lin_sub_linea, mco_clave
go
exec sp_foreignkey material_corporativo , linea_sublinea ,
lin_clave , lin_sub_linea
go
exec sp_foreignkey material_corporativo , unidad_medida ,
ume_clave
go
drop trigger abinmco01_ins
go
drop trigger abinmco01_del
go
drop trigger abinmco01_upd
go
/* TRIGGER CODIGO/CORP ----INSERT ***** */
create trigger abinmco01_ins on material_corporativo for insert
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount
  if @rowcount = 0
    return
  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */
  exec abvalusu_pro
/* verifica existencia de linea sublinea */
if not exists(select * from Inserted ins, linea_sublinea lin
where ins.lin_clave = lin.lin_clave and
ins.lin_sub_linea = lin.lin_sub_linea)
begin
  raiserrorr 20010 'No existe linea_sublinea'
  rollback tran
  return
end
end
go
/* TRIGGER CODIGO/CORP----DELETE ***** */
create trigger abinmco01_del on material_corporativo for delete
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount

```

```

if @rowcount = 0
return
/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/
exec abvalusu_pro
if exists(select * from deleted del, partida requisicion prq
where del.mco_clave = prq.mco_clave and
del.lin_clave = prq.lin_clave)
begin
raiserror 20030 'Existen requisiciones de este material'
rollback tran
return
end
/* elimina rows de tabla asociativa */
delete matun matco from matun_matco mco, deleted de
where mco.muc_mat_corpo = de.muc_mat_corpo
/* incluir cuando exista compras anteriores de este material,
* el delete de tabla asociativa o tabla de compras cuando se
* de delete de material
*/
end
go
/* TRIGGER CODIGO/CORP---UPDATE ***** */
create trigger abinmco01_upd on material_corporativo for update
as
begin
declare @rowcount int
select @rowcount = @@rowcount
if @rowcount = 0
return
/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/
exec abvalusu_pro
/* No se permite cambio de Linea sublinea ni codigo */
if update (lin_clave) or update (lin_sub_linea) or
update (mco_clave) or update (muc_mat_corpo)
begin
raiserror 20050 'No se permite cambio de Linea/sublin o codigo.'
rollback tran
return
end
end
go

```

```

create table material_unidad
( alm_clave          cchar(4)          not null, /* Cve. almacen PKEY FKEY */
  mun_clave          char(12)         not null, /*Codigo de material PKEY */
  ume_clave          char(5)          not null, /* Unidad de medida de matr1 */
  mun_act_fijo       bit              not null, /* Indicador activo fijo ON-OFF */
  des_clave          int              not null, /* FKEY a tabla de descripcion */
  mun_num_desc       tinyint          not null, /* Num. de registr de descrip. */
  hcu_clave          tinyint          null, /* FKEY a compras anteriores */
  mun_num_hcu        tinyint          null, /* Num. de regist.d e compras an */
  muc_mat_unida     int              not null, /* FKEY consec.asoc.matun matco */
  mun_fec_alta      fecha_hoy)       /* Fecha de alta de registro */
go
create unique nonclustered index xp_material_unidad on material_unidad
( mun_clave )
go
exec sp_primarykey material_unidad ,
      mun_clave
go
exec sp_foreignkey material_unidad , almacen ,
      alm_clave
go
drop trigger abinmun01_ins
go
drop trigger abinmun01_del
go
drop trigger abinmun01_upd
go
/* TRIGGER          CODIGO/UNID ----INSERT          **** */
create trigger abinmun01_ins on material_unidad for insert
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount
  if @rowcount = 0
    return
  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */
  exec abvaluu_pro
/* verifica existencia de almacen */
if not exists(select * from inserted ins, almacen alm
              where ins.alm_clave = alm.alm_clave)
  begin
    raiserror 20010 'No existe almacen'
    rollback tran
    return
  end
end
go
/* TRIGGER          CODIGO/UNID---DELETE          **** */
create trigger abinmun01_del on material_unidad for delete
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount
  if @rowcount = 0
    return
  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.

```

```

*/
exec abvalusu_pro

if exists(select * from deleted del, partida_requisicion prq
         where del.mun_clave = prq.mun_clave)
begin
    raiserror 20030 'Existen requisiciones de este material'
    rollback tran
    return
end

/* elimina rows de tabla asociativa */
delete matun_matco from matun_matco muc, deleted de
where muc.muc_mat_unida = de.muc_mat_unida

/* incluir cuanto este compras anteriores, el delete de tabla asociativa
o tabla de este material
*/
end
go

/* TRIGGER          CODIGO/UNID---UPDATE          ***** */
create trigger abinmun01_upd on material_unidad for update
as
begin
    declare @rowcount int
    select @rowcount = @@rowcount

    if @rowcount = 0
        return

/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/
exec abvalusu_pro

/* No se permite cambio de almacen ni codigo */
if update (alm_clave) or update (mun_clave) or update (muc_mat_unida)
begin
    raiserror 20050 'No se permite cambio de almacen o codigo.'
    rollback tran
    return
end
end
go

```

```

create table matun_matco
  ( muc_mat_corpo int
    muc_mat_unida int
  /* Tabla asociativa Matuni Matcorpo */
  not null, /* Consec. Matrl.Corpor. PKEY */
  not null) /* Consec. Matrl.Unidad PKEY */
go
create unique nonclustered index xp_matun_matco on matun_matco
  ( muc_mat_corpo , muc_mat_unida )
go
exec sp_primarykey matun_matco ,
  muc_mat_corpo , muc_mat_unida
go
exec sp_foreignkey matun_matco , material_unidad ,
  muc_mat_unida
go
exec sp_foreignkey matun_matco , material_corporativo ,
  muc_mat_corpo
go
drop trigger abmumco01_ins
go
drop trigger abmumco01_del
go
drop trigger abmumco01_upd
go
/* TRIGGER MATUN MATCO INSERT */
create trigger abmumco01_ins on matun_matco for insert
as
begin
  if @@rowcount = 0
    return
  exec abvaluu_pro /* verifica actualizacion desde aplicacion */
  if not exists(select * from inserted ins, material_unidad mun,
    material_corporativo mco
    where ins.muc_mat_corpo = mco.muc_mat_corpo and
    ins.muc_mat_unida = mun.muc_mat_unida)
  begin
    raiserror 20010 'No existe material unidad o corporativo'
    rollback tran
    return
  end
end
end
go
/* TRIGGER MATUN MATCO --DELETE */
create trigger abmumco01_del on matun_matco for delete
as
begin
  if @@rowcount = 0
    return
  exec abvaluu_pro /* verifica actualizacion desde aplicacion */
end
end
go
/* TRIGGER MATUN MATCO ---UPDATE */
create trigger abmumco01_upd on matun_matco for update
as
begin
  raiserror 20010 'No se permite cambio de matun_matco'
  rollback tran
  return
end
end
go

```

```

create table moneda
( mon_clave      char(2)      not null, /* Clave de moneda      */
  mon_nom_espan  varchar(20)  not null, /* descrip. espanol   */
  mon_nom_ingle  varchar(20)  not null) /* descrip. ingles    */
go
create unique clustered index xp_moneda on moneda
( mon_clave )
go
exec sp_primarykey moneda,
      mon_clave
go

drop trigger abinmon01_ins
go
drop trigger abinmon01_del
go
drop trigger abinmon01_upd
go

/* TRIGGER          MONEDA ----INSERT      ***** */
create trigger abinmon01_ins on moneda for insert
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount

  if @rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvalusu_pro
end
go

/* TRIGGER          MONEDA ---DELETE      ***** */
create trigger abinmon01_del on moneda for delete
as
begin
  raiserror 20030 'No se permite eliminacion de moneda'
  rollback tran
  return
end
go

/* TRIGGER          MONEDA ---UPDATE      ***** */
create trigger abinmon01_upd on moneda for update
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount

  if @rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvalusu_pro

  /* No se permita cambio de almacen ni codigo      */
  if update (mon_clave)
    begin

```

```

create table notas_catalogo
( nta_clave      char(14)      not null, /* clave de Nota PKEY */
  nta_num_conse tinyint      not null, /* num. consecutivo PKEY */
  nta_fec_nota  datetime      not null, /* fecha de la nota */
  nta_nomBre    varchar(255)  not null) /* Descripcion de la nota */
go
create unique clustered index xp_notas_catalogo on notas_catalogo
( nta_clave , nta_num_conse )
go
exec sp_primarykey notas_catalogo ,
      nta_cave , nta_num_conse
go

drop trigger abinmun01_ins
go
drop trigger abinmun01_del
go
drop trigger abinmun01_upd
go

/* TRIGGER CODIGO/UNID ----INSERT ***** */
create trigger abinmun01_ins on material_unidad for insert
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount
  if @rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvalusu_pro

/* verifica existencia de almacen */
if not exists(select * from inserted ins, almacen alm
              where ins.alm_clave = alm.alm_clave)
  begin
    raiserror 20010 'No existe almacen'
    rollback tran
    return
  end
end
go

/* TRIGGER CODIGO/UNID---DELETE ***** */
create trigger abinmun01_del on material_unidad for delete
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount
  if @rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvalusu_pro

if exists(select * from deleted del, partida_requisicion prq
          where del.mun_clave = prq.mun_clave)
  begin
    raiserror 20030 'Existen requisiciones de este material'
    rollback tran
  end
end

```

```

        return
    end

/* incluir cuanto este compras anteriores, el delete de tabla asociativa
o tabla de este material
*/
end
go

/* TRIGGER )          CODIGO/UNID---UPDATE          ***** */
create trigger abinmun01_upd on material_unidad for update
as
begin
    declare @rowcount int
    select @rowcount = @@rowcount

    if @rowcount = 0
        return

/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/

    exec abvaluu_pro

/* No se permite cambio de almacen ni codigo          */
if update (alm_clave) or update (mun_clave)
    begin
        raiserror 20050 'No se permite cambio de almacen o codigo.'
        rollback tran
        return
    end
end
end
go

```

```

create table notas_pedido
(alm_clave      char(4)      not null, /* Clave de almacén      PKEY */
 ped_clave      int          not null, /* Número de pedido     PKEY */
 ped_num_conse  tinyint     not null, /* Número de anexo pedido PKEY */
 nta_clave      char(14)    not null, /* Clave de Nota        PKEY */
 nta_num_conse  tinyint     not null, /* Num. consecutivo     PKEY */
 nta_fec_nota   datetime    not null, /* Fecha de la nota     PKEY */
 nta_descripcion char(255) not null) /* Descripción de la nota */
go
create unique nonclustered index xp_notas_pedido on notas_pedido
(alm_clave, ped_clave, ped_num_conse, nta_clave, nta_num_conse)
go
exec sp_primarykey notas_pedido
alm_clave, ped_clave, ped_num_conse, nta_clave, nta_num_conse
go
exec sp_foreignkey notas_pedido, pedido
alm_clave, ped_clave, ped_num_conse
go
/* TRIGGER NOTAS-PEDIDO---INSERT ***** */
create trigger abnopin01_ins on notas_pedido for insert
as
begin
declare @rowcount int
select @rowcount = @@rowcount
if @rowcount = 0
return
/* verifica existencia de llaves foraneas */
if not exists (select count(*) from inserted ins, pedido pe
where ins.alm_clave = pe.alm_clave and
ins.ped_clave = pe.ped_clave and
ins.ped_num_conse = pe.ped_num_conse )
begin
raiserror 20100
'Falla de integridad en llaves foraneas '
rollback tran
return
end
end
go
/* TRIGGER NOTAS-PEDIDO---UPDATE ***** */
create trigger abnopin01_upd on notas_pedido for update
as
begin
if @@rowcount = 0
return
if (update (alm_clave) or update (ped_clave)) or
(update (ped_num_conse))
begin
raiserror 20050 'No se permite cambio de Centro o Numero de ped.'
rollback tran
return
end
end
go

```

```

create table notas_pedido_modifica
(alm_clave      char(4)      not null, /* clave de almacen PKEY */
 ped_clave     int          not null, /* numero de pedido PKEY */
 ped_num_conse tinyint      not null, /* numero de anexo pedido */
 nta_clave     char(14)     not null, /* clave de Nota */
 nta_num_conse tinyint      not null, /* num. consecutivo */
 nta_fec_nota  datetime     not null, /* fecha de la nota */
 nta_nombre    varchar(255) not null) /* Descripcion de la nota */
go
create unique nonclustered index xp_notas_pedido_modifica
on notas_pedido_modifica
( alm_clave , ped_clave , ped_num_conse , nta_clave , nta_num_conse )
go
exec sp_primarykey notas_pedido_modifica
alm_clave , ped_clave , ped_num_conse , nta_clave , nta_num_conse
go
/* TRIGGER NOTAS-PEDIDO-MODIFICA---INSERT ***** */
create trigger abnomin01_ins on notas_pedido_modifica for insert
as
begin
declare @rowcount int
select @rowcount = @@rowcount
if @rowcount = 0
return

/* verifica existencia de llaves foraneas */
if @rowcount = (select count(*) from inserted ins, pedido pe,
where ins.alm_clave = pe.alm_clave and
ins.ped_clave = pe.ped_clave and
ins.ped_num_conse = pe.ped_num_conse )
begin
raiserror 20100
'Falla de integridad en llaves foraneas '
rollback tran
return
end
end
go
/* TRIGGER NOTAS-PEDIDO-MODIFICA---UPDATE ***** */
create trigger abnomin01_upd on notas_pedido_modifica for update
as
begin
if @@rowcount = 0
return
raiserror 20050 'No se permite cambios'
rollback tran
return
end
go

```

```

create table notas_requisicion
(alm_clave char(4) not null, /* clave de almacen PKEY */
req_num_conse int not null, /* numero de requisic PKEY */
nta_clave char(14) not null, /* clave de Nota */
nta_num_conse tinyint not null, /* num. consecutivo */
nta_fec_notas datetime not null, /* fecha de la nota */
nta_nomBre varchar(255) not null) /* Descripcion de la nota */
go
create unique nonclustered index xp_notas_requisicion on notas_requisicion
(alm_clave, req_num_conse, nta_clave, nta_num_conse)
go
exec sp_primarykey notas_requisicion,
alm_clave, req_num_conse, nta_clave, nta_num_conse
go
exec sp_foreignkey notas_requisicion,
alm_clave, req_num_conse
go

drop trigger abnotrq01_ins
go
drop trigger abnotrq01_del
go
drop trigger abnotrq01_upd
go

/* TRIGGER NOTAS REQUISICION INSERT ***** */
create trigger abnotrq01_ins on notas_requisicion for insert
as
begin
if @@rowcount = 0
return

/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/
exec abvalusu_pro

if not exists (select * from inserted ins, requisicion req
where ins.alm_clave = req.alm_clave and
ins.req_num_conse = req.req_num_conse)
begin
raiserror 20090 'Requisicion no existe'
rollback tran
return
end
end
go

/* TRIGGER NOTAS REQUISICION ----DELETE ***** */
create trigger abnotrq01_del on notas_requisicion for delete
as
begin
if @@rowcount = 0
return

/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
*/
exec abvalusu_pro

end
go

/* TRIGGER NOTAS REQUISICION ----UPDATE ***** */
create trigger abnotrq01_upd on notas_requisicion for update
as
begin

```

```

create table ofna_regional
( ofi_clave      char(4)      not null, /* Todas las oficinas de compras */
  ofi_nombre    char(40)     not null, /* Nombre de oficina */
  ofi_dom_regio varchar(108)  not null, /* Domicilio de oficina */
  ofi_ind_oper  bit          not null, /* Indicador que oficina opera */
  ofi_fac_alta  fecha hoy,    not null, /* Fecha de alta de registro */
  ofi_dom_corr  varchar(108) null) /* Domicilio Correspondencia */
go
create unique clustered index xp_ofna_regional on ofna_regional
( ofi_clave )
go
exec sp_primarykey ofna_regional ,
  ofi_clave
go

drop trigger abinofr01_ins
go
drop trigger abinofr01_del
go
drop trigger abinofr01_upd
go

/* TRIGGER          OFNA REGIONAL ----INSERT      ***** */
create trigger abinofr0I_ins on ofna_regional for insert
as
begin
  if @@rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */

  exec abvaluu_pro

  if (select count(*) from ofna_regional
      where ofi_ind_oper = 1) > 1
    begin
      raiserror 20020 'Mas de una oficina en operacion'
      rollback tran
      return
    end
end
go

/* TRIGGER          OFNA REGIONAL---DELETE      ***** */
create trigger abinofr0I_del on ofna_regional for delete
as
begin
  if @@rowcount = 0
    return

  if exists(select * from deleted del, comprador com
            where del.ofi_clave = com.ofi_clave)
    begin
      raiserror 20020 'Existen usuarios de esta oficina '
      rollback tran
      return
    end
end
go

/* TRIGGER          OFNA REGIONAL---UPDATE      ***** */
create trigger abinofr0I_upd on ofna_regional for update
as
begin
  if @@rowcount = 0
    return

```

```

create table partida_pedido
(alm_clave      Char(4)      not null, /* Cve. de almacen          */
 ped_clave      int           not null, /* Num. Requis/ Pedido PKEY */
 ped_num_conse  tinyint      not null, /* Numero de anexo         */
 ppe_num_conse  smallint     not null, /* Num. consec. partida/pedido */
 req_num_conse  int          not null, /* Num. consec. requisicion  */
 prq_num_conse  smallint     not null, /* Num. consec. partida req.  */
 ppe_cantidad  float         not null, /* Cantidad requerida        */
 ppe_pre_unita  money        not null, /* Precio unitario           */
 ppe_ptj_dcto  float         not null, /* Porcentaje de descuento  */
 des_clave      int          null, /* Descripcion partida      */
 ppe_cotizada  tinyint      null, /* partida cotizada         */
 ppe_cve_prog  char(1)      null, /* Indica partida programada "S" */
 ppe_cve_statu char(2)      not null, /* Status de la partida     */
 ppe_fec_statu datetime     not null) /* Fecha de estatus        */

go
create unique nonclustered index xp_partida_pedido on partida_pedido
( alm_clave , ped_clave , ped_num_conse , ppe_num_conse )
go
sp_primarykey partida_pedido
alm_clave , ped_clave , ped_num_conse , ppe_num_conse
go
sp_foreignkey partida_pedido , pedido ,
alm_clave , ped_clave , ped_num_conse
go
sp_foreignkey partida_pedido , partida_requisicion ,
alm_clave , req_num_conse , prq_num_conse
go

/* asociacion de RULE */

sp_bindrule 'gtr_0' , 'partida_pedido.ppe_num_conse'
go
sp_bindrule 'gte_0' , 'partida_pedido.ppe_cantidad'
go
sp_bindrule 'gtr_0' , 'partida_pedido.ppe_pre_unita'
go
sp_bindrule 'ptj' , 'partida_pedido.ppe_ptj_dcto'
go
sp_bindrule 'Progr' , 'partida_pedido.ppe_cve_prog'
go

```

```

create table partida_requisicion
( alm_clave      Char(4)      not null, /* Clave almacen FKEY PKEY */
  req_num_conse  int          not null, /* Consec. requis FKEY PKEY */
  prq_num_conse  smallint     not null, /* Partida requis PKEY */
  prq_can_requs  float        not null, /* Cantidad solicitada */
  prq_can_coloc  float        not null, /* Cantidad colocada */
  prq_des_indic  tinyint      not null, /* Indica si unid, corpo o n */
  prq_cod_estat  char(2)      not null, /* Clave de estatus actual */
  prq_fec_estat  datetime     not null, /* Fecha de estatus actual */
  prq_fec_alta  fecha hoy,    not null, /* Fecha de alta de registro */
  prq_num_descr  tinyint      not null, /* Indica partidas programa */
  mun_clave      char(12)     null, /*Codigo matr1 unidad */
  com_clave      char(2)      not null, /* Comprador */
  ume_clave      char(5)      not null, /* Unidad de medida */
  lin_clave      tinyint      not null, /* Clave de linea */
  lin_sub_linea  char(6)      null, /* Clave de la Sublinea */
  mco_clave      char(14)     null, /*Codigo del material Corp */
  des_clave      int          null, /* FKEY a decripcion matr1 */
  prq_can_cance  float        null) /* Cantidad cancelada */

go
create unique nonclustered index xp_partida_requisicion on partida_requisicion
( alm_clave , req_num_conse , prq_num_conse )

go
exec sp_primarykey patida_requisicion ,
  alm_clave , req_num_conse , prq_num_conse
go
exec sp_foreignkey patida_requisicion , requisicion ,
  alm_clave , req_num_conse
go
exec sp_foreignkey patida_requisicion , compra_linea ,
  com_clave , lin_clave
go
exec sp_foreignkey patida_requisicion , linea_sublinea ,
  lin_clave , lin_sub_linea
go
exec sp_foreignkey patida_requisicion , unidad_medida ,
  ume_clave
go
exec sp_foreignkey patida_requisicion , material_corporativo ,
  mco_clave
go
exec sp_foreignkey patida_requisicion , unidad_medida ,
  ume_clave
go
exec sp_foreignkey patida_requisicion , material_unidad ,
  mun_clave
go

```

```

create table partida_requisicion_programa
(alm_clave      char(4)      not null, /* Clave de almacen PKEY */
 req_num_conse  int          not null, /* Numero reqen PKEY */
 prq_num_conse  smallint    not null, /* Num de partida PKEY */
 emb_num_conse  tinyint     not null, /* Numero de entrega PKY */
 emb_cantidad  float        not null, /* Cantidad esta entrega */
 emb_fec_entre  datetime    null) /* Fecha esta Entrega */
go
create unique clustered index xp_ptd req_pgm on partida_requisicion_programa
( alm_clave, req_num_conse, prq_num_conse, emb_num_conse )
go
exec sp_primarykey partida_requisicion_programa ,
      alm_clave, req_num_conse, prq_num_conse, emb_num_conse
go

drop trigger abpgmrq01_ins
go
drop trigger abpgmrq01_del
go
drop trigger abpgmrq01_upd
go

/* TRIGGER          PTDA_REQ_PGM ----INSERT          ***** */
create trigger abpgmrq01_ins on partida_requisicion_programa for insert
as
begin
  if @@rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */
  exec abvaluu_pro

  if not exists (select * from inserted ins, partida_requisicion prq
                where ins.alm_clave = prq.alm_clave and
                       ins.req_num_conse = prq.req_num_conse and
                       ins.prq_num_conse = prq.prq_num_conse)
    begin
      raiserror 20090 'Partida no existe'
      rollback tran
      return
    end
end
go

/* TRIGGER          PTDA_REQ_PGM ----DELETE          ***** */
create trigger abpgmrq01_del on partida_requisicion_programa for delete
as
begin
  if @@rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */
  exec abvaluu_pro
end
go

/* TRIGGER          PTDA_REQ_PGM ----UPDATE          ***** */
create trigger abpgmrq01_upd on partida_requisicion_programa for update
as
begin
  if @@rowcount = 0
    return

```

```

create table pedido
(alm_clave          char(4)      not null, /* Cve de almacen FKEY      PKEY */
 ped_clave         int          not null, /* Num consec.requis.      PKEY */
 ped_num_conse     tinyint     not null, /* Num. consec (anexo)     PKEY */
 prv_rfc           char(20)     not null, /* RFC del proveedor      */
 prv_rfc_repre    char(20)     null, /* RFC del representante  */
 pdo_num_conse    tinyint     not null, /* Llave de domicilio PRV */
 pdo_atencion     char(40)     not null, /* Persona que atiende al comp. */
 com_clave        char(2)      not null, /* Clave comprador        */
 ped_fec_elabo    datetime     not null, /* Fecha de elaboracion pedido */
 ped_fec_entre    datetime     null, /* Fecha prometida de embarque */
 ped_fec_ultima_repro datetime  null, /* Fecha de ul. reprogramacion */
 mon_clave        char(2)      not null, /* Tipo de moneda del monto */
 ped_imp_total    money         not null, /* Importe total neto      */
 ped_pje_imppto   float        not null, /* Porcentaje impuesto     */
 ped_pje_descto   float        not null, /* Porcentaje descuento   */
 prv_dto_pront_pago float      not null, /* Porcent. descto. pronto pago */
 prv_dia_pront_pago tinyint    null, /* Dias por pronto pago    */
 prv_cnd_pago     char(2)      null, /* Clave de condicion de pago */
 prv_dia_condi_pago tinyint    null, /* Dias de cond. de pago   */
 prv_cnd_entre    char(3)      null, /* Condicion de entrega    */
 prv_nom_lugar_entre char(35)  null, /* Clave de lugar de entrega */
 pdo_cve_domic_consi tinyint  null, /* Nombre lugar de entrega */
 ped_via_consi    char(2)      null, /* Clave de domicilio consigna. */
 ped_num_conve    smallint    null, /* Clave via de consignacion */
 ped_mat_origa    char(1)      null, /* Numero de convenio      */
 agt_clave        char(2)      null, /* Origen de la mercancia I o N */
 ped_tip_segur    char(2)      null, /* Clave de agente aduanal */
 ped_fec_autor    datetime     null, /* Clave de tipo de seguro */
 ped_per_autor    char(2)      null, /* Fecha de autorizacion   */
 ped_num_modif    tinyint     not null, /* Persona que autoriza    */
 ped_fec_modif    datetime     null, /* Numero de modificacion  */
 ped_imp_pagado   money         not null, /* Fecha ultima modificacion */
 ped_tpo_pedido  tinyint     not null, /* Importe pagado          */
 ped_comision     float        not null, /* Tipo de Pedido          */
 ped_status      char(2)      not null, /* Comision oficina internacion. */
 ped_fec_statu   datetime     null, /* Status del pedido       */
 pdo_cve_domic_factu tinyint  null, /* Clave de domicilio facturar. */
 cot_clave        char(14)     null, /* Clave de cotizacion     */
 ped_imp_modif    money         null, /* Clave de cotizacion     */
 ped_fec_autor_modif datetime  null, /* Monto Modificado       */
 /* Fecha de autorizacion de Modif*/
)

go
create unique nonclustered index xp_pedido on pedido
( alm_clave, ped_clave , ped_num_conse)

go
/* llaves involucradas */
exec sp_primarykey pedido ,
alm_clave, ped_clave , ped_num_conse

go
exec sp_foreignkey pedido , requisicion,
alm_clave, ped_clave , ped_num_conse

go
exec sp_foreignkey pedido , proveedor,
prv_rfc

go
exec sp_foreignkey pedido , proveedor,
prv_rfc_repre

go
exec sp_foreignkey pedido , comprador,
com_clave

go
exec sp_foreignkey pedido , moneda,
mon_clave

go
exec sp_foreignkey pedido , agente_aduanal,
agt_clave

go

```

```

create table pedido_modifica
(alm_clave char(4) not null, /* Cve de almacen FKKEY PKEY */
ped_clave int not null, /* Num consec.requis. PKEY */
ped_num_conse tinyint not null, /* Num. consec (anexo) PKEY */
prv_rfc char(20) not null, /* RFC del proveedor */
prv_rfc_repre char(20) null, /* RFC del representante */
pdo_num_conse tinyint not null, /* LLave de domicilio PRV */
pdo_atencion char(40) not null, /* Persona que atiende al comp. */
com_clave char(2) not null, /* Clave comprador */
ped_fec_elabo datetime not null, /* Fecha de elaboracion pedido */
ped_fec_entra datetime null, /* Fecha prometida de embarque */
ped_fec_ultima_repro datetime null, /* Fecha de ul. reprogramacion */
mon_clave char(2) not null, /* Tipo de moneda del monto */
ped_imp_total money not null, /* Importe total neto */
ped_pjc_impcto float not null, /* Porcentaje impuesto */
ped_pjc_dscto float null, /* Importe de descuento */
prv_dto_pront_pago float null, /* Percent. dscto. pronto pago */
prv_dia_pront_pago tinyint null, /* Dias por pronto pago */
prv_dia_pago char(2) null, /* Clave de condicion de pago */
prv_dia_condi_pago tinyint null, /* Dias de cond. de pago */
prv_cnd_entra char(3) null, /* Condicion de entrega */
prv_cve_lugar_entra char(2) null, /* Clave de lugar de entrega */
ped_nom_lugar_entra char(35) null, /* Nombre lugar de entrega */
pdo_cve_domic_consi tinyint null, /* Clave de domicilio consigna. */
ped_via_consi char(2) null, /* Clave via de consignacion */
ped_num_conve smallint null, /* Numero de convenio */
ped_mat_orige char(1) null, /* Origen de la mercancia I o N */
agt_clave char(2) null, /* Clave de agente aduanal */
ped_tip_segur char(2) null, /* Clave de tipo de seguro */
ped_fec_autor datetime null, /* Fecha de autorizacion */
ped_per_autor char(2) null, /* Persona que autoriza */
ped_num_modif tinyint not null, /* Numero de modificacion */
ped_fec_modif datetime not null, /* Fecha ultima modificacion */
ped_imp_pagado money not null, /* Importe pagado */
ped_tpo_pedid tinyint not null, /* Tipo de Pedido */
ped_comision float null, /* Comision oficina internac */
ped_status char(2) not null, /* Status del pedido */
ped_fec_statu datetime null, /* Fecha ultimo status */
pdo_cve_domic_factu tinyint null, /* Clave de domicilio facturar. */
cot_clave char(12) null, /* Clave de cotizacion */
ped_imp_modif money null, /* Monto Modificado */
ped_fec_autor_modif datetime null) /* Fecha de autorizacion de Modif*/

```

```

go
create unique nonclustered index xp_pedido_modifica on pedido_modifica
( alm_clave, ped_clave , ped_num_conse)
go
exec sp_primarykey pedido_modifica ,
alm_clave, ped_clave , ped_num_conse
go
exec sp_foreignkey pedido_modifica , requisicion,
alm_clave, ped_clave
go

```

```

create table proveedor
(prv_rfc          char(20)          not null, /*Tabla principal de proveedores*/
prv_rzo_socia   char(40)          not null, /*R.F.C del proveedor PKEY */
prv_nacionali   char(1)           not null, /*Razon social PKEY */
prv_mod_pago    char(4)           null, /*Tipo de proveedor N, I, L */
prv_via_notif   char(3)           null, /*Modalidad de Pago */
prv_dia_pago    tinyint           null, /*Via de notificacion */
prv_tpo_entre   smallint          null, /*Dias de pago */
prv_lug_entre   char(2)           null, /*Tiempo de entrega */
prv_cnd_entre   char(3)           null, /*Lugar de entrega */
prv_dto_gener   float             null, /*Condiciones de entrega */
prv_cnd_pago    char(2)           null, /*Descuento general */
prv_dia_condi_pago tinyint        null, /*Clave condicion de pago */
prv_rfc_repre   char(20)          null, /*Dias condiciones de pago */
prv_fec_vigen   datetime         not null, /*R.F.C. del Representante */
prv_fec_alta    fecha hoy         not null, /*Fecha limite vigencia */
prv_vigente     char(2)           null, /*Fecha de alta */
prv_ind_calidad char(2)           null, /*Proveedor vigente (SI/NO) */
prv_ind_precio  char(2)           null, /*Calidad */
prv_ind_servicio char(2)          null, /*Precio */
prv_dia_pront_pago tinyint       null, /*Servicio */
prv_dto_pront_pago float         null) /*Dias pronto pago */
/*Descuento por pronto pago */

go
create unique nonclustered index xp_proveedor on proveedor
( prv_rfc )
go
exec sp_primarykey proveedor ,
prv_rfc
go

drop trigger abinprv01_ins
go
drop trigger abinprv01_del
go
drop trigger abinprv01_upd
go

/* TRIGGER PROVEEDOR ----INSERT ***** */
create trigger abinprv01_ins on proveedor for insert
as
begin
declare @rowcount int
select @rowcount = @@rowcount

if @rowcount = 0
return

if @rowcount > 1
begin
raiserror 20040 'No se permite transaccion multiple'
rollback tran
return
end

/* Verifica que la transaction sea solicitada por medio del sistema
de Abastecimiento.
*/

exec abvaluu_pro

end
go

/* TRIGGER PROVEEDOR---DELETE ***** */
create trigger abinprv01_del on proveedor for delete
as
begin
declare @rowcount int

```

```

select @rowcount = @@rowcount
if @rowcount = 0
return
if @rowcount > 1
begin
raiserror 20040 'No se permite transaccion multiple'
rollback tran
return
end
/* Verifica que la transaction sea solicitada por medio del sistema
de Abastecimiento.
*/
exec abvalusu_pro
/* NO se permite delete de proveedores si tiene pedidos */
if exists (select * from deleted del, pedido ped
where del.prv_rfc = ped.prv_rfc)
begin
raiserror 20020 'Existen pedidos de este proveedor'
rollback tran
return
end
end
go
/* TRIGGER PROVEEDOR---UPDATE ***** */
create trigger abinprv01_upd on proveedor for update
as
begin
declare @rowcount int
select @rowcount = @@rowcount
if @rowcount = 0
return
if @rowcount > 1
begin
raiserror 20040 'No se permite transaccion multiple'
rollback tran
return
end
if update (prv_rfc)
begin
raiserror 20050 'No se permite cambio de rfc_proveedor'
rollback tran
return
end
/* Verifica que la transaction sea solicitada por medio del sistema
de Abastecimiento.
*/
exec abvalusu_pro
end
go

```

```

create table requisicion
(
  cia_clave          smallint      not null, /* Cve de la Compañía */
  uni_clave          char(4)        not null, /* Cve de la Unidad */
  alm_clave          char(4)        not null, /* Cve de almacén PKEY PKEY */
  req_num_conse      int             not null, /* Num consec.requis. PKEY */
  req_ord_traba      varchar(6)     null, /* Orden de trabajo */
  req_num equip      varchar(12)    null, /* Numero de equipo */
  req_aut_plant      char(2)        null, /* Planta (autoriz. desemb) */
  req_aut_folio      int            null, /* Folio (autoriz. desemb) */
  req_aut_perio      tinyint        null, /* Periodo (autoriz. desemb) */
  req_num_emple      int            null, /* Num empleado que autoriz */
  req_nom_solic      varchar(25)    null, /* Nombre de solicitante */
  req_tip_servi      char(1)        not null, /* Clave de tipo de servicio */
  req_cod_prior      char(1)        not null, /* Clave de prioridad */
  req_imp_estim      importe,       /* Monto estimado */
  req_folio          varchar(10)    null, /* Nímero de folio de requis */
  req_dep_nombr      varchar(20)    null, /* Nombre depto solicitante */
  req_act_fijo       char(1)        not null, /* Indicador de activo fijo */
  req_num_ptdas      smallint       not null, /* Numero de partidas */
  req_num_coloc      smallint       null, /* partidas colocadas */
  req_num_pedid      smallint       null, /* pedidos colocados */
  req_fec_elabo      datetime       not null, /* Fecha de Elaborac. requi */
  req_fec_recep      datetime       not null, /* Fecha de recepcion requi */
  req_fec_reque      datetime       not null, /* Fecha requerida en unidad */
  req_fec_asign      fecha hoy      not null, /* Fecha asignacion compradr */
  req_cve_statu      char(2)        not null, /* Clave de estatus actual */
  req_fec_statu      datetime       not null, /* Fecha de estatus actual */
  req_fec_alta      fecha hoy,      /* Fecha de alta de registro */
  req_des_gener      varchar(255)   null, /* Descripción Generica */
  alm_cta_cuent      smallint       null, /* Cuenta de cargo -Mayor- */
  alm_cta_subct      tinyint        null, /* Subcuenta */
  alm_cta_ssubc      tinyint        null, /* Subsubcuenta */
  alm_cta_conco      smallint       null, /* Concepto */
  mon_clave          char(2)        null, /* Tipo de moneda del monet */
  lin_clave          tinyint        not null, /* Clave de la línea */
  com_clave          char(2)        not null, /* Clave comprador */
  req_fec_coloc     datetime       null) /* Fecha de colocacion */

go
create unique nonclustered index xp_requisicion on requisicion
( alm_clave , req_num_conse )

go
exec sp_primarykey requisicion ,
  alm_clave , req_num_conse

go
exec sp_foreignkey requisicion , almacen ,
  alm_clave

go
exec sp_foreignkey requisicion , comprador ,
  com_clave

go
exec sp_foreignkey requisicion , moneda ,
  mon_clave

go
exec sp_bindefault tipo_serv, 'requisicion.req_tip_servi'
go
exec sp_bindefault prioridad, 'requisicion.req_cod_prior'
go
exec sp_bindefault act_fij, 'requisicion.req_act_fijo'
go
exec sp_bindrule req_tip_servi, 'requisicion.req_tip_servi'
go
exec sp_bindrule req_cod_prior, 'requisicion.req_cod_prior'
go
exec sp_bindrule req_act_fijo, 'requisicion.req_act_fijo'
go

```

```

create table requisicion transf
(
  cia_clave          smallint      null, /* Cve de la Compañia */
  uni_clave          char(4)        null, /* Cve de la Unidad */
  alm_clave          char(4)        null, /* Cve de almacenes PKEY PKEY */
  req_num_conse      int            null, /* Num consec.requis. PKEY */
  req_ord_traba      varchar(6)     null, /* Orden de trabajo */
  req_num equip      varchar(12)    null, /* Numero de equipo */
  req_aut_plant      char(2)        null, /* Planta (autoriz. desem) */
  req_aut_folio      int            null, /* Folio (autoriz. desem) */
  req_aut_perio      tinyint        null, /* Periodo (autoriz. desem) */
  req_num_emple      int            null, /* Num empleado que autoriz */
  req_nom_solic      varchar(25)    null, /* Nombre de solicitante */
  req_tip_servi      char(1)        null, /* Clave de tipo de servi */
  req_cod_prior      char(1)        null, /* Clave de prioridad */
  req_imp_estim      importe        null, /* Monto estimado */
  req_folio          varchar(10)     null, /* Numero de folio de requis */
  req_dep_nombr      varchar(20)     null, /* Nombre depto solicitante */
  req_act_fijo       char(1)        null, /* Indcador de activo fijo */
  req_num_ptdas      smallint       null, /* Numero de partidas */
  req_num_coloc      smallint       null, /* partidas colocadas */
  req_num_pedid      smallint       null, /* pedidos colocados */
  req_fec_elabo      datetime        null, /* Fecha de Elaborac. requ */
  req_fec_recep      datetime        null, /* Fecha de recepcion requi */
  req_fec_reque      datetime        null, /* Fecha requerida en unidad */
  req_fec_asign      fecha hoy       null, /* Fecha asignacion compradr */
  req_cve_statu      char(2)        null, /* Clave de estatus actual */
  req_fec_statu      datetime        null, /* Fecha de estatus actual */
  req_fec_alta      fecha hoy        null, /* Fecha de alta de registro */
  req_des_gener      varchar(255)    null, /* Descripcion Generica */
  alm_cta_cuent      smallint       null, /* Cuenta de cargo -Mayor- */
  alm_cta_subct      tinyint        null, /* Subcuenta */
  alm_cta_ssubc      tinyint        null, /* Subsubcuenta */
  alm_cta_conce      smallint       null, /* Concepto */
  mon_clave         char(2)         null, /* Tipo de moneda del monet */
  lin_clave         tinyint        null, /* Clave de la linea */
  com_clave         char(2)         null, /* Clave comprador */
  req_fec_coloc     datetime        null) /* Fecha de colocacion */

```

go

```

create table tipo_cambio
( mon_clave char(2) not null, /* Clave de moneda */
  tca_fec_cotiz datetime not null, /* Fecha de cotizacion */
  tca_tip_cambi float not null) /* Tipo de cambio */
go
create unique clustered index xp_tipo_cambio on tipo_cambio
( mon_clave , tca_fec_cotiz )
go
exec sp_primarykey tipo_cambio,
mon_clave , tca_fec_cotiz
go
exec sp_foreignkey tipo_cambio , moneda ,
mon_clave
go

drop trigger abintca01_ins
go
drop trigger abintca01_del
go
drop trigger abintca01_upd
go

/* TRIGGER TIPO CAMBIO ----INSERT ***** */
create trigger abintca01_ins on tipo_cambio for insert
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount

  if @rowcount = 0
    return

  /* Verifica que la transaction sea solicitada por medio del sistema
  * de Abastecimiento.
  */
  exec abvalusu_pro

  /* verifica existencia de clave de moneda */
  if not exists(select * from inserted ins, moneda mon
               where ins.mon_clave = mon.mon_clave)
  begin
    raiserror 20010 'No existe clave de moneda'
    rollback tran
    return
  end
end
go

/* TRIGGER TIPO CAMBIO---DELETE ***** */
create trigger abintca01_del on tipo_cambio for delete
as
begin
  raiserror 20030 'No se permite eliminacion de tipo_cambio'
  rollback tran
  return
end
go

/* TRIGGER TIPO CAMBIO---UPDATE ***** */
create trigger abintca01_upd on tipo_cambio for update
as
begin
  declare @rowcount int
  select @rowcount = @@rowcount

  if @rowcount = 0

```

```

create table unidad
( cia_clave      smallint      not null, /* Cve compania llave foranea */
  uni_clave      char(4)        not null, /* Clave de Unidad PKEY */
  uni_nombre     varchar(40)    not null, /* Nombre de unidad */
  uni_fec_alta   fecha_hoy)     /* Fecha de alta de registro */
go
create unique clustered index xp_unidad on unidad
( cia_clave , uni_clave )
go
exec sp_primarykey unidad ,
    cia_clave , uni_clave
go
exec sp_foreignkey unidad , compania ,
    cia_clave
go
drop trigger abinuni01_ins
go
drop trigger abinuni01_del
go
drop trigger abinuni01_upd
go

/* TRIGGER          UNIDAD ----INSERT      ***** */
create trigger abinuni01_ins on unidad for insert
as
begin
    if @@rowcount = 0
        return

    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    */

    exec abvaluu_pro
    if not exists(select * from inserted ins, compania cia
                  where ins.cia_clave = cia.cia_clave)
        begin
            raiserror 20110 'No existe compania'
            rollback tran
            return
        end
end
go

/* TRIGGER          UNIDAD---DELETE      ***** */
create trigger abinuni01_del on unidad for delete
as
begin
    if @@rowcount = 0
        return

    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    */

    exec abvaluu_pro

    /* Si la unidad tiene almacenes vigentes, rollback tran */

    if exists (select * from deleted de, almacen alm
              where de.uni_clave = alm.uni_clave)
        begin
            raiserror 20020 'Existen almacenes de esta unidad'
            rollback tran
            return
        end
end
go

```

```

create table unidad_medida
( ume_clave      char(3)      not null, /* Clave unidad de medida PKEY */
  ume_unimed     char(5)      not null, /* Unidad de medida */
  ume_clave_ing  char(3)      not null, /* Clave unidad de medida ing */
  ume_unimed_ing char(5)      not null, /* Unidad de medida ingles */
  ume_nombre     varchar(20)  not null) /* Descripcion de la unidad m. */
go

create unique clustered index xp_unidad_medida on unidad_medida
( ume_clave )
go

exec sp_primarykey unidad_medida,
    ume_clave
go

drop trigger abinume01_ins
go

drop trigger abinume01_deu
go

/* TRIGGER          UNIDAD_MEDIDA ----INSERT      ***** */
create trigger abinume01_Ins on unidad_medida for insert
as
begin
    if @@rowcount = 0
        return

    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    */

    exec abvalu01_pro
end
go

/* TRIGGER          UNIDAD_MEDIDA---DELETE/UPD      ***** */
create trigger abinume01_deu on unidad_medida for delete, update
as
begin
    if @@rowcount = 0
        return

    /* NO se permite modificacion ni delete de unidades de medida */
    raiserror 20020 'Transaccion no autorizada'
    rollback tran
    return
end
go

```

APENDICE B.

GRANTS, PROCEDIMIENTOS - ALMACENADOS Y TRIGGERS

NOTA; En este apéndice sólo se incluye una parte de la información referente al mismo, debido a lo extenso del sistema.

```
grant select on division
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on compania
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on unidad
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on almacan
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on ofna regional
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on ofnareg at compania
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on comprador
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on linea sublinea
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select, insert, update, delete on compra linea
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on material unidad
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on material corporativo
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on matun matco
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on unidad medida
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant select on requisicion
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
grant update on requisicion (req num coloc,req num pedid,req cve statu,req fec stat
to jan5842,jcp6102,afs0736,ghh2099,npo0920,fb2996,pzr7727,lam1619,mpo10147,lth3974
go
grant update on requisicion
to dbc1063,omb2763,msj3185
go
grant select on partida requisicion
to jan5842,jcp6102,afs0736,ghh2099,npo0920,fb2996,pzr7727,lam1619,mpo10147,lth3974
go
grant update on partida requisicion (prq can coloc,prq cod estat,prq fec estat)
to jan5842,jcp6102,afs0736,ghh2099,npo0920,fb2996,pzr7727,lam1619,mpo10147,lth3974
go
grant select, insert, update on partida_requisicion
to dbc1063,omb2763,msj3185
go
grant select on partida requisicion programa
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fb2996,msj3185,pzr7727,
go
```

grant select, insert, update, delete on tabla autori
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on descr temporal
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on descr temporal modifica
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select on moneda
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select on descripcion
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select on requisicion campo
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on solicitud cotizacion
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on solicitud transporte
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on solicitud transporte pedido
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on proveedor
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on atencion proveedor
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on banco proveedor
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on telefono proveedor
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on cotizacion
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on partida cotizacion
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on solicitud partida requisicion
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on solicitud proveedor
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on domicilio proveedor
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on representacion proveedor
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fab2996,msj3185,pzr7727,
go

grant select, insert, update, delete on proveedor_linea_sublinea
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on pedido
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on plano_pedido
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on modificacion
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on partida_pedido
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on embarque
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on requisicion_cotizacion
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on notas_requisicion
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on notas_pedido
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on notas_catalogo
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select on tabla
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select on tabla_detalle
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select on agente_aduanal
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on factura_trafico_pedido
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on factura_trafico
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on bulto
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select on trafico
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on status_trafico
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on modalidad_status
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on movimiento_material
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on movimiento_partida_pedido
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on bulto
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on gastos reales
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on pedido modifica
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on partida_pedido modifica
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on domicilio temporal
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select on almacen domicilio
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select on domicilio
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on embarque modifica
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on domicilio temporal modifica
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on notas_pedido modifica
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on status
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select on moneda
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select on tipo cambio
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on proveedor convenio
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on lista precio
to jan5842,dbc1063,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,
go

grant select, insert, update, delete on historico
to dbc1063
go

grant select, insert, update, delete on historico_pedido
to dbc1063
go

grant select on historico
to jan5842,jcp6102,afs0736,ghh2099,omb2763,npo0920,fsb2996,maj3185,pzr7727,lam1619,
go

```

grant select on historico_pedido
to jan5842,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,lam1619,
go
grant select on his_req_mensual
to jan5842,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,lam1619,
go
grant select on his_ped_mensual
to jan5842,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,lam1619,
go
grant select on impresora
to jan5842,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,lam1619,
go
grant select on tipo_impresora
to jan5842,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,lam1619,
go
grant select on reporte
to jan5842,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,lam1619,
go
grant exec on abvalusu_pro
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on abreqin01_isq
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on abreqin01_dsq
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on abreqin01_dsq
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbalmac
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbcomlin
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbcompra
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbcomsup
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbdescri
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbdestem
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbmatco
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbmatun
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbmatunco
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbreq52
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbreqcam
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbreqnot
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbreqnum
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,
go
grant exec on rbreqprg

```

```
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,  
go  
grant exec on rbreqtm01  
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,  
go
```

```
Grant : grant.dom
```

```
grant select on domicilio  
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,  
go
```

```
grant select on almacen domicilio  
to jan5842,dbc1063,jcp6102,afa0736,ghh2099,omb2763,npo0920,feb2996,maj3185,pzr7727,  
go
```

```
grant select, insert, update, delete on division
to sistemas,dbc1063
go
grant select, insert, update, delete on compania
to sistemas,dbc1063
go
grant select, insert, update, delete on unidad
to sistemas,dbc1063
go
grant select, insert, update, delete on almacen
to sistemas,dbc1063
go
grant select, insert, update, delete on ofna_regional
to sistemas,dbc1063
go
grant select, insert, update, delete on ofnareg_at_compania
to sistemas,dbc1063
go
grant select, insert, update, delete on comprador
to sistemas,dbc1063
go
grant select, insert, update, delete on linea_sublinea
to sistemas,dbc1063
go
grant select, insert, update, delete on compra_linea
to sistemas,dbc1063
go
grant select, insert, update, delete on material_unidad
to sistemas,dbc1063
go
grant select, insert, update, delete on material_corporativo
to sistemas,dbc1063
go
grant select, insert, update, delete on matun_matco
to sistemas,dbc1063
go
grant select, insert, update, delete on unidad_medida
to sistemas,dbc1063
go
grant select, insert, update, delete on requisicion
to sistemas,dbc1063
go
grant select, insert, update, delete on partida_requisicion
to sistemas,dbc1063
go
grant select, insert, update, delete on partida_requisicion_programa
to sistemas,dbc1063
go
grant select, insert, update, delete on tabla_autori
to sistemas,dbc1063
go
grant select, insert, update, delete on descr_temporal
to sistemas,dbc1063
go
grant select, insert, update, delete on descr_temporal_modifica
to sistemas,dbc1063
go
grant select, insert, update, delete on moneda
to sistemas,dbc1063
go
grant select, insert, update, delete on descripcion
to sistemas,dbc1063
go
grant select, insert, update, delete on requisicion_campo
to sistemas,dbc1063
go
grant select, insert, update, delete on solicitud_cotizacion
to sistemas,dbc1063
```

```

go
grant select, insert, update, delete on proveedor
to sistemas,dbc1063
go
grant select, insert, update, delete on cotizacion
to sistemas,dbc1063
go
grant select, insert, update, delete on partida_cotizacion
to sistemas,dbc1063
go
grant select, insert, update, delete on solicitud_partida_requisicion
to sistemas,dbc1063
go
grant select, insert, update, delete on solicitud_proveedor
to sistemas,dbc1063
go
grant select, insert, update, delete on domicilio_proveedor
to sistemas,dbc1063
go
grant select, insert, update, delete on representa_proveedor
to sistemas,dbc1063
go
grant select, insert, update, delete on proveedor_linea_sublinea
to sistemas,dbc1063
go
grant select, insert, update, delete on pedido
to sistemas,dbc1063
go
grant select, insert, update, delete on partida_pedido
to sistemas,dbc1063
go
grant select, insert, update, delete on embarque
to sistemas,dbc1063
go
grant select, insert, update, delete on requisicion_cotizacion
to sistemas,dbc1063
go
grant select, insert, update, delete on notas_requisicion
to sistemas,dbc1063
go
grant select, insert, update, delete on notas_pedido
to sistemas,dbc1063
go
grant select, insert, update, delete on notas_catalogo
to sistemas,dbc1063
go
grant select, insert, update, delete on tabla
to sistemas,dbc1063
go
grant select, insert, update, delete on tabla_detalle
to sistemas,dbc1063
go
grant select, insert, update, delete on agente_aduanal
to sistemas,dbc1063
go
grant select, insert, update, delete on factura_trafico_pedido
to sistemas,dbc1063
go
grant select, insert, update, delete on factura_trafico
to sistemas,dbc1063
go
grant select, insert, update, delete on bulto
to sistemas,dbc1063
go
grant select, insert, update, delete on trafico
to sistemas,dbc1063
go
grant select, insert, update, delete on status_trafico
to sistemas,dbc1063
go
grant select, insert, update, delete on modalidad_status
to sistemas,dbc1063
go
grant select, insert, update, delete on movimiento_material
to sistemas,dbc1063
go

```

```

grant select, insert, update, delete on movimiento_partida_pedido
to sistemas,dbc1063
go
grant select, insert, update, delete on bulto
to sistemas,dbc1063
go
grant select, insert, update, delete on gastos_reales
to sistemas,dbc1063
go
grant select, insert, update, delete on pedido_modifica
to sistemas,dbc1063
go
grant select, insert, update, delete on partida_pedido_modifica
to sistemas,dbc1063
go
grant select, insert, update, delete on domicilio_temporal
to sistemas,dbc1063
go
grant select, insert, update, delete on domicilio
to sistemas,dbc1063
go
grant select, insert, update, delete on almacen_domicilio
to sistemas,dbc1063
go
grant select, insert, update, delete on embarque_modifica
to sistemas,dbc1063
go
grant select, insert, update, delete on domicilio_temporal_modifica
to sistemas,dbc1063
go
grant select, insert, update, delete on notas_pedido_modifica
to sistemas,dbc1063
go
grant select, insert, update, delete on status
to sistemas,dbc1063
go
grant select, insert, update, delete on moneda
to sistemas,dbc1063
go
grant select, insert, update, delete on tipo_cambio
to sistemas,dbc1063
go
grant select, insert, update, delete on proveedor_convenio
to sistemas,dbc1063
go
grant select, insert, update, delete on status
to sistemas,dbc1063
go
grant select, insert, update, delete on lista_precio
to sistemas,dbc1063
go
grant select, insert, update, delete on historico
to sistemas,dbc1063
go
grant select, insert, update, delete on historico_pedido
to sistemas,dbc1063
go
grant select, insert, update, delete on his_req_mensual
to sistemas,dbc1063
go
grant select, insert, update, delete on his_ped_mensual
to sistemas,dbc1063
go
grant select, insert, update, delete on banco_proveedor
to sistemas,dbc1063
go
grant select, insert, update, delete on telefono_proveedor
to sistemas,dbc1063
go
grant select, insert, update, delete on atencion_proveedor
to sistemas,dbc1063
go
grant select, insert, update, delete on modificacion
to sistemas,dbc1063
go
grant select on situacion_trafico

```

```
to sistemas,dbc1063
go
grant exec on abvalusu_pro
to sistemas,dbc1063
go
grant exec on abreqin01_isq
to sistemas,dbc1063
go
grant exec on abreqin01_desq
to sistemas,dbc1063
go
grant exec on rbalmac
to sistemas,dbc1063
go
grant exec on rbcomlin
to sistemas,dbc1063
go
grant exec on rbcompra
to sistemas,dbc1063
go
grant exec on rbcomsup
to sistemas,dbc1063
go
grant exec on rbdescr1
to sistemas,dbc1063
go
grant exec on rbdestem
to sistemas,dbc1063
go
grant exec on rbmatco
to sistemas,dbc1063
go
grant exec on rbmatun
to sistemas,dbc1063
go
grant exec on rbmatunco
to sistemas,dbc1063
go
grant exec on rbreq52
to sistemas,dbc1063
go
grant exec on rbreqcam
to sistemas,dbc1063
go
grant exec on rbreqnot
to sistemas,dbc1063
go
grant exec on rbreqnum
to sistemas,dbc1063
go
grant exec on rbreqprg
to sistemas,dbc1063
go
grant exec on rbreqtm01
to sistemas,dbc1063
go
```

```

/* requisicion.trg.ins */
drop trigger abreqin01_ins
go

/* En la tabla de requisicion, no deben existir requisiciones duplicadas
 * en almacen y numero de requisicion. Esto es evitado solo mediante la
 * asignacion de un INDEX (unique, nonclustered) que siempre debe estar
 * activo cuando hay acceso de usuarios a la base de datos.
 */

/* TRIGGER          REQUISICION---INSERT          ***** */
create trigger abreqin01_ins on requisicion for insert
as
begin
    if @@rowcount = 0
        return

    if @@rowcount > 1
        begin
            raiserror 20120 'No se permite insercion multiple'
            rollback tran
            return
        end

    /* Verifica que la transaction sea solicitada por medio del sistema
     * de Abastecimiento.

    exec abvaluu_pro */

    /* valida que la requisicion no exista

    if exists(select * from inserted ins, requisicion req
              where ins.alm_clave = req.alm_clave
                and ins.req_num_conse = req.req_num_conse )
    begin
        raiserror 20100 'La requisición ya existe'
        rollback tran
        return
    end

    */
    /* valida existencia de almacen */

    if not exists(select * from inserted ins, almacen al
                  where al.alm_clave = ins.alm_clave )
    begin
        raiserror 20100 'Inconsistencia en almacen'
        rollback tran
        return
    end

    /* valida que exista la linea */

    if not exists(select * from inserted ins, linea_sublinea lin
                  where lin.lin_clave = ins.lin_clave )
    begin
        raiserror 20100 'Inconsistencia en la Linea'
        rollback tran
        return
    end

    /* valida que exista el comprador */

    if not exists(select * from inserted ins, comprador com
                  where com.com_clave = ins.com_clave )
    begin
        raiserror 20100 'Inconsistencia en el Comprador'
        rollback tran
        return
    end

end
go

```

```
/* requisicion.trg.upd */
drop trigger abrequin01_upd
go
/* TRIGGER REQUISICION---UPDATE ***** */
create trigger abrequin01_upd on requisicion for update
as
begin
    if @@rowcount = 0
        begin
            raiserror 20110 'No se encontro registro'
            rollback tran
            return
        end
    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    exec abvaluu_pro
    if @@rowcount > 1
        begin
            raiserror 20120 'No se permite update multiple'
            rollback tran
            return
        end
    */
    /* No se permite cambio de almacen(centro) ni numero de requisicion. */
    if (update (alm_clave) or update (req_num_conse))
        begin
            raiserror 20050 'No se permite cambio de Centro o Numero de req.'
            rollback tran
            return
        end
    /* valida que exista la linea */
    if not exists(select * from inserted ins, linea sublinea lin
        where lin.lin_clave = ins.lin_clave )
        begin
            raiserror 20100 'Inconsistencia en la Linea'
            rollback tran
            return
        end
    /* valida que exista el comprador */
    if not exists(select * from inserted ins, comprador com
        where com.com_clave = ins.com_clave )
        begin
            raiserror 20100 'Inconsistencia en el Comprador'
            rollback tran
            return
        end
    end
go
```

```

/* requisicion.trg.del */
drop trigger abreqin01_del
go
/* TRIGGER      REQUISICION---DELETE      ***** */
create trigger abreqin01_del on requisicion for delete
as
begin
    if @@rowcount = 0
        return
    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    */
    exec abvaluu_pro
    /* Si procede la baja de requisicion, delete las partidas de esta *
    * requisicion. (en trigger de baja de partidas se incluye la baja *
    * de descr_temporal si la partida indica tipo de descr. = 3).
    * y tambien dar de baja las notas de requisicion existentes.
    */
    /* Valida que la requisicion no tenga solicitud de cotizacion */
    /* if exists (select *
    from deleted de, solicitud_partida_requisicion sp
    where de.alm_clave = sp.alm_clave
    and de.req_num_conse = sp.req_num_conse )
    begin
        raiserror 20010 'La Requisicion Tiene Cotizaciones Colocadas'
        rollback tran
        return
    end
    /* Valida que la requisicion no tenga cotizaciones colocadas */
    /* if exists ( select * from deleted de, partida_cotizacion pc
    where de.alm_clave = pc.alm_clave
    and de.req_num_conse = pc.req_num_conse )
    begin
        raiserror 20010 'La Requisicion Tiene Cotizaciones Colocadas'
        rollback tran
        return
    end
    /* Valida que la requisicion no tenga pedidos colocados */
    if exists ( select * from deleted de, pedido pp
    where de.alm_clave = pp.alm_clave
    and de.req_num_conse = pp.ped_clave )
    begin
        raiserror 20010 'La Requisicion Tiene Pedidos Colocados'
        rollback tran
        return
    end
    /* Da de baja las partidas de la requisicion */
    delete partida_requisicion from partida_requisicion pr, deleted de
    where pr.alm_clave = de.alm_clave and
    pr.req_num_conse = de.req_num_conse
    /* Da de baja las notas de la requisicion */
    delete notas_requisicion from notas_requisicion nr, deleted de
    where nr.alm_clave = de.alm_clave and
    nr.req_num_conse = de.req_num_conse
end
go

```

```

drop trigger abpedin01_ins
go
/* TRIGGER          PEDIDO---INSERT          ***** */
create trigger abpedin01_ins on pedido for insert
as
begin
    declare @rowcount int
    select @rowcount = @@rowcount
    if @rowcount = 0
        return
    if @rowcount > 1
    begin
        raiserror 20100 'No es permitido hacer inserciones multiples'
        rollback tran
        return
    end
    /* Verifica que la transaction sea solicitada por medio del sistema
    * de Abastecimiento.
    * exec abvaluu_pro
    */
    /* valida existencia de requisicion */
    if not exists(select * from inserted ins, requisicion req
        where ins.alm_clave = req.alm_clave and
              ins.ped_clave = req.req_num_conse and
              ins.com_clave = req.com_clave and
              ins.ped_status = 'PA' )
    begin
        raiserror 20100 'Inconsistencia en requisicion o no es el mismo comprador'
        rollback tran
        return
    end
    /* valida existencia del proveedor */
    if not exists(select * from inserted ins, proveedor prv
        where ins.prv_rfc = prv.prv_rfc )
    begin
        raiserror 20100
        'Inconsistencia en proveedor'
        rollback tran
        return
    end
    /* valida existencia de moneda */
    if not exists(select * from inserted ins, moneda mon
        where ins.mon_clave = mon.mon_clave )
    begin
        raiserror 20100
        'Inconsistencia en moneda'
        rollback tran
        return
    end
    /* valida existencia del rep_proveedor */
    if exists(select * from inserted ins
        where ins.prv_rfc != ins.prv_rfc_repre and
              ins.prv_rfc_repre is not null )

```

```
if not exists(select * from inserted ins , proveedor pr
              where ins.prv_rfc_repre      = pr.prv_rfc)
begin
raiserror 20100 'Inconsistencia del representante proveedor. '
rollback tran
return
end
```

```
/* valida existencia del agt aduanal */
```

```
if exists(select * from inserted ins
          where ins.agt_clave is not null )
if not exists(select * from inserted ins , agente_aduanal ad
              where ins.agt_clave = ad.agt_clave )
begin
raiserror 20100 'Inconsistencia del Agente aduanal. '
rollback tran
return
end
```

```
end
go
```

```

drop trigger abpedin01_upd
go
/* TRIGGER          PEDIDO---UPDATE          ***** */
create trigger abpedin01_upd on pedido for update
as
begin
    declare @rowcount          int
           , @importe_pedido float

    select @rowcount = @@rowcount
    if @rowcount = 0
        return

/*
    if @rowcount > 1
    begin
        raiserror 20100 'No es permitido hacer updates multiples'
        rollback tran
        return
    end */

/* Verifica que la transaction sea solicitada por medio del sistema
   * de Abastecimiento.
   * exec abvalusu_pro
   */

    if (update (alm_clave) or update (ped_clave) or
        update (ped_num_conse) /* or update (com_clave) */)
    begin
        raiserror 20050
        'No se permiten cambios de Centro, Num. de ped. Importe o comprador'
        rollback tran
        return
    end

    if update (prv_rfc)
    if not exists(select * from inserted ins, proveedor prv
                 where ins.prv_rfc = prv.prv_rfc )
    begin
        raiserror 20100 'Inconsistencia de proveedor '
        rollback tran
        return
    end

/* valida existencia de proveedor y moneda */

    if update (mon_clave)
    if not exists(select * from inserted ins, moneda mon
                 where ins.mon_clave = mon.mon_clave )
    begin
        raiserror 20100 'Inconsistencia en moneda'
        rollback tran
        return
    end

    if update (prv_rfc_repre)
    if exists(select * from inserted ins
              where ins.prv_rfc != ins.prv_rfc_repre
                and ins.prv_rfc_repre is not null )
    if not exists(select * from inserted ins , proveedor pr
                  where ins.prv_rfc_repre = pr.prv_rfc)
    begin
        raiserror 20100 'Inconsistencia del representante proveedor. '
        rollback tran
        return
    end
end

```

```

if update (agt_clave)
  if exists(select * from inserted ins
            where ins.agt_clave is not null )
    if not exists(select * From inserted ins , agente_aduana ad
                  where ins.agt_clave = ad.agt_clave )
      begin
        raiserror 20100 'Inconsistencia del Agente aduana. '
        rollback tran
        return
      end

    /* limitar el a status validos*/

if update (ped_status)
  if not exists (select * from inserted ins, status s
                where s.sta_clave = ins.ped_status
                  and s.sta_tipo = 'P')
    begin
      raiserror 20100 'Status invalido del pedido. '
      rollback tran
      return
    end

    /* cuando se cancela una pedido se recalcula Monto */

if update (ped_imp_total) and update (ped_status)
  if exists ( select * from inserted ins
            where ins.ped_status = "PC"
              and ins.ped_imp_total = 0 )
    update pedido set
      ped_imp_total = (select sum(ppe.cantidad * ppe.pre_unita
                                * (1 - ppe.pt_dcto/100)
                        from inserted ins , partida_pedido ppe
                        where ppe.alm_clave = ins.alm_clave
                          and ppe.ped_clave = ins.ped_clave
                          and ppe.ped_num_conse = ins.ped_num_conse )
                    from inserted ins ,pedido ped
                    where ped.alm_clave = ins.alm_clave
                      and ped.ped_clave = ins.ped_clave
                      and ped.ped_num_conse = ins.ped_num_conse

end
go

```

```

drop trigger abpedin01_del
go
/* TRIGGER          PEDIDO---DELETE          ***** */
create trigger abpedin01_del on pedido for delete
as
begin
    if @@rowcount = 0
        return
/* Verifica que la transaction sea solicitada por medio del sistema
* de Abastecimiento.
* exec abvaluu_pro
*/
/* Si procede la baja de pedido, delete las partidas de esta *
* pedido. (en trigger de baja de partidas se incluye la baja *
* de descr temporal si la partida indica tipo de descr. = 3).
* y tambien dar de baja las notas de pedido existentes.
*/
if exists (select * from deleted de, movimiento_partida_pedido mp
           where de.alm_clave = mp.alm_clave and
                 de.ped_clave = mp.ped_clave and
                 de.ped_num_conse = mp.ped_num_conse )
    begin
        raiserror 20030 'Alguna Partida tiene Mov., se rechaza eliminacion'
        rollback tran
        return
    end

delete partida_pedido from partida_pedido pr, deleted de
    where pr.alm_clave = de.alm_clave and
          pr.ped_clave = de.ped_clave and
          pr.ped_num_conse = de.ped_num_conse

delete notas_pedido from notas_pedido nr, deleted de
    where nr.alm_clave = de.alm_clave and
          nr.ped_clave = de.ped_clave and
          nr.ped_num_conse = de.ped_num_conse

delete domicilio_temporal from domicilio_temporal dt, deleted de
    where dt.alm_clave = de.alm_clave and
          dt.ped_clave = de.ped_clave and
          dt.ped_num_conse = de.ped_num_conse

delete pedido_modifica from pedido_modifica pr, deleted de
    where pr.alm_clave = de.alm_clave and
          pr.ped_clave = de.ped_clave and
          pr.ped_num_conse = de.ped_num_conse

delete partida_pedido_modifica from partida_pedido_modifica pr, deleted de
    where pr.alm_clave = de.alm_clave and
          pr.ped_clave = de.ped_clave and
          pr.ped_num_conse = de.ped_num_conse

delete embarque_modifica from embarque_modifica pr, deleted de
    where pr.alm_clave = de.alm_clave and
          pr.ped_clave = de.ped_clave and
          pr.ped_num_conse = de.ped_num_conse

delete notas_pedido_modifica from notas_pedido_modifica nr, deleted de
    where nr.alm_clave = de.alm_clave and
          nr.ped_clave = de.ped_clave and

```

```
nr.ped_num_conse = de.ped_num_conse
delete domicilio_temporal_modifica
from domicilio_temporal_modifica dt, deleted de
where dt.alm_clave = de.alm_clave and
dt.ped_clave = de.ped_clave and
dt.ped_num_conse = de.ped_num_conse

delete modificacion
from modificacion dt, deleted de
where dt.alm_clave = de.alm_clave and
dt.ped_clave = de.ped_clave and
dt.ped_num_conse = de.ped_num_conse

end
go
```

```

/* TRIGGER          EMBAQUE---INSERT ***** */
drop trigger      abembin01_ins
go

create trigger abembin01_ins on embarque for insert
as
begin
declare @rowcount int,
        @almacen char(4),
        @n_pedido int,
        @añexo tinyint,
        @n_partida smallint,
        @sum_emb float,
        @can_par float,
        @importe_t float,
        @importe_p float

select @rowcount = @@rowcount

if @rowcount = 0
return

/* Verifica que la transaction sea solicitada por medio del sistema
de Abastecimiento.
exec abvaluu_pro
*/
/* valida que solo se cargen embarques a una partida a al vez */

/* if @rowcount > 1
begin
select alm_clave, ped_clave, ped_num_conse, ppe_num_conse from inserted
group by alm_clave, ped_clave, ped_num_conse, ppe_num_conse
if 1 != @@rowcount
begin
raiserror 20100 'No es valido cargar embarques a mas de una partida'
rollback tran
return
end
end
*/
/* verifica existencia de llaves foraneas */

if @rowcount != (select count(*) from inserted ins , partida_pedido pe
where ins.alm_clave = pe.alm_clave and
ins.ped_clave = pe.ped_clave and
ins.ped_num_conse = pe.ped_num_conse and
ins.ppe_num_conse = pe.ppe_num_conse )/* and
ins.emb_cv_e_cantidad > 0 and
pe.ppe_cv_e_statu != 'PC' and
ins.emb_cv_e_statu = 'PA' */
begin
raiserror 20100 'Falla de integridad en llaves foraneas'
rollback tran
return
end

/* la validacion de cantidades solo se hace cuando se atoziza
select @can_par = ppe cantidad from partida_pedido
where alm_clave = @almacen
and ped_clave = @n_pedido
and ped_num_conse = @añexo
and ppe_num_conse = @n_partida

select @sum_emb = sum (emb cantidad) from embarque
where alm_clave = @almacen

```

```

                and ped_clave      = @n_pedido
                and ped_num_conse = @anexo
                and ppe_num_conse = @n_partida
                and ppe_cve_statu i= "PC"
if @sum_emb i= @can_par
begin
    raisererror 20100
    'La cantidad de la partida no es igual a la de los embarques'
    rollback tran
    return
end
*/
select @almacen = alm_clave, @n_pedido = ped_clave , @anexo = ped_num_conse
, @n_partida = ppe_num_conse from inserted

update partida_pedido set
    ppe_cve_statu = "PA"
, ppe_cve_progr = "S"
    where alm_clave      = @almacen
    and ped_clave       = @n_pedido
    and ped_num_conse   = @anexo
    and ppe_num_conse   = @n_partida

select @importe_t = sum( ppe_cantidad *
                        ppe_pre_unita * (1- ppe_ptj_dcto/100))
    from partida_pedido p
    where p.alm_clave   = @almacen
    and p.ped_clave    = @n_pedido
    and p.ped_num_conse = @anexo
    and p.ppe_cve_statu i= "PC"
    and p.ppe_cve_progr i= "S"

if @importe_t is null
select @importe_t = 0

select @importe_p = sum( emb_cantidad * emb_pre_unita )
    from embarque e
    where e.alm_clave   = @almacen
    and e.ped_clave    = @n_pedido
    and e.ped_num_conse = @anexo
    and e.emb_cve_statu i= "PC"

if @importe_p is null
select @importe_p = 0

update pedido set
    ped_imp_total = @importe_t + @importe_p
    where alm_clave      = @almacen
    and ped_clave       = @n_pedido
    and ped_num_conse   = @anexo
end
go

```

```

/* TRIGGER          EMBARQUES---UPDATE          ***** */
drop trigger      abembin01_upd
go

create trigger    abembin01_upd  on embarque      for update
as
begin
declare @rowcount int
        , @nivel      tinyint
        , @almacen    char(4)
        , @n_pedido   int
        , @anexo      tinyint
        , @n_partida  smallint
        , @sum_emb    float
        , @can_par    float

select @rowcount = @@rowcount
if @rowcount = 0
    return

        /* Verifica que la transaction sea solicitada por medio del sistema
        * de Abastecimiento.
exec abvalusu_pro
        */

        /* No se permite cambio de almacen(centro) ni numero de pedido. */

/* if (update (alm_clave) or update (ped_clave) or update (ped_num_conse)
or update (ppe_num_conse) or update (emb_num_conse))
begin
    raiserror 20050 'No se permite cambio de LLave.'
    rollback tran
    return
end
*/

        /* No se permite actualizar mas de un partida a la vez */

if @rowcount > 1
begin
select alm_clave, ped_clave, ped_num_conse, ppe_num_conse from inserted
group by alm_clave, ped_clave, ped_num_conse, ppe_num_conse
/* if 1 != @@rowcount
begin
    raiserror 20050
    'No se permite Actualizar embarques de mas de un partida.'
    rollback tran
    return
end */
end

if update (emb_cve_statu)
begin
if not exists (select * from inserted ins, status s
               where ins.emb_cve_statu = s.sta_clave
                  and s.sta_tipo = 'P')
begin
    raiserror 20100
    'Alguno de los embarques no tiene un estado valido'
    rollback tran
    return
end

select @almacen = alm_clave, @n_pedido = ped_clave
        , @anexo = ped_num_conse, @n_partida = ppe_num_conse
from inserted

select @nivel = min(sta_nivel)

```

```

, @sum_emb = sum (emb_cantidad)
from status s, embarque_p
  where s.sta_tipo = "P"
  and p.emb_cve_statu = s.sta_clave
  and p.alm_clave = @almacen
  and p.ped_clave = @n_pedido
  and p.ped_num_conse = @añexo
  and p.ppe_num_conse = @n_partida
  and p.emb_cve_statu != "PC" */
/*
/* si embarques ya estan autorizados valida cantidades */
if 'PA' != (select sta_clave from status
           where sta_tipo = "P" and sta_nivel = @nivel)
  /* si alguno fue autorizado */
  if exists (select * from deleted where emb_cve_statu = 'PA' )
    begin
      select @can_par = ppe_cantidad from partida_pedido
      where alm_clave = @almacen
      and ped_clave = @n_pedido
      and ped_num_conse = @añexo
      and ppe_num_conse = @n_partida
      if @sum_emb != @can_par
        begin
          raiserror 20100
          'La cantidad de la partida no es igual a la de los embarques'
          rollback tran
          return
        end
      */
    end
  end
select "ojo edo de ppe :", sta_clave from status
      where sta_tipo = "P" and sta_nivel = @nivel

update partida_pedido set
  ppe_cve_statu = (select sta_clave from status
                  where sta_tipo = "P" and sta_nivel = @nivel)
  where alm_clave = @almacen
  and ped_clave = @n_pedido
  and ped_num_conse = @añexo
  and ppe_num_conse = @n_partida
end
end
go

```

```

/* TRIGGER          EMBARQUIES---DELETE ***** */
drop trigger  abembin01_del
go

create trigger abembin01_del on embarque for delete
as
begin
declare @rowcount int
        , @nivel      tinyint
        , @almacen     char(4)
        , @n_pedido   int
        , @anexo      tinyint
        , @n_partida  smallint

select @rowcount = @@rowcount
if @rowcount = 0
return

        /* Verifica que la transaction sea solicitada por medio del sistema
        * de Abastecimiento.
exec abvaluu_pro
        */

        /* valida que solo se borren embarques a una partida a al vez */
/* if @rowcount > 1
begin
select alm_clave, ped_clave, ped_num_conse, ppe_num_conse from deleted
group by alm_clave, ped_clave, ped_num_conse, ppe_num_conse
if 1 != @@rowcount
begin
raiserror 20100 'No es valido borrar embarques a mas de una partida'
rollback tran
return
end
end
*/

/* Si el embarque a borrar tiene mov. material, rollback tran */
if exists (select * from deleted de, movimiento partida_pedido mp
where de.alm_clave = mp.alm_clave and
de.ped_clave = mp.ped_clave and
de.ped_num_conse = mp.ped_num_conse and
de.ppe_num_conse = mp.ppe_num_conse and
de.emb_num_conse = mp.emb_num_conse )
begin
raiserror 20030 'Embarque colocado, se rechaza eliminacion'
rollback tran
return
end

select @almacen = alm_clave, @n_pedido = ped_clave
, @anexo = ped_num_conse, @n_partida = ppe_num_conse from inserted

if exists (select * from embarque
where alm_clave = @almacen
and ped_clave = @n_pedido
and ped_num_conse = @anexo
and ppe_num_conse = @n_partida )
update partida_pedido set
ppe_cve_statu = "PA"
, ppe_cve_progr = "S"
where alm_clave = @almacen
and ped_clave = @n_pedido
and ped_num_conse = @anexo
and ppe_num_conse = @n_partida

```

```
else
  update partida_pedido set
    ppe_cve_statu = "PA"
  , ppe_cve_progr = NULL
    where alm_clave      = @almacen
    and ped_clave       = @n_pedido
    and ped_num_conse   = @anexo
    and ppe_num_conse   = @n_partida
end
go
```

```
revoke select, insert, update, delete on division
from logistica
go
revoke select, insert, update, delete on compania
from logistica
go
revoke select, insert, update, delete on unidad
from logistica
go
revoke select, insert, update, delete on almacen
from logistica
go
revoke select, insert, update, delete on ofna_regional
from logistica
go
revoke select, insert, update, delete on ofnareg_at_compania
from logistica
go
revoke select, insert, update, delete on comprador
from logistica
go
revoke select, insert, update, delete on linea_sublinea
from logistica
go
revoke select, insert, update, delete on compra_linea
from logistica
go
revoke select, insert, update, delete on material_unidad
from logistica
go
revoke select, insert, update, delete on material_corporativo
from logistica
go
revoke select, insert, update, delete on matun_matco
from logistica
go
revoke select, insert, update, delete on unidad_medida
from logistica
go
revoke select, insert, update, delete on requisicion
from logistica
go
revoke select, insert, update, delete on partida_requisicion
from logistica
go
revoke select, insert, update, delete on partida_requisicion_programa
from logistica
go
revoke select, insert, update, delete on tabla_autori
from logistica
go
revoke select, insert, update, delete on descr_temporal
from logistica
go
revoke select, insert, update, delete on descr_temporal_modifica
from logistica
go
revoke select, insert, update, delete on moneda
from logistica
go
revoke select, insert, update, delete on descripcion
from logistica
go
revoke select, insert, update, delete on requisicion_campo
from logistica
go
revoke select, insert, update, delete on solicitud_cotizacion
from logistica
```

```

go
revoke select, insert, update, delete on proveedor
from logistica
go
revoke select, insert, update, delete on cotizacion
from logistica
go
revoke select, insert, update, delete on partida_cotizacion
from logistica
go
revoke select, insert, update, delete on solicitud_partida_requisicion
from logistica
go
revoke select, insert, update, delete on solicitud_proveedor
from logistica
go
revoke select, insert, update, delete on domicilio_proveedor
from logistica
go
revoke select, insert, update, delete on representa_proveedor
from logistica
go
revoke select, insert, update, delete on proveedor_linea_sublinea
from logistica
go
revoke select, insert, update, delete on pedido
from logistica
go
revoke select, insert, update, delete on partida_pedido
from logistica
go
revoke select, insert, update, delete on embarque
from logistica
go
revoke select, insert, update, delete on requisicion_cotizacion
from logistica
go
revoke select, insert, update, delete on notas_requisicion
from logistica
go
revoke select, insert, update, delete on notas_pedido
from logistica
go
revoke select, insert, update, delete on notas_catalogo
from logistica
go
revoke select, insert, update, delete on tabla
from logistica
go
revoke select, insert, update, delete on tabla_detalle
from logistica
go
revoke select, insert, update, delete on agente_aduanal
from logistica
go
revoke select, insert, update, delete on factura_pedido
from logistica
go
revoke select, insert, update, delete on trafico
from logistica
go
revoke select, insert, update, delete on status_trafico
from logistica
go
revoke select, insert, update, delete on modalidad_status
from logistica
go
revoke select, insert, update, delete on movimiento_material
from logistica
go
revoke select, insert, update, delete on movimiento_partida_pedido
from logistica
go
revoke select, insert, update, delete on bulto
from logistica
go

```

revoke select, insert, update, delete on gastos_reales
from logistica
go
revoke select, insert, update, delete on pedido_modifica
from logistica
go
revoke select, insert, update, delete on partida_pedido_modifica
from logistica
go
revoke select, insert, update, delete on domicilio_temporal
from logistica
go
revoke select, insert, update, delete on almacen_domicilio
from logistica
go
revoke select, insert, update, delete on embarque_modifica
from logistica
go
revoke select, insert, update, delete on domicilio_temporal_modifica
from logistica
go
revoke select, insert, update, delete on notas_pedido_modifica
from logistica
go
revoke select, insert, update, delete on status
from logistica
go
revoke select, insert, update, delete on moneda
from logistica
go
revoke select, insert, update, delete on tipo_cambio
from logistica
go
revoke select, insert, update, delete on proveedor_convenio
from logistica
go
revoke select, insert, update, delete on status
from logistica
go
revoke select, insert, update, delete on lista_precio
from logistica
go

```

/* Procedimiento para resecuenciar partidas cuando se inserta *
* una partida entre las existentes.
* las condiciones para ejecutar este procedimiento son establecidas en la
* forma abreqin02 (Actualizacion de partidas) y debe ser ejecutado previo
* a la insercion de nueva partida(solo cuando el numero de ptda ya existe)
* Este procedimiento no debe ser ejecutado en trigger de insert
*/
create proc abreqin01_isq (@alm_clave char(4),          /* almacen */
                          @req_num_conse int,         /* num. requisicion */
                          @prq_num_conse smallint,   /* partida a insert */
                          @wrk_row_next smallint)    /* num. partdas */
as
/* Cambia num. de partida + 1, de la ultima hasta la que se insertara */
begin
    begin tran updptd
    while @wrk_row_next >= @prq_num_conse
    begin
        update partida_requisicion set prq_num_conse = prq_num_conse + 1
        where alm_clave = @alm_clave and
              req_num_conse = @req_num_conse and
              prq_num_conse = @wrk_row_next
        if @@error != 0
        begin
            raiserror 20050 'No se puede cambiar secuencia de partida'
            rollback tran updptd
            return
        end
        else
        begin
            select @wrk_row_next = @wrk_row_next - 1
            continue
        end
    end
    end commit tran updptd
    return
end
go

/* Procedimiento para resecuenciar partidas cuando se elimina una de ellas.
* Las condiciones para esto, se establecen en la forma abreqin02
* (actualizacion de partidas) y debe ser ejecutado despues de realizar
* el delete de la partida.
* Este procedimiento no debe incluirse en trigger de delete
*/
create proc abreqin01_dsq (@alm_clave char(4),          /* almacen */
                          @req_num_conse int,         /* num requisicio */
                          @prq_num_conse smallint,   /* ptda cancelada */
                          @wrk_row_next smallint)    /* num de ptdas */
as
/* Cambia num de partida -1, desde la sigte a la cancelada hasta el final */
begin
    begin tran updptd
    select @prq_num_conse = @prq_num_conse + 1
    while @prq_num_conse <= @wrk_row_next
    begin
        update partida_requisicion set prq_num_conse = prq_num_conse - 1
        where alm_clave = @alm_clave and
              req_num_conse = @req_num_conse and
              prq_num_conse = @prq_num_conse
        if @@error != 0
        begin
            raiserror 20050 'No se puede cambiar secuencia de partida'
            rollback tran updptd
            break
        end
        else
        begin
            select @prq_num_conse = @prq_num_conse + 1
        end
    end
end

```

```

                continue
            end
        end
        commit tran updptd
        return
    end
go
create table #user (clave char(7) not null)
go
/* Verifica que la transaction sea solicitada por medio del sistema
 * de Abastecimiento.
 */
create proc abvalusu_pro
as
begin
/*
    if not exists (select * from #user, comprador
                  where com_cve_usuar = user_name(-1) and
                        com_cve_usuar = clave)
        begin
            raiserror 20099 'Transaccion no Autorizada'
            rollback tran
            return
        end
    if @@error != 0
        begin
            raiserror 20100 'Acceso no autorizado'
            rollback tran
            return
        end
    end
    return */
end
go

```

```
create procedure rbalmac @alm_clave char(4) as
select * from almacen al, unidad un
where al.alm_clave = @alm_clave
and un.uni_clave = al.uni_clave
go
```

Proc Almacenado: rbcomlin

```
create procedure rbcomlin @lin_clave tinyint as
select * from linea_sublinea ls, compra_linea cl, comprador co
where ls.lin_clave = @lin_clave
and ls.lin_sub_linea is null
and cl.lin_clave = ls.lin_clave
and co.com_clave = cl.com_clave
go
```

Proc Almacenado: rbcompra

```
create procedure rbcompra @com_clave char(2) as
select * from comprador co, ofna regional of
where co.com_clave = @com_clave
and co.ofi_clave = of.ofi_clave
go
```

Proc Almacenado: rbcomsup

```
create procedure rbcomsup @com_clave_sup char(2) as
select com_clave wk_com, com_clave_sup
from comprador
where com_clave = @com_clave_sup
go
```

Proc Almacenado: rbdescr1

```
create procedure rbdescr1 @des_clave int as
select * from descripción
where des_clave = @des_clave
order by des_num_conse
go
```

Proc Almacenado: rbdestem

```
create procedure rbdestem @des_clave int as
select * from descr temporal
where des_clave = @des_clave
order by des_num_conse
go
```

```

create procedure difdias (@fec_ini    datetime,
                        @fec_fin    datetime,
                        @dias      int    output)

```

```

as
select @dias = count(*)
  from calendario
 where convert(int,cal_ano_mes + cal_dia) >=
        convert(int,convert(char(8),@fec_ini,112))
        and convert(int,cal_ano_mes + cal_dia) <=
        convert(int,convert(char(8),@fec_fin,112))
return
go

```

Proc Almacenado: rbmatco

```

create procedure rbmatco @muc_mat_corpo int    as
select * from material_corporativo
 where muc_mat_corpo = @muc_mat_corpo
go

```

Proc Almacenado: rbmatun

```

create procedure rbmatun @alm_clave char(4),
                        @mun_clave char(12) as
select des_clave, mun_num_desc, ume_clave, muc_mat_unida
  from material_unidad
 where alm_clave = @alm_clave
        and mun_clave = @mun_clave

```

Proc Almacenado: rbmatunco

```

create procedure rbmatunco @muc_mat_unida int as
select * from matun_matco
 where muc_mat_unida = @muc_mat_unida
go

```

Proc Almacenado: rbreq52

```

create proc rbreq52 @alm_clave char(4), @req_num_conse int as
select prq_num_conse wrk_num_prtda, prq_num_conse wrk_dup_prtda,
       prq_can_reque wrk_can_solic,
       mun_clave     wrk_cod_matun,
       ume_clave     wrk_uni_medid,
       prq_des_indic wrk_des_indic,
       prq_can_coloc wrk_can_coloc,
       prq_num_descr wrk_num_descr,
       ptd_des_clave wrk_des_clave,
       prq_cod_estat wrk_cod_estat,
       lin_sub_linea wrk_sub_lin,
       mco_clave     wrk_cod_corpo,
       lin_clave     wrk_lin_clave,
       com_clave     wrk_com_clave,
       prq_can_cance wrk_can_cance,
       convert(char(6),prq_fec_estat,12) wrk_fec_estat,
       des_nombre   wrk_des_matri
  from parEida_requisición pEd, descr_temporal dte
 where alm_clave = @alm_clave and
       req_num_conse = @req_num_conse and
       dte.des_clave = * ptd.des_clave and
       des_num_conse = 1
 order by prq_num_conse

```

go

```

create procedure rbreqcam @alm_cve indic char(1) as
select * from requisicion campo
where alm_cve_indic = @alm_cve_indic
go

```

Proc Almacenado: rbreqnot

```

create procedure rbreqnot @alm_clave char(4), @req_num_conse int as
select wrk_notas = count(nts_clave)
from notas requisicion
where alm_clave = @alm_clave
and req_num_conse = @req_num_conse
go

```

Proc Almacenado: rbreqnum

```

create procedure rbreqnum @alm_clave char(4), @req_num_conse int as
select cia_clave, req_num_conse, req_ord_traba,
req_num_equip, req_aut_plant, req_aut_folio,
req_aut_perio, req_num_emple, req_nom_solic,
req_tip_servi, req_cod_prior, req_imp_estim,
req_folio, req_dep_nombr, req_act_fijo,
req_num_ptdas, req_num_coloc, req_num_padid,
req_fec_asign, req_cve_statu, req_fec_elabo,
req_fec_recep, req_fec_statu, req_fec_reque,
alm_cta_cuent, alm_cta_subct, alm_cta_subc,
alm_cta_conce, mon_clave, lin_clave,
com_clave, req_des_gener,
convert(char(6), req_fec_elabo, 12) wrk_fec_elabo,
convert(char(6), req_fec_recep, 12) wrk_fec_recep,
convert(char(6), req_fec_statu, 12) wrk_fec_statu,
convert(char(6), req_fec_reque, 12) wrk_fec_reque,
convert(char(6), req_fec_asign, 12) wrk_fec_asign
from requisicion
where alm_clave = @alm_clave
and req_num_conse = @req_num_conse
go

```

Proc Almacenado: rbreqprg

```

create procedure rbreqprg @alm_clave char(4), @req_num_conse int as
select wrk_ptd_progr = count(prq_num_conse)
from partida requisicion
where alm_clave = @alm_clave
and req_num_conse = @req_num_conse
and prq_num_descr = 1
go

```

Proc Almacenado: rbreqtm01

```

create procedure rbreqtm01 as
drop table #programa
create table #programa
(alm_clave char(4) not null,
req_num_conse int not null,
prq_num_conse smallint not null,
emb_num_conse tinyint not null,
emb_cantidad float not null,
emb_fec_entre datetime null)
go

```

APENDICE C.

PROGRAMACION

NOTA: En este apéndice sólo se incluye una parte de la información referente al mismo, debido a lo extenso del sistema.

C.1. Programas en APT Workbenck (Transact-SQL)

```

create apt proc abinmen00_1() as
begin
  useform abinmen01_1
  variable comand _char(220) /* comando user keys */
  comand = 'ukey.exe 1 "Compras F1 " "OP1"'
           + ' 2 "Tráficos F2 " "OP2"'
           + ' 3 "Infraest F3 " "OP3"'
           + ' 4 "'
           + ' 5 "'
           + ' 6 "'
           + ' 7 "'
  system '%',comand
  return
end

```

Programa APT : abinmen00_2.q

```

create apt proc abinmen00_2() as
begin
  useform abinmen02_1
  variable comand _char(220) /* comando user keys */
  comand = 'ukey.exe 1 "Actualiz F1 " "OP0"'
           + ' 2 "Consulta F2 " "OP1"'
           + ' 3 "Reportes F3 " "OP2"'
           + ' 4 "Infraest F4 " "OP3"'
           + ' 5 "'
           + ' 6 "'
           + ' 7 "'
  system '%',comand
  return
end

```

Programa APT : abinmen02_1.q

```

create apt proc abinmen02_1() as
begin
  useform abinmen01_1
  if $curpick:text like 'Req%'
    wrk_formname = 'abreqin01'
  if $curpick:text like 'Sol%'
    wrk_formname = 'abctsol01'
  if $curpick:text like 'Cot%'
    wrk_formname = 'abctcot01'
  if $curpick:text like 'Col%'
    wrk_formname = 'abpegen01'
  if $curpick:text like 'Mod%'
    wrk_formname = 'abpegen02'
  if $curpick:text like 'Mov%'
    wrk_formname = 'abmmmov01'
  if $curpick:text like 'Tra%'
    wrk_formname = 'abmmsolt01'
  return
end

```

Programa APT : abinmen03_1.q

```

create apt proc abinmen03_1() as
begin
  useform abinmen01_1
  wrk_formname = 'abpegen01'
  return
end

```

```
create apt proc abinmen04_1() as
begin
    useform abinmen01 1
    wrk_formname = 'aBraqin05'
    return
end
```

Programa APT : abinmen06_1.q

```
create apt proc abinmen06_1() as
begin
    useform abinmen01 1
    if $curpick:text like '%Movia%'
        wrk_formname = 'abmmtra01'
    if $curpick:text like '%Cons%'
        wrk_formname = 'abpegen02'
    return
end
```

Programa APT : abinmen06_2.q

```
create apt proc abinmen06_2() as
begin
    useform abinmen02 1
    if $curpick:text like '%Traff%'
        wrk_formname = 'abmmtra01'
    if $curpick:text like '%Pedi%'
        wrk_formname = 'abpegen02'
    if $curpick:text like '%A. A%'
        wrk_formname = 'abmmagt01'
    if $curpick:text like '%Comp%'
        wrk_formname = 'abincia01'
    if $curpick:text like '%Prov%'
        wrk_formname = 'abinpro10'
    if $curpick:text like '%Cotiza%'
        wrk_formname = 'abinmon01'
    if $curpick:text like '%Tablas%'
        wrk_formname = 'abintab01'
    if $curpick:text like '%Solic%'
        wrk_formname = 'abmmsolt01'
    call exit_routine
    return
end
```

```

create apt proc abinmen01_1() as
begin
useform abinmen01_1
/*submit ncmag[0-16]
  sqlbegin
    drop table #user
    create table #user (clave char(7) not null)
    insert #user values (user_name(-1))
  sqlend
*/
submit
sqlbegin
  use abaste
  sqlend
submit
sqlbegin
  select com_clave
    from comprador
  where com_cve_usuar = user_name(-1)
sqlend
if $channel:count = 0
  begin
    print 'Usuario no autorizado'
    cancelform
  end
fecha_sis = $date
uname = null
upassword = null
server = null
submit
sqlbegin
  use master
  sqlend
submit
sqlbegin
  select uname = name,
    upassword = password
    from get_password
  select server = @@servername
sqlend
submit
sqlbegin
  use abaste
  sqlend
if uname is null
  begin
    print 'Error al detectar Usuario...!'
    return
  end
if upassword is null
  begin
    print 'Error al detectar Password...!'
    return
  end
if server is null
  begin
    print 'Error al detectar servidor...!'
    return
  end
call abinmen00_1 /* carga User Keys */
return
end

```

```
create apt proc abinmen01_2() as
begin
  useform abinmen02_1
  submit
  sqlbegin
    select com clave
      from comprador
     where com_cve_usuar = user_name(-1)
  sqlend
  if Schannel:count = 0
    begin
      print 'Usuario no autorizado'
      cancelform
    end
  fecha_sis = $date
  uname     = null
  upassword = null
  server    = null
  submit
  sqlbegin
    use master
  sqlend
  submit
  sqlbegin
    select uname      = name,
           upassword = password
      from get password
     select server    = @@servername
  sqlend
  submit
  sqlbegin
    use abaste
  sqlend
  if uname is null
    begin
      print 'Error al detectar Usuario...!'
      return
    end
  if upassword is null
    begin
      print 'Error al detectar Password...!'
      return
    end
  if server is null
    begin
      print 'Error al detectar servidor...!'
      return
    end
  end
  call abinmen00_2      /* carga User Keys */
  return
end
```

```
create apt proc abinmen05_1() as
begin
    useform abinmen01 1
    if $Scurpick:text like '%Divi%'
        callform abindiv01_1
    if $Scurpick:text like '%Compa%'
        callform abincia01_1
    if $Scurpick:text like '%Unid%'
        callform abinuni01_1
    if $Scurpick:text like '%Cent%'
        callform abinalm01_1
    if $Scurpick:text like '%Ofic%'
        callform abinofr01_1
    if $Scurpick:text like '%Compr%'
        callform abincom01_1
    if $Scurpick:text like '%Prov%'
        callform abinpro0_1
    if $Scurpick:text like '%Cod.A%'
        callform abinmat01_1
    if $Scurpick:text like '%Cod.Co%'
        callform abincor01_1
    if $Scurpick:text like '%U/Medi%'
        callform abinmed01_1
    if $Scurpick:text like '%Line%'
        callform abinlin01_1
    if $Scurpick:text like '%Nota%'
        callform abinnta01_1
    if $Scurpick:text like '%Tipot%'
        callform abintca01_1
    if $Scurpick:text like '%Mone%'
        callform abinmon01_2
    if $Scurpick:text like '%Camp%'
        callform abinfd01_1
    if $Scurpick:text like '%Domit%'
        callform abindom01_1
    if $Scurpick:text like '%Tabla%'
        callform abintab01_1
    if $Scurpick:text like '%A. Ad%'
        callform abmmagt01_1
    if $Scurpick:text like '%C. Pass%'
        callform abinpass01_1
    call abinmen00_1 /* reset user keys */
return
end
```

```

/* Proceso de ALTA de REQUISICION ( Header ) */
/*
* 1. Si ya existe no ejecutar proceso.
* 2. Inicializa campos que nos son de captura
* 3. Verifica existencia de campos requeridos en la requisicion.
* 4. Valida fechas. Elaboracion no > a recepcion,
*      Requerida no < a elaboracion.
* 5. Inserta la requisicion en tabla y llama a la forma de partidas *
*      para iniciar en esta la captura de partidas.
*      Pasar parametros: almacen , numero req., linea, comprador clave y *
*      nombre, numero de partidas (en alta = 0) y partidas colocadas *
*      ( en alta = 0)
* 6. Al regreso de la forma de partidas verificar si existen partidas *
*      asociadas a esta requisicion. En caso negativo, cancelar la *
*      requisicion.
* 7. Almacen y Numero = null y salvar solo numero para que el siguiente *
*      sea diferente. El almacen no, pues si es el mismo ya no se inicia *
*      liza la forma dependiente del almacen en postproceso de numero req. *
*/
create apt proc abrequin1_1() as
begin
  useform abrequin01_1
  variable status Int
  status = 0
  if wrk indexi = 1 /* indica que ya existe */
  begin
    print 'Requisicion ya existe'
    return
  end
  req_num_coloc = 0
  req_num_ptdas = 0
  req_num_pedid = 0
  req_fec_statu = Sdate
  req_cve_statu = 'SC'
  if req_tip_servi is null
    req_tip_servi = 'A' /* req_tip_servi:default */
  if req_cod_prior is null
    req_cod_prior = 'C' /* req_cod_prior:default */
  if req_act_fijo is null
    req_act_fijo = 'N' /* req_act_fijo:default */
  if wrk_fec_recep is null
    wrk_fec_recep = wrk def_recep
  if wrk_fec_elabo is null
    wrk_fec_elabo = wrk def_recep
  if wrk_fec_reque is null
    wrk_fec_reque = '000101'
  if wrk_fec_asign is null
    wrk_fec_asign = '990101'
  if cia_clave is null
    cia_clave = wok cia clave
  wrk_fec_statu = wrk def_recep
  req_tip_servi = upper(req_tip_servi)
  req_cod_prior = upper(req_cod_prior)
  req_act_fijo = upper(req_act_fijo)
  /* foreach j in $curform
  begin
    if j:name = 'uni_clave' or
       j:name = 'alm_clave' or
       j:name = 'req_num_conce' or
       j:name = 'lin_clave' or
       j:name = 'req_cod_prior' or
       j:name = 'req_tip_servi' or
       j:name = 'req_act_fijo' or
       j:name = 'com_clave')
    begin
      if j:value is null
        begin

```

```

        status = 1 /* para indicar que falta algun campo */
    end
end
if status = 1
begin
    print 'Falta informacion en la requisicion'
    positionform req_num_conse
    return
end
*/
req_fec_elabo = wrk_fec_elabo
req_fec_recep = wrk_fec_recep
req_fec_reque = wrk_fec_reque
req_fec_asign = wrk_fec_asign
if req_fec_elabo > req_fec_recep
begin
    print 'Fecha de elaboracion es mayor a fecha de recepcion'
    positionform req_fec_elabo
    return
end
if req_fec_reque < req_fec_elabo
begin
    print 'Fecha requerida es menor a fecha de elaboracion'
    positionform req_fec_reque
    return
end
submit
sqlbegin
    insert requisicion (cia_clave,uni clave,
alm clave, req_num_conse,
req_ord traba, req_num equip,
alm_cta cuent, alm_cta subct,
alm_cta ssubc, alm_cta conce,
req_aut plant, req_aut folio,
req_aut perio, req_num emple,
req_nom solic, req_tip servi,
req_cod prior, req_imp estim,
mon clave, req_num pedid,
req_folio, req_dep nombr,
req_act fiijo, req_fec asign,
com clave, req_des gener,
req_fec elabo, req_fec recep,
req_fec reque, req_num coloc,
req_num ptdas, lin clave,
req_fec statu, req_cve statu)
values ((cia_clave), {sqlexpr(uni clave)},
{sqlexpr(alm clave)}, {sqlexpr(req_num conse)},
{sqlexpr(req_ord traba)}, {sqlexpr(req_num equip)},
{sqlexpr(alm_cta cuent)}, {sqlexpr(alm_cta subct)},
{sqlexpr(alm_cta ssubc)}, {sqlexpr(alm_cta conce)},
{sqlexpr(req_aut plant)}, {sqlexpr(req_aut folio)},
{sqlexpr(req_aut perio)}, {sqlexpr(req_num emple)},
{sqlexpr(req_nom solic)}, {sqlexpr(req_tip servi)},
{sqlexpr(req_cod prior)}, {sqlexpr(req_imp estim)},
{sqlexpr(mon clave)}, {sqlexpr(req_num pedid)},
{sqlexpr(req_folio)}, {sqlexpr(req_dep nombr)},
{sqlexpr(req_act fiijo)}, {sqlexpr(req_fec asign)},
{sqlexpr(com clave)}, {sqlexpr(req_des gener)},
{sqlexpr(req_fec elabo)}, {sqlexpr(req_fec recep)},
{sqlexpr(req_fec reque)}, {sqlexpr(req_num coloc)},
{sqlexpr(req_num ptdas)}, {sqlexpr(lin clave)},
{sqlexpr(req_fec statu)}, {sqlexpr(req_cve statu)})

sqlend
if $channel:count = 0
begin
    print 'Alta de requisicion no procede'
    positionform alm_clave
    return
end
callform abrequin02_1 (alm clave, req_num_conse, com clave,
com nombre, lin clave, req_num ptdas,
req_num coloc, lin sub linea, wrk_fec_reque,
wrk_def_recep, wk_riive1)
submit
sqlbegin

```

```

select req_num ptdas from requisicion
  where alm_clave = '(alm clave)' and
         req_num_conse = (req_num_conse)
sqlend
if req_num_ptdas = 0
  begin
    wrk_indexi = 0
    submit
    sqbegin
      delete requisicion
        where alm_clave = '(alm clave)' and
              req_num_conse = (req_num_conse)
    sqlend
    if Schannel:count = 0
      begin
        print 'Error, no hay requisicion que cancelar '
        cancelform
      end
    end
  else
    wrk_indexi = 1
    call abrequin91_1
    call abrequin98_1
    return
  end
end

```

/* reset function keys */
/* cuenta ptdas programadas */

end

```

/* Proceso para llamar forma de PARTIDAS */
/*
* 1. Llamar forma de partidas,
* pasando parametros: almacen, numero de req. comprador clave y *
* numero, numero de partidas y ptas colocadas *
* 2. Al regreso de la forma de partidas, si el numero de partidas es 0,
* eliminar la requisicion.
*/
create apt proc abreqin29_1() as
begin
  useform abreqin01_1
  if wrk_indexi = 0 /* indica que no existe */
  begin
    print 'Requisicion no dada de alta'
    return
  end
  if alm_clave is null or req_num_conse is null
  begin
    print 'Identificacion de requisicion incompleta'
    positionform alm_clave
    return
  end
  callform abreqin02_1 (alm_clave, req_num_conse, com_clave,
    com_nombre, lin_clave, req_num_ptdas,
    req_num_coloc, lin_sub_linea, wrk_fec_reque,
    wrk_def_recep, wk_nivel)

  submit
  sqlbegin
    select req_num_ptdas, req_cve_statu from requisicion
    where alm_clave = '{alm_clave}' and
    req_num_conse = {req_num_conse}
  sqlend
  if req_num_ptdas = 0
  begin
    wrk_indexi = 0
    submit
    sqlbegin
      delete requisicion
      where alm_clave = '{alm_clave}' and
      req_num_conse = {req_num_conse}
    sqlend
    if $channel:count = 0
    begin
      print 'Error, no hay requisicion que cancelar'
    end
    call abreqin02_1 /* reset campos */
  end
  else
    wrk_indexi = 1
    call abreqin91_1 /* reset user keys */
    call abreqin98_1 /* lee partidas programadas */
    positionform alm_clave
  return
end
end

```

```

create apt proc abctsol63_1() as
begin
  useform abctsol01_1
  channel canal01
  variable long_comentarios int,
           contador         int,
           porcion         int
  submit on canal01
  sqlbegin
    select *
      from solicitud_cotizacion
     where sco_num_conse = {sco_num_conse}
  sqlend
  if canal01:count = 1
    return
  if sco_fec_lim_cotiz is null
    begin
      print 'La fecha limite de cotizaciones requerida...'
      positionform sco_fec_lim_cotiz
      return
    end
  /* if sco_comentarios:value is null or
     datalength(sco_comentarios) = 0
     begin
       print 'Las notas son requeridas...'
       positionform sco_comentarios
       return
     end */
  if sco_comentarios:value is null or
     datalength(sco_comentarios) = 0
     des_clave = null
  else
    begin
      submit on canal01
      sqlbegin
        select des_clave = max(des_clave)
          from descripcion
      sqlend
      if des_clave is null
        des_clave = 0
      des_clave = des_clave + 1
    end
    submit on canal01
    sqlbegin
      insert solicitud_cotizacion (sco_fec_sol,
                                   sco_num_conse,
                                   sco_fec_lim_cotiz,
                                   des_clave)
      values
        ({sqlexpr(sco_fec_sol)},
         {sqlexpr(sco_num_conse)},
         {sqlexpr(sco_fec_lim_cotiz)},
         {des_clave})
    sqlend
  if canal01:errors is not null or
     canal01:count = 0
     print 'Error al dar de alta la Solicitud de cotizacion...'
  else
    begin
      if sco_comentarios:value is not null or
         datalength(sco_comentarios) != 0
         begin
           long_comentarios = datalength(sco_comentarios)
           contador         = 0
           porcion         = 1
           while long_comentarios > 0
             begin
               des_nombre   = substring(sco_comentarios,porcion,255)
               contador     = contador + 1
             end
           end
         end
    end
  end
end

```

```

submit on canal01
sqlbegin
  insert descripcion
  values ({des_clave},{contador},' {des_nombre}')
sqlend
if canal01:errors is not null
begin
  print "No se pudo insertar descripcion"
  break
end
else
begin
  porcion = porcion + 255
  long_comentarios = long_comentarios - 255
end
end
end
canal01:errors = null
submit on canal01
sqlbegin
  insert into solicitud_partida_requisicion (sco_num_conse,
                                             alm_clave,
                                             req_num_conse,
                                             prq_num_conse,
                                             des_clave)
select sco_num_conse,
       alm_clave,
       req_num_conse,
       prq_num_conse,
       partida_requisicion.des_clave
from solicitud_cotizacion,
     partida_requisicion
where sco_num_conse = {sco_num_conse}
and req_num_conse = {req_num}
and alm_clave = '{alm_clave}'
sqlend
/*
call abtcsol65 1
call abtcsol83 1
call abtcsol61_1 */
end
return
end

```

```
create apt proc abctsol91_1() as
begin
  useform abctsol01 1
  variable sqlins c$ar(3000)
  channel canal01
  canal01:errors = null
  sqlins = 'insert into
           solicitud_partida_requisicion ( sco_num_conse,
                                           alm_clave,
                                           req_num_conse,
                                           prq_num_conse,
                                           des_clave)
           select sco_num_conse,
                  alm_clave,
                  req_num_conse,
                  prq_num_conse,
                  partida_requisicion.des_clave
           from solicitud_cotizacion,
                partida_requisicion
           where sco_num_conse = ' + sqlexpr(sco_num_conse)
  if $curpick:id = 85
    sqlins = sqlins + ' and req_num_conse = ' + sqlexpr(wk_req_act)
  if $curpick:id = 86
    sqlins = sqlins + ' and req_num_conse between '
    sqlins = sqlins + ' + sqlexpr(wk_req_ini) + ' and ' + sqlexpr(wk_req_fin)
  sqlins = sqlins + ' and alm_clave = ' + sqlexpr(alm_clave)
  submit on canal01
  sqlbegin
  {sqlins}
  sqlend
  if canal01:errors is not null
  begin
    print 'Error al insertar'
    print '%',canal01:text
  end
  return
end
```

```

create apt proc abctcot05_1() as
begin
  useform abctcot01_1
  channel canal01
  canal01:errors = null
  if fecha_vig is null
  begin
    print 'La Fecha de Vigencia es requerida...'
    positionform fecha_vig
    return
  end
  save field mon_clave
  call abctcot14_1
  if mon_clave != mon_clave:savedvalue
  return
  if mon_clave is null or
  datalength(mon_clave) = 0
  begin
    print 'El Tipo de Moneda es requerido...'
    positionform mon_clave
    return
  end
  if prv_tpo_entre is null
  begin
    print 'El Tiempo de Entrega es requerido...'
    positionform prv_tpo_entre
    return
  end
  save field prv_lug_entre
  call abctcot19_1
  if prv_lug_entre != prv_lug_entre:savedvalue
  return
  if prv_lug_entre is null or
  datalength(prv_lug_entre) = 0
  begin
    print 'El Lugar de Entrega es requerido...'
    positionform prv_lug_entre
    return
  end
  if prv_dia_condi_pago is null
  prv_dia_condi_pago = 0
  /* if prv_cnd_pago is not null or
  datalength(prv_cnd_pago) > 0
  begin
    if prv_dia_condi_pago is null or
    prv_dia_condi_pago = 0 or
    datalength(prv_cnd_pago) = 0
    begin
      print 'Los Dias de Condicion de Pago son requeridos...'
      positionform prv_dia_condi_pago
      return
    end
  end */
  if prv_dia_condi_pago is not null or
  prv_dia_condi_pago > 0
  begin
    save field prv_cnd_pago
    call abctcot15_1
    if prv_cnd_pago != prv_cnd_pago:savedvalue
    return
    if prv_cnd_pago is null
    begin
      print 'La Condicion de Pago es requerida...'
      positionform prv_cnd_pago
      return
    end
  end
  if prv_dto_pront_pago is null

```

```

prv_dto_pront_pago = 0
if prv_dto_pront_pago > 0
begin
  if prv_dia_pront_pago is null or
  prv_dia_pront_pago = 0
  begin
    print 'Los Dias Por Pronto Pago son requeridos...'
    positionform prv_dia_pront_pago
    return
  end
end
if prv_dia_pront_pago is null
prv_dia_pront_pago = 0
if prv_dia_pront_pago > 0
begin
  if prv_dto_pront_pago is null or
  prv_dto_pront_pago = 0
  begin
    print 'El Porcentaje Por Pronto de Pago es requerido...'
    positionform prv_dto_pront_pago
    return
  end
end
save field prv_cnd_entre
call abctcot17_1
if prv_cnd_entre != prv_cnd_entre:savedvalue
return
/* Alta */
submit on canal01
sqlbegin
insert cotizacion (prv_rfc,
cot_clave,
cot_fec_cotiz,
cot_fec_limit_vigen,
mon_clave,
prv_cnd_pago,
prv_dia_condi_pago,
prv_dto_gener,
prv_tpo_entre,
prv_lug_entre,
prv_cnd_entre,
prv_dia_pront_pago,
prv_dto_pront_pago,
prv_rfc_repre,
pdo_num_conse,
cot_des_lug)
values ({{sqlexpr(prv_rfc)},
{sqlexpr(cot_clave)},
{sqlexpr(cot_fec_cotiz)},
{sqlexpr(fecha_vig)},
{sqlexpr(mon_clave)},
{sqlexpr(prv_cnd_pago)},
{sqlexpr(prv_dia_condi_pago)},
{sqlexpr(prv_dto_gener)},
{sqlexpr(prv_tpo_entre)},
{sqlexpr(prv_lug_entre)},
{sqlexpr(prv_cnd_entre)},
{sqlexpr(prv_dia_pront_pago)},
{sqlexpr(prv_dto_pront_pago)},
{sqlexpr(prv_rfc_repre)},
{sqlexpr(pdo_num_conse)},
{sqlexpr(des_lugar)}})
sqlend
if canal01:count = 0
begin
  print 'Error al dar de alta la cotizacion...'
  print '%',canal01:text
end
else
begin
  if sco_num_conse is not null
  begin
    submit on canal01 nomsg [0-16]
    sqlbegin
      update solicitud_proveedor
      set cot_clave = '{cot_clave}'

```

```

      where sco_num_conse = {sco_num_conse} and
            prv_rfc       = '{prv_rfc}'
sqlend
if canal01:count = 0
  begin
    print 'Error al actualizar proveedor_solicitud...'
    print '%',canal01:text
  end
end
save field alm_clave
save field alm_nombre
save field wk_req_ini
save field wk_req_fin
callform abctcot02_1(prv_rfc,
                    prv_rzo_socia,
                    cot_clave,
                    wk_req_ini,
                    wk_req_fin,
                    alm_clave,
                    sco_num_conse)
call abctcot12_1
/* wk_req_ini =wk_req_ini:savedvalue
wk_req_fin = wk_req_fin:savedvalue
alm_clave = alm_clave:savedvalue
alm_nombre = alm_nombre:savedvalue*/
end
return
end

```

```

create apt proc abctcot08_1() as
begin
  useform abctcot01 1
  variable command Char(220)
  menu act 67
  menu alta 72
  menu baja 73
  menu camb 74
  menu con 68
  menu sol 69
  menu pro 66
  menu ant 70
  menu sig 71
  menu bor 77
  menu sal 78
  menu asoc 76
  channel canal01
  if ($curpick:id = 62 or
    $curpick:id = 65 ) and
    (alta:active = false and
    cot_clave is not null and
    prv_rfc is not null)
  if counter is null
  begin
    canal01:errors = null
    submit on canal01
    sqlbegin
      select *
      from partida cotizacion
      where cot_clave = '{cot_clave}'
      and prv_rfc = '{prv_rfc}'
    sqlend
    if canal01:count = 0
    begin
      print 'La cotizacion se dara de baja por carecer de
        Partida(s).
          Confirme por Favor...'
      counter = 0
      return
    end
  end
  else
  begin
    canal01:errors = null
    submit on canal01
    sqlbegin
      delete cotizacion
      where cot_clave = '{cot_clave}'
      and prv_rfc = '{prv_rfc}'
    sqlend
    if canal01:count = 0
    print 'Error al dar de baja la Cotizacion...'
    print 'Cotizacion dada de baja por carecer de Partidas...'
  end
  if $curpick:id = 78
  begin
    wrk_formname = 'abinmen01'
    callextern exit_routine
    return
  end
  command = 'ukey.exe'
  if (canal01:open = false)
  connect canal01
  call abutlim01 1
  cotiz_no:visible = false
  cotiz_de:visible = false
  prv_nacional:visible = false

```

```

cot_clave:tabable = true
sco_num_conse:tabable = true
prv_rzo_socia:tabable = true
des_lugar:tabable = true
alm_clave:tabable = false
wk_req_ini:tabable = false
wk_req_fin:tabable = false
wk_req_fin:visible = false
coE_foS_cotiz = $date
act:active = false
alta:active = false
baja:active = false
camb:active = false
con:active = true
sol:active = false
pro:active = false
ant:active = false
sig:active = false
bor:active = false
sal:active = true
asoc:active = true
command = command + ' 1 "Actualiz F1 " "OP1"'
+ ' 2 "Consulta F2 " "{18-"'
+ ' 3 " Solici. F3 " "{19-"'
+ ' 4 " Prov. F4 " "{20-"'
+ ' 5 " C/Ant. F5 " "{21-"'
+ ' 6 " C/Sig. F6 " "{23-"'
+ ' 7 "Sol Asoc F7 " "{24-"'

system '%',command
positionform cot_clave
return
end

```

```

/* PREPROCESO DE FORMA ABPEGEN01 */
create apt proc abpegen00_1() as
begin
    useform abpegen01_1
    channel mychannel
    if mychannel:open = false
        connect mychannel
    alm_clave = null
    wrk_num_conse = null
    wrk_ped_conse = null
    if alm_clave is not null
        begin
            if wrk_num_conse is null or
                wrk_ped_conse is null
                begin
                    print 'Error, faltan parametros abpegen00'
                    cancelform
                end
            end
        submit on mychannel
            sqlbegin
                select com_clave, com_nivel, com_nombre
                from comprador
                where com_cve_usuar = user_name(-1)
            sqlend
        if mychannel:count = 0
            begin
                print 'Usuario no autorizado'
                cancelform
                return
            end
        if com_nivel < 80
            com_clave:tabable = false
        submit
            sqlbegin
                select tca_tip_cambi
                from tipo_cambio
                where mon_clave = 'US'
                and tca_fec_cotiz > dateadd(day, -1, '{ $date }')
                and tca_fec_cotiz < dateadd(day, 1, '{ $date }')
            sqlend
        if $channel:count = 0
            begin
                print 'El tipo de cambio del dollar no a sido capturado'
            end
        if alm_clave is not null
            begin
                alm_clave:tabable = false
                com_clave:tabable = false
                wrk_prv_rfc:tabable = false
                wrk_rzo_socia:tabable = false
                submit on mychannel
                    sqlbegin
                        select prv.prv_rfc wrk_prv_rfc, ped.com_clave,
                            prv.prv_rzo_socia wrk_rzo_socia, ped_tpo_pedido
                        from pedido ped, proveedor_prv
                        where alm_clave = '{alm_clave}' and
                            ped_clave = {wrk_num_conse} and
                            ped_num_conse = {wrk_ped_conse} and
                            prv.prv_rfc = ped.prv_rfc
                    sqlend
                if mychannel:count = 0
                    begin
                        print 'Pedido no existe'
                        cancelform
                    end
                submit on mychannel
                    sqlbegin

```

```

select com_nivel, com_nombre
      from comprador
      where com_clave = '{com_clave}'
sqlend
if mychannel:count = 0
  begin
    print 'Usuario no autorizado'
    cancelform
  end
  if ped_tpo_pedid = 1
    wrk_tpo_pedid = 'S'
  else
    wrk_tpo_pedid = 'N'
  wrk_com_nivel = 0
  wrk_ind_pedido = 1
  wk_prov = wrk_prv_rfc
  positionform wrk_uñi_conse
end
wrk_com_nivel = com_nivel
wrk_com_clave = com_clave
save field com_clave
call abpegen09_1 /* borra caampos */
call abpegen43_1 /* User Keys */
return

```

end

```

create apt proc abpegen15_1() as
begin
  useform abpegen01_1
  channel mychannel
  channel ptda_channel
  if wrk_num_selec = 0
  begin
    print 'No ha seleccionado requisicion'
    positionform wrk_req_selec
    wrk_req_selec = null
    return
  end
  if wrk_tpo_pedid is null
  begin
    perform wrk_tpo_pedid post
  end
  if alm_clave is null
  begin
    print 'Debe especificar Centro'
    positionform alm_clave
    return
  end
  if com_clave is null
  begin
    print 'Debe especificar Clave de Comprador'
    positionform com_clave
    return
  end
  if wrk_prv_rfc is null
  begin
    print 'Debe especificar Proveedor'
    positionform wrk_rzo_socia
    return
  end
  wrk_tpo_pedid = 'N'
  submit on mychannel nomsg[0-16]
  sqlbegin
  drop table #requis
  sqlend
  mychannel:errors = null
  submit on mychannel
  sqlbegin
  create table #requis
  (req_num_conse int not null,
  cot_clave char(14) null,
  prv_rfc char(14) null)
  sqlend
  foreach j in req_cot
  begin
    if j.req_num_conse:hilite = true
    begin
      submit on mychannel
      sqlbegin
      insert #requis (req_num_conse,
      cot_clave,
      prv_rfc
      values
      ({sqlexpr(j.req_num_conse)},
      {sqlexpr(j.cot_clave)},
      {sqlexpr(j.prv_rfc)})
      sqlend
      if mychannel:count = 0
      begin
        print 'Error al paso de parametros abpegen15'
        return
      end
    end
  end
end
end

```

```

callform abpegen03_1 (alm_clave, wrk_num_conse, wrk_ped_conse,
                    wrk_num_selec, wrk_tpo_pedid,
                    wrk_prv_rfc, wrk_ind_pedido, wok_des_part)
if alm_clave = '9999'
begin
  alm_clave = alm_clave:savedvalue
  positionform wrk_req_selec
  call abpegen43_1 /* restore user keys */
  return
end
if alm_clave:tabable = false
exitform
else
begin
  if ptda_channel:open = false
  connect ptda_channel
  submit on ptda_channel nomsg[0-16]
  sqlbegin
  drop table #parametro
  sqlend
  submit on ptda_channel
  sqlbegin
  create table #parametro
  (alm_clave          char(4)    not null,
   ped_clave         int        not null,
   ped_num_conse     tinyint   not null,
   prv_rfc           char(20)   not null,
   com_clave         char(2)    not null,
   cot_clave         char(14)   not null,
   ped_tpo_pedid    tinyint    null)
  insert #parametro values
  ((sqlexpr(alm_clave)), (sqlexpr(wrk_num_conse)),
   (sqlexpr(wrk_ped_conse)), (sqlexpr(wrk_prv_rfc)),
   (sqlexpr(com_clave)), (sqlexpr(wrk_cot_clave)),
   (sqlexpr(ped_tpo_pedid)))
  sqlend
  if ptda_channel:count = 0
  begin
  print 'Error, no paso parametros'
  cancelform
  end
  wrk_formname = 'abpegen02'
  callextern exit_routine
  cancelform
end
return
end
end

```

```

create apt proc abmmmov12_1() as
begin
/* FUNCION: Da de alta en "movimiento material".
*****T*****/
useform abmmmov01_1
/* Boton de Movimientos */
menu movto 65 */
menu altm 73 /* Alta Manual del Movimiento */
menu bajm 67 /* Baja del Movimiento */
menu camm 68 /* Cambio del Movimiento */
/* Boton de Partidas */
menu parti 72 */
menu todp 83 /* Alta/Cambio todas las partidas ajustando cant. al Eve. Ant. */
menu canp 82 /* Cancela el Alta de la Partida */
menu bajp 81 /* Da de baja la partida marcada */
/* Boton de Facturas */
menu factu 87 */
/* Boton de Bultos */
menu bul 61 */
/* Boton de Salida */
menu otrx 91 /* Otro Trafico */
menu otrm 92 /* Otro Movimiento */
menu pesi 93 /* Pedido Siguiete */
menu sal 94 /* Salida de la Forma */
if mov_num_conse is not null
begin
submit
sqlbegin
select mov_tipo
from movimiento_material
where agt_clave = '{agt_clave}'
and tra_clave = '{tra_clave}'
and mov_tipo = '{mov_tipo}'
sqlend
if $channel:count > 0
begin
print "Ya existe un % para este tráfico !...",mov_tipo
return
end
end
if mov_tipo in ('EM','RA')
begin
factu:active = true
agt_clave = 'AB'
submit
sqlbegin
if exists (select *
from movimiento_material
where agt_clave = 'AB')
begin
select tra_clave = convert(char(14),max(convert(int,tra_clave) + 1))
from movimiento_material
where mov_tipo in ('EM','RA')
end
else
select tra_clave = "1"
sqlend
end
submit
sqlbegin
select mov_num_conse = max(mov_num_conse) + 1
from movimiento_material
sqlend
if mov_num_conse is null
mov_num_conse = 1
if mov_fecha is null
mov_fecha = $date
if mov_gui_talon is null

```

```

begin
  print 'El numero de guia o talon es requerido.'
  positionform mov_gui_talon
  return
end
if mov_tipo = 'RX'
begin
  $channel:errors = null
  submit
  sqlbegin
    select sta_nivel
      from status
     where sta_clave = 'RT'
       and sta_tipo = 'T'
  sqlend
end
$channel:errors = null
submit
sqlbegin
  begin tran actualiza movimiento modalidad
  insert movimiento_maEerial (agt_clave,
                             tra_clave,
                             mov_tipo,
                             mov_fecha,
                             mov_num_conse,
                             mov_cnd_flete,
                             mov_remitente,
                             mov_gui_talon,
                             mov_tip_trans)
    values (('agt_clave'),
           ('tra_clave'),
           ('mov_tipo'),
           ('mov_fecha'),
           ('mov_num_conse'),
           ('mov_cnd_flete'),
           ('mov_remitente'),
           ('mov_gui_talon'),
           ('mov_tip_trans'))

  if @@error != 0
    rollback tran
  commit tran
sqlend
if $channel:errors is not null
  print "Error al dar de alta el movimiento de material !..."
else
  begin
    bul:active = true
    factu:active = true
    if mov_tipo = 'RX'
      begin
        submit
        sqlbegin
          if not exists (select *
                        from status_ trafico
                       where agt_clave = '{agt_clave}'
                         and tra_clave = '{tra_clave}'
                         and str_cve_statu = 'RT')
            insert status_ trafico (agt_clave,
                                   tra_clave,
                                   str_cve_statu,
                                   str_nivul,
                                   str_fec_statu)
              values (('agt_clave'),
                    ('tra_clave'),
                    'RT',
                    (sta_nivel),
                    ('mov_fecha'))
          sqlend
        end
        sw_exi_movto = 0
        alm:active = false
        bajm:active = true
        camm:active = true
        if alm_clave is null
          begin
            parti:active = false

```

```
    todp:active = false
    canp:active = false
    bajp:active = false
    pesi:active = false
    alm_clave:tabable = true
    ped_clave:tabable = true
    ped_num_conse:tabable = true
    ppe_num_conse:tabable = false
    positionform alm_clave
end
else
begin
    parti:active = true
    todp:active = true
    canp:active = false
    bajp:active = false
    pesi:active = true
    alm_clave:tabable = false
    ped_clave:tabable = false
    ped_num_conse:tabable = false
    ppe_num_conse:tabable = true
    positionform ppe_num_conse
end
end
return
end
```

```

create apt proc abmmmov38_1() as
begin
/* FUNCION: Llama la forma abmmmov02_1 "pedidos por movimiento".
*****/
useform abmmmov01_1
if agt_clave is null or
tra_clave is null or
mov_tipo is null
return
save field alm_clave
save field ped_clave
save field ped_num_conse
callform abmmmov02_1(agt_clave,
tra_clave,
mov_tipo,
mov_num_conse,
alm_clave,
ped_clave,
ped_num_conse)
if (alm_clave is not null) and
(alm_clave != alm_clave:savedvalue or
ped_clave != ped_clave:savedvalue or
ped_num_conse != ped_num_conse:savedvalue)
call abmmmov09_1
else
begin
alm_clave = alm_clave:savedvalue
ped_clave = ped_clave:savedvalue
ped_num_conse = ped_num_conse:savedvalue
end
return
end
end

```

C.2. Programas en SQR

```

|*****
|*   Reporte del Depto. Corp. de Abastecimiento   *
|*   Requisiciones Recibidas                     *
|* Realizado por: Sistemas Logist. y Abaste. (GCN) *
|* Programa: abrqerr01_1.sqr                     *
|* Fecha de Realizaciun: 09/02/93                *
|* Ultima Actualizacion: 15/07/93                *
|*****
begin-report
do obtener_periodo
do imprimir_reqs
end-report

begin-setup
page-size 80 132
end-setup

begin-heading 15
print 'FECHA DE PROCESO: '
date-time
page-number
print 'abrqrec01_1'
print 'SERVICIOS INDUSTRIALES PEIOLES, S.A. DE C.V.'
print 'Direccion de Logística y Abastecimiento'
print 'Subdireccion de Abastecimiento'
print 'REPORTE MENSUAL DEL DEPTO. CORP. DE ABASTECIMIENTO'
print $tipo_req
print $periodo
print 'Clasificadas por Area y Comprador'
if #ultima_pag = 0
print 'AREA.' (11,20)
print $cve_area (0,26,2)
print $den (0,30,25)
print 'COMPRADOR:' (0,80)
print $cve_comp (0,91,2)
print $nom_comp (0,94,29)
print ' ' (12,2,128) fill
print 'NUM' (13,18)
print 'FECHA' (0,25)
print 'FECHA' (0,100)
print 'DIAS' (0,108)
print 'SEM' (0,117)
print 'MESES' (0,124)
print 'ALM' (14,3)
print 'FOLIO' (0,+3)
print 'PARTS' (0,+3)
print 'RECIB' (0,+3)
print 'DESCRIPCION GENERICA' (0,+20)
print 'ST' (0,+25)
print 'STATUS' (0,+3)
print 'TRANS' (0,+2)
print 'TRANS' (0,+3)
print 'TRANS' (0,+3)
else
print ' ' (11,2,128) fill
print ' ' (12,2,128) fill
end-if
end-heading

|*****
|* Procedimiento que valida las fechas del periodo *
|*****

begin-procedure valid_fecha
move 0 to #err
extract $dial from $fechal 4 2

```

```

extract $mes1 from $fecha1 2 2
extract $año1 from $fecha1 0 2
extract $dia2 from $fecha2 4 2
extract $mes2 from $fecha2 2 2
extract $año2 from $fecha2 0 2
if $mes1 > '12' or $mes2 > '12'
  move 1 to #err
else
  if $dia1 > '31' or $dia2 > '31'
    move 1 to #err
  end-if
end-if
if $fecha1='' or $fecha2=''
  !show (23,5) clear-line
  !show (23,5) 'ERROR: Las fechas no deben estar en blanco !! ' beep
  input $pausa noprompt type=char
else
  if #st1 != 0 or #st2 != 0
    ! show (23,5) 'Las fechas deben ser numéricas con un máximo de 6 cifras !!' be
    input $pausa noprompt type=char
  else
    if #err != 0
      !show (23,5) clear-line
      !show (23,5) 'ERROR: La fecha ha sido mal proporcionada !!' beep
      input $pausa noprompt type=char
    else
      if $fecha1 > $fecha2
        !show (23,5) clear-line
        !show (23,5) 'ERROR: La fecha de inicio debe ser menor o igual que la d
        input $pausa noprompt type=char
      else
        move 1 to #ok
      end-if
    end-if
  end-if
end-if
end-procedure

```

```

|*****
!* Procedimiento que obtiene las fechas del período del
!* cual se desea el reporte de las requisiciones.
|*****

```

```

begin-procedure obten_periodo
input $fecha1 maxlen=6 noprompt status=#st1 type=integer
input $fecha2 maxlen=6 noprompt status=#st2 type=integer
!do valid fecha
extract $año1 from $fecha1 4 2
extract $mes1 from $fecha1 2 2
extract $año2 from $fecha2 4 2
extract $mes2 from $fecha2 2 2
extract $año2 from $fecha2 0 2
string $dia1 $mes1 $año1 by '/' into $fec1
string $dia2 $mes2 $año2 by '/' into $fec2
let $período="en el período comprendido del '||$fec1||' al '||$fec2||'"
end-procedure

```

```

|*****
!* Procedimiento que imprime las Requisiciones Colocadas
!* y Canceladas correspondientes a cada comprador. <Principal>
|*****

```

```

begin-procedure imprime_reqs
move 1 to #termino
let $tipreq='REQUISICIONES RECIBIDAS'
let $línea='COMPRADOR RECIBIDAS PARTIDAS DIAS POR MOVTO'
do imp_req_rec
end-procedure

```

```

|*****
!* Procedimiento que imprime las Requisiciones Colocadas
!* correspondientes a cada comprador. <Principal>
|*****

```

begin-procedure imp_req_rec

```
create-array name=resum size=25                !Inicializa Arreglo para el
field=ar_cve_comp:char                        !resumen Final.
field=ar_req_colo:number
field=ar_dias_tot:number
field=ar_promd_mov:number
field=ar_tot_parts:number
```

!Inicializa variables para el resumen por comprador.

```
let $tipo_req=$tipreq
move 0 to #n                                ! Inicializa Índice para el arregloO
move 0 to #nmova                             ! N°mero de movimientos/comprador
move 0 to #ndias                             ! Total de días/comprador
move 0 to #promd_mov                         ! Prom. días/movimiento
move 0 to #ultima_pag                       ! Var. para indicar que es la última hoja
move 0 to #combcap                           ! Indica cuantos son del Área Bienes de Capital
move 0 to #compr1                            ! Indica cuantos son del Área Materias Primas
move 0 to #tot_part_com                     ! Indica Partidas/Comprador.
begin-select
requisicion.com_clave (0,0) on-break after=imp_resum print=never save=$com_ant
tde.clave,tde_descri
comprador.com_clave,comprador.com_area,com_nombre,req_des_gener
alm.clave (+1,3,4)
req_num_conse (0,+2,5) edit 99999
req_num_ptdas (0,+4,3) edit 999 ! Num. Partidas/Req.
req_fec_recep (0,+3,8) edit DD/MM/YY
print $req_des_gener (0,+4,55) wrap 55 1 STRIP=<13> ON=<10>
req_cve_statu (0,+4,2)
req_fec_statu (0,+2,8) edit DD/MM/YY
datediff(day,req_fec_recep,getdate()) $wk_dias_trans
if $wk_dias_trans < 0
  print '0' (0,+2,4) edit 9999 ! Si los días transcurridos son
  ! negativos se imprime un '0'
else
  add $wk_dias_trans to #ndias ! Suma días transcurridos
  print $wk_dias_trans (0,+2,4) edit 9999 ! Imprime días transcurridos
end-if
do sem_tra($req_fec_recep,#sem_ret)
do mes_tra($req_fec_recep,'R',#mes_ret)
print #sem_ret (0,+4,3) edit 999 ! Semanas de Retraso
print #mes_ret (0,+5,3) edit 999 ! Meses de Retraso
add 1 to #nmova ! Cuenta movimientos/Comprador
add $req_num_ptdas to #tot_part_com ! Cuenta Partidas/Comprador
!move $tde.clave to $cve_area
move $comprador.com_clave to $cve_area
move $tde_descri to $desc
move $comprador.com_clave to $com_comp
from table $detalle,requisicion,comprador
where requisicion.com_clave=comprador.com_clave ! Compara Cve. Comp.
and com_area=$tde.clave ! Compara Area
and req_cve_statu != 'RT'
and convert(char(6),req_fec_recep,12) between $fecha1 and $fecha2
and tab_num_conse=18
order by requisicion.com_clave,alm.clave,req_fec_recep ! Comprador
end-select
move 1 to #ultima_pag !Indica que se imprimira el
do resumen final !Resumen Final.
end-procedure
```

! Procedimiento que obtiene las Semanas de retraso

```
!Include '/sist01/leg/aba/abaeste/reportes/include/abanti01_1.inc'
```

```
!*****
!* Procedimiento que calcula e imprime el resumen de movimientos *
!* para cada comprador. Llamado por "imp_req_pend" *
!*****
```

```
begin-procedure imp_resum
if #nmova > 0 ! Solo si hubo movimientos
  let #promd_mov = round(#ndias/#nmova,0) ! calcula los promedios.
else
  move 0 to #promd_mov
end-if
! Coloca los datos del n-esimo comprador en el arreglo "resum".
add 1 to #n ! Incrementa en 1 el Índice
```

```

put $com ant #nmova #ndias #promd_mov #tot_part com into resum(#n)
if $cve_area='02'
  add 1 to #combcap
end-if
if $cve_area='10'
  add 1 to #commpri
end-if

! Imprime Resumen por comprador

print 'TOTAL DE MOVIMIENTOS:' (+2,3)
print #nmova (0,40,5) edit 9,999
print 'TOTAL PARTIDAS:' (0,84)
print #tot_part com (0,123,5) edit 9,999
print 'TOTAL DE DIAS:' (+1,3)
print #ndias (0,39,6) edit 99,999
print 'PROMEDIO DE DIAS POR MOVIMIENTO:' (0,84)
print #promd_mov (0,123,5) edit 9,999
move 0 to #nmova !Inicializa variables para
move 0 to #ndias !el siguiente comprador.
move 0 to #tot_part_com
new-page !Cambia de hoja
end-procedure

!*****
!** Procedimiento que Imprime el Resumen Final **
!*****

begin-procedure resumen_final
  print 'REQUISICIONES' PROMEDIO DIAS' (+1,4
  print $linea (+1,32)
  print '-' (+1,32,9) fill
  print '-' (0,46,13) fill
  print '-' (0,62,8) fill
  print '-' (0,74,6) fill
  print '-' (0,85,13) fill
  !Inicializa variables para los totales.
  move 0 to #tot1 ! Total requisiciones colocadas/cancel.
  move 0 to #tot2 ! Total dias transcurridos
  move 0 to #tot3 ! Promedio dias por movimiento.
  move 0 to #tot4 ! Total Partidas/Comprador
  let #bc mp = #combcap + #commpri
  let #ncom = #combcap
  move 1 to #c
  while #c <= #bc mp
    ! Inicializa Variables para los subtotaes:
    move 0 to #subtot1 ! Subtotal requisiciones colocadas
    move 0 to #subtot2 ! Subtotal dias pronosticados
    move 0 to #subtot4 ! Subtotal partidas/comprador
    while #c <= #ncom
      get $dt_cve_comp #dt_req_coca #dt_dias tot #dt_promd_mov #dt_tot_parts from res
      print $dt_cve_comp (+1,35,2) edit 99
      print #dt_req_coca (0,51,3) edit 999
      print #dt_tot_parts (0,64,5) edit 9,999
      print #dt_dias tot (0,74,6) edit 99,999
      print #dt_promd_mov (0,90,3) edit 999
      add #dt_req_coca to #subtot1
      add #dt_dias tot to #subtot2
      add #dt_tot_parts to #subtot4
      add 1 to #c
    end-while
    if #subtot1>0
      let #subtot3=round(#subtot2/#subtot1,0)
    end-if
    print '-' (+1,46,13) fill
    print '-' (0,62,8) fill
    print '-' (0,74,6) fill
    print '-' (0,85,13) fill
    print 'SUBTOTAL' (+1,32)
    print #subtot1 (0,51,3) edit 999
    print #subtot4 (0,63,6) edit 99,999
    print #subtot2 (0,74,6) edit 99,999
    print #subtot3 (0,90,3) edit 999
    add #subtot1 to #tot1
    add #subtot2 to #tot2
    add #subtot4 to #tot4
  end-while
end-procedure

```

```

print ' ' (+1,2)
move #bc_mp to #ncom
end-while
if #tot1 > 0
let #tot3=round(#tot2/#tot1,0)
else
move 0 to #tot3
end-if
print '-' (+1,46,13) fill
print '-' (0,62,8) fill
print '-' (0,73,7) fill
print '-' (0,85,13) fill
print 'TOTALES' (+1,32)
print #tot1 (0,51,3) edit 999
print #tot4 (0,62,7) edit 999,999
print #tot2 (0,73,7) edit 999,999
print #tot3 (0,90,3) edit 999
end-procedure

```

```

|*****
|*   Reporte del Depto. Corp. de Abastecimiento   *
|*           Pedidos Cerrados                   *
|*   Ordenado por Area y Comprador             *
|* Programa: abpedcer01_1.sqr                  *
|* Realizado por: Sistemas Logist. y Abaste. (GCN) *
|* Fecha de Realizaciun: 11/02/93              *
|* Ultima Actualizacion: 26/08/93              *
|*****

```

```

begin-report
do obten_periodo
do imp_ped_coloc
end-report

```

```

begin-setup
page-size 80 132
end-setup

```

```

begin-heading 15
print 'FECHA DE PROCESO: '
data-time (1,2)
page-number (0,21) DD/MM/YY
print 'abpedcer01_1' (0,119) 'HOJA: '
print 'SERVICIOS INDUSTRIALES PEIOLES, S.A. DE C.V.' (2,2)
print 'Direccion de Logística y Abastecimiento' (2) center
print 'Subdireccion de Abastecimiento' (3) center
print 'REPORTE MENSUAL DEL DEPTO. CORP. DE ABASTECIMIENTO' (4) center
print $tipo ped (6) center
print $periodo (7) center
print 'Clasificados por Area y Comprador' (8) center
if $ultima pag = 0 (9) center
! Checa que no sea la última hoja
print 'AREA: ' (11,20)
print $cve area (0,26,2)
print $dase (0,30,25)
print 'COMPRADOR:' (0,80)
print $cve_comp (0,91,2)
print $nom_comp (0,94,29)
print ' ' (12,2,128) fill
print 'NUM' (13,19)
print 'FECHA' (13,26)
print 'FECHA' (13,97)
print 'FECHA' (13,107)
print 'DIAS' (13,116)
print 'SEM' (13,122)
print 'MES' (13,127)
print 'ALM' (14,3)
print 'PEDIDO' (0,+2)
print 'PARTS' (0,+4)
print 'ENTREGA' (0,+2)
print 'PROVEEDOR' (0,+14)
print 'DESCRIPCION GENERICA' (0,+#pl)
print 'CIERRE' (0,+6)
print 'EMB/REB' (0,+3)
print 'RET' (0,+3)
print 'RET' (0,+3)
print 'RET' (0,+2)
else (15,2,128) fill
print ' ' (11,2,128) fill !Si es la última hoja no imprime
print ' ' (12,2,128) fill !Area, Comprador, ni encabezado.
end-if
end-heading

```

```

|*****
|* Procedimiento que obtiene las fechas del período del *
|* cual se desea el reporte de las requisiciones. *
|*****

```

```
begin-procedure obtener periodo
input $fecha1 maxlen=6 noprompt status=#st1 type=integer
input $fecha2 maxlen=6 noprompt status=#st2 type=integer
extract $dial from $fecha1 4 2
extract $mes1 from $fecha1 2 2
extract $año1 from $fecha1 0 2
extract $dia2 from $fecha2 4 2
extract $mes2 from $fecha2 2 2
extract $año2 from $fecha2 0 2
string $dial $mes1 $año1 by '/' into $fec1
string $dia2 $mes2 $año2 by '/' into $fec2
let $periodo='en el periodo comprendido del '||$fec1||' al '||$fec2
end-procedure
```

!* Procedimiento que imprime las Requisiciones Colocadas *
!* correspondientes a cada comprador. <Principal> *

```
begin-procedure imp ped coloc
create-array name=resum size=25 !Inicializa Arreglo para el
field=ar_cve comp:char !resumen Final.
field=ar_tot_dias:number
field=ar_promd mov:number
field=ar_sin_ret:number
field=ar_ped_ret:number
field=ar_promd ret:number
!Inicializa variables para el resumen por comprador.
let $tipo_ped = 'PEDIDOS CERRADOS'
move 0 to #n ! Inicializa Índice para el arreglo
move 0 to #nmovs ! Número de movimientos/comprador
move 0 to #ndias ! Total de días/comprador
move 0 to #promd mov ! Prom. días/movimiento
move 0 to #tot_ped_ret ! Total pedidos retrasados
move 0 to #tot_sin_ret ! Total pedidos sin retraso
move 0 to #promd ret ! Promedio de días retrasados
move 0 to #ultima pag ! Var. para indicar que es la última hoja
move 0 to #combcap ! Indica cuantos son del Área Bienes de Capital
move 16 to #pl
move 28 to #long
begin-select ! Imprime Requisición.
pedido.com_clave (0,0) on-break after=imp resum print=never save=$com ant
tde_clave,Eda_descrl,comprador.com_clave,com_nombre,prv_rfc,prv_cnd_entre
alm_clave (+1,3,4)
ped_clave (0,+1,5) edit 99999
ped_num_conse (0,+2,2) edit 99
do obtén partidas($alm_clave,&ped_clave,&ped_num_conse,#nptdas)
print #nptdas (0,+2,3) edit 999 ! Num. Partidas/Req.
ped fec entre (0,+3,8) edit DD/MM/YY
do obtén desyrzo($alm_clave,&ped_clave,&prv_rfc,$desc_gen,$razon_soc)
print $razon_soc (0,+2,#long)
print $desc_gen (0,+2,29) wrap 29 1 STRIP=<13> ON=<10>
ped fec statū (0,+2,8) edit DD/MM/YY
move 1 to #es LAB
if $prv_cnd_entre != 'LAB'
let $mov='EM'
do obtén fec EmRe($alm_clave,&ped_clave,&ped_num_conse,$mov,&ped_fec_entre,$fec
if not !null($fec EmRe)
move 0 to #es LAB
end-if
end-if
if #es LAB = 1
let $mov = 'RA'
do obtén fec EmRe($alm_clave,&ped_clave,&ped_num_conse,$mov,&ped_fec_entre,$fec
end-if
print $fec EmRe (0,+2,8) edit DD/MM/YY
if #wk_dia_trans <= 0
print '0' (0,+2,4) edit 9999
add 1 to #tot_sin_ret !Cuenta los pedidos sin
else !retraso.
add #wk_dia_trans to #ndias !Suma los días transcurridos.
add 1 to #tot_ped_ret !Cuenta los pedidos con retraso
print #wk_dia_trans (0,+2,4) edit 9999
```

```

end-if
print #sem_ret          (0,+2,3) edit 999 ! Semanas retraso
print #mes_ret         (0,+2,3) edit 999 ! Meses retraso
add 1 to #nmovs                ! Cuenta movimientos
move &tde clave to %cve area
move &tde descri to %de%c
move &comprador.com clave to %cve_comp
move &com nombre to %nom_comp
from tabla_detalle_pedido,comprador
where pedido.com clave=comprador.com clave      ! Compara Cve. Comp.
and com_area=tde clave                          ! Compara Area
and ped_status = 'CE'
and convert(char(6),ped_fec_statu,12) between %fecha and %fecha2
and tab num conse=18
order by pedido.com clave,pedido.alm_clave,ped_clave,ped_num_conse ! Comprador
end-select
move 1 to #ultima_pag          !Indica que se imprimira el
do resumen final              !Resumen Final.
end-procedure

```

```

|*****
!* Procedimiento que obtiene el n°mero de partidas por pedido *
|*****

```

```

begin-procedure obten_partidas(%almacen,%pedido,%ped_cons,%nptdas)
begin-select
count(*) %nptdas
move %nptdas to #nptdas
from partida_pedido
where alm_clave = %almacen
and ped_clave = %pedido
and ped_num_conse = %ped_cons
end-select
end-procedure

```

```

|*****
!* Procedimiento que obtiene la descripciun genérica del pedido y la *
!* razon social del proveedor. *
|*****

```

```

begin-procedure obten_desyzzo(%almacen,%pedido,%rfc_prov,%desc_gen,%razon_soc)

```

```

begin-select                                !Obtiene la descripciun generica de
req_des gener                                !la tabla de requisiciones.
move %req_des gener to %desc_gen
from requisición
where alm_clave=%almacen
and req_num_conse=%pedido
end-select

```

```

begin-select                                !Obtiene la razon social del proveedor
prv_rzo socia                                !de la tabla de proveedores.
move %prv_rzo_socia to %razon_soc
from proveedor
where prv_rfc=%rfc_prov
end-select

```

```

end-procedure

```

```

|*****
!* Procedimiento que obtiene la fecha de Embarque/Recepciun *
!* para cada pedido. Llamado por "imp_ped_colcan" *
|*****

```

```

begin-procedure obten_fec_EmRe(%almacen,%pedido,%ped_cons,%mov,%fech_ent,
:%fec_EmRe,%%wk_dia_trans,%sem_ret,%mes_ret)

```

```

begin-select
max(mov fecha) %fec EmRe
datediff(day,%fech_ent,max(mov_fecha)) %wk_dia_trans
move %fec EmRe to %fec EmRe
move %wk_dia_trans to %wk_dia_trans
do sem tra ped(%fech_ent,%fec_EmRe,%sem_ret)
do mes tra ped(%fech_ent,%fec_EmRe,%mes_ret)
from movimiento_partida_pedido mpp, movimiento_material mm
where mpp.mov_tipo = %mov

```

```

and alm clave = Salmacen
and ped_Clave = #pedido
and ped_num_conse = #ped_cons
and nm.mov_num_conse = mpp.mov_num_conse
end-select

```

```
end-procedure
```

```
! Procedimiento que calcula la antigüedad en Semanas y Meses para los Pedidos.
#include '/sist01/logyaba/abaste/reportes/include/abanti02_1.inc'
```

```

!*****
!* Procedimiento que calcula e imprime el resumen de movimientos *
!* para cada comprador. Llamado por "imp_ped_colcan" *
!*****

```

```

begin-procedure imp_resum
if #nmovs > 0 ! Solo si hubo movimientos
let #promd_mov = round(#ndias/#nmovs,0) ! calcula los promedios.
else
move 0 to #promd_mov
end-if
if #tot_ped_ret > 0
let #promd_ret = round(#ndias/#tot_ped_ret,0)
else
move 0 to #promd_ret
end-if
! Coloca los datos del n-esimo comprador en el arreglo "resum".
add 1 to #n ! incrementa en 1 el índice
put $com_ant #ndias #promd_mov #tot_sin_ret #tot_ped_ret #promd_ret into resum(#n
if $cva_área='02' ! Checa si el comprador pertenece
add 1 to #combcap ! al área de Bienes de capital
end-if

```

```
! Imprime Resumen por comprador
```

```

print 'TOTAL PEDIDOS RETRASADOS:' (+2,3)
print #tot_ped_ret (0,36,5) edit 9,999
print 'PROMEDIO DIAS POR RETRASADOS:' (0,91)
print #promd_mov (0,125,5) edit 9,999
print 'TOTAL SIN RETRASO:' (+1,3)
print #tot_sin_ret (0,36,5) edit 9,999
print 'TOTAL DE MOVIMIENTOS:' (+1,3)
print #nmovs (0,36,5) edit 9,999
print 'TOTAL DE DIAS:' (0,57)
print #ndias (0,75,6) edit 99,999
print 'PROMEDIO DIAS POR MOVIMIENTO:' (0,91)
print #promd_mov (0,125,5) edit 9,999
move 0 to #ndias !Inicializa vars. para el
move 0 to #promd_mov
move 0 to #promd_ret
move 0 to #tot_ped_ret !sig. comprador.
move 0 to #tot_sin_ret
move 0 to #nmovs
new-page !Cambia de hoja
end-procedure

```

```

!*****
!* Procedimiento que Imprime el Resumen Final **
!* Llamado por "imp_ped_colcan". **
!*****

```

```

begin-procedure resumen_final
print 'CUMPLIDOS' (+1,28)
print 'NO CUMPLIDOS' (0,45)
print 'DIAS RETRASO' (0,67)
print 'PROMEDIO DIAS' (0,87)
print 'PROMEDIO GRAL.' (0,109)
print 'COMPRADOR' (+1,11)
print 'NO CUMPLIDOS' (0,87)
!print $linea (+1,11)
print '-' (+1,11,9) fill
print '-' (0,28,9) fill
print '-' (0,45,12) fill
print '-' (0,67,12) fill
print '-' (0,87,13) fill

```

```

print '-' (0,109,14) fill
!Inicializa variables para los totales.
move 0 to #tot1 ! Total pedidos cumplidos
move 0 to #tot2 ! Total pedidos no cumplidos
move 0 to #tot3 ! Total dias transcurridos
move 1 to #c
while #c <= #n
! Inicializa variables para los subtotales:
move 0 to #subtot1 ! Subtotal pedidos cumplidos
move 0 to #subtot2 ! Subtotal pedidos no cumplidos
move 0 to #subtot3 ! Subtotal dias transcurridos
while #c <= #combcap
get $dt_cve_comp $dt_tot_dias $dt_promd mov $dt_sin_ret $dt_ped_ret $dt_promd_r
print $dt_cve_comp (7,15,2) edit 99
print $dt_sin_ret (0,31,3) edit 999
print $dt_ped_ret (0,50,3) edit 999
print $dt_tot_dias (0,70,6) edit 99,999
print $dt_promd_ret (0,92,3) edit 999
print $dt_promd_mov (0,115,3) edit 999
add $dt_sin_ret to #subtot1
add $dt_ped_ret to #subtot2
add $dt_tot_dias to #subtot3
add 1 to #c
end-while
if #subtot3>0
let #subtot4=round(#subtot3/#subtot2,0)
let #subtot5=round(#subtot3/((#subtot1+#subtot2),0)
else
move 0 to #subtot4
move 0 to #subtot5
end-if
print '-' (+1,28,9) fill
print '-' (0,45,12) fill
print '-' (0,67,12) fill
print '-' (0,87,13) fill
print '-' (0,109,14) fill
print 'SUBTOTAL' (+1,12)
print #subtot1 (0,31,3) edit 999
print #subtot2 (0,50,3) edit 999
print #subtot3 (0,70,6) edit 99,999
print #subtot4 (0,92,3) edit 999
print #subtot5 (0,115,3) edit 999
add #subtot1 to #tot1
add #subtot2 to #tot2
add #subtot3 to #tot3
print ' ' (+1,2)
move #n to #combcap
end-while
if #tot3>0
let #tot4=round(#tot3/#tot2,0)
let #tot5=round(#tot3/((#tot1+#tot2),0)
else
move 0 to #tot4
move 0 to #tot5
end-if
print '-' (+1,28,9) fill
print '-' (0,45,12) fill
print '-' (0,67,12) fill
print '-' (0,87,13) fill
print '-' (0,109,14) fill
print 'TOTALES' (+1,12)
print #tot1 (0,31,3) edit 999
print #tot2 (0,50,3) edit 999
print #tot3 (0,69,7) edit 999,999
print #tot4 (0,92,3) edit 999
print #tot5 (0,115,3) edit 999
end-procedure

```

```

*****
!*      Reporte de la Solicitud de Transporte      *
!*      ordenada por pedido que ampara           *
!*      *                                           *
!* Realizado por: Sistemas Log. y Aba.           *
!* Ultima actualizacion: 28/09/93                *
*****

```

```

begin-report
do obten_datos
do imprime_datos
end-report --

```

```

begin-setup
page-size 84 84
end-setup

```

```

begin-heading 12
print 'abmmsolt01' (1,1)
print 'Pagina' (0,73)
page-number (0,81)
print $fb (7,6)
print 'SERVS INDUSTRIALES PEÑALES, SA DE CV' ( )
print $fl ( )
print 'Dirección Logística y Abastecimiento' (9) center
print 'Subdirección de Logística' (10) center
print 'Solicitud de Transporte' (11) center
date-time (0,64,17)
print ' ' (12,1,83) fill
end-heading

```

```

begin-procedure obten_datos
encode '<27>&k8S' into $fb
encode '<27>&kOS' into $fl
input $numsol maxlen=10 'No.Sol' type=char
do obten_datos2
end-procedure

```

```

begin-procedure obten_datos2
create-array name=pedido size=6
      field=ar_alm_cve:char
      field=ar_ped_cve:number
      field=ar_ped_num_conse:number

```

```

move 0 to $fl
begin-select distinct
st.sot_num_conse &sot_num_conse
st.com_clave &com_clave
st.agt_clave &agt_clave
st.tra_clave &tra_clave
st.dim_largo &st_dim_largo
st.dim_ancho &st_dim_ancho
st.dim_alto &st_dim_alto
st.ume_solic &st_ume_solic
st.tip_mecia &st_tip_mecia
st.dom_agent &st_dom_agent
st.rem_horar &st_rem_horar
st.des_horar &st_des_horar
st.notas &st_notas
st.lin_trans &st_lin_trans
st.gui_talon &st_gui_talon
st.tra_placa &st_tra_placa
st.num_econo &st_num_econo
st.caj_numero &st_caj_numero
convert (char(8),st_fec_embar,11) &st_fec_embar
convert (char(8),st_fec_cruce,11) &st_fec_cruce
convert (char(8),st_fec_coloc,11) &st_fec_coloc
st.operador &st_operador
com_nombre &com_nombre

```

```

stp.alm clave &alm clave
  put &alm clave into pedido(#i) ar_alm_cve
stp.ped clave &ped clave
  put &ped clave into pedido(#i) ar_ped_cve
stp.ped num conse &ped_num_conse
  put &ped_num_conse into pedido(#i) ar_ped_num_conse
getdate() &getdate
add 1 to #i
  from solicitud transporte st,comprador com,solicitud_transporte_pedido stp
where st.sot_num conse = convert(int, $numsol)
  and stp.sot_num conse = st.sot_num conse
  and st.com_clave = com.com_clave
end-select

```

```

begin-select
bul_cantidad
bul_descripcion
bul_peso
ume_clave
  agt_nombre
  agt_calle
  agt_colonia
  agt_estado
  agt_pais
  agt_cod_posta
  agt_atencion
  from buito bl,agente_aduanal aa
  where aa.agt_clave = &agt_clave
  and bl.agt_clave = aa.agt_clave
  and bl.tra_clave = &tra_clave
end-select

```

```

begin-select
prv_nacionali &prv_nacionali
  from proveedor pr,pedido pd
  where pd.alm_clave = &alm_clave
  and pd.ped_clave = &ped_clave
  and pd.ped_num_conse = &ped_num_conse
  and pd.prv_rfc = pf.prv_rfc
end-select

```

```

begin-select
cia_nombre &cia_nombre
cia_dom_fisca &cia_dom_fisca
  from compania cm,unidad un,almacen al
  where al.alm_clave = &alm_clave
  and al.uni_clave = un.uni_clave
  and un.cia_clave = cm.cia_clave
end-select

```

```

begin-select
dom_nombre
dom_tel
alm_nom_respo
  from domicilio dm,almacen al,almacen_domicilio ad
  where al.alm_clave = &alm_clave
  and al.alm_clave = ad.alm_clave
  and ad.ad_clave = dm.dom_clave
end-select
end-procedure

```

```

begin-procedure imprime_datos
move 0 to #i
print 'Sol.Num' (+5,1)
print 'Sot.num conse' (0,15)
print 'Solicitante' (0,35)
print 'Com.clave' (0,+2)
print 'Com.nombre' (0,+2)
print 'Tráfico' (+8,1)
print 'Agnt.clave' (0,15)
print 'Tra.clave' (0,19)
print 'Pedido' (+2,1)
while #i < 6
  get &almclave from pedido(#i) ar_alm_cve
  print &almclave (0,+8,3)

```

```

get #pedclave from pedido(#i) ar_ped_cve
if #pedclave != 0
  print #pedclave (0,+2) edit 99999
end-if
get #pedconse from pedido(#i) ar_ped_num_conse
if #pedclave != 0
  print #pedconse (0,+2) edit 99
end-if
add 1 to #i
end-while
print 'Origen' (+2,1)
print &prv nacionali (0,15)
print 'Tipō Mcia' (0,62)
print &got tip_mcia (0,+2)
print 'Empaque' (+2,1)
print &bul cantidad (0,15)
print &bul_descripcion (0,19,30)
print 'Peso' (0,67)
print &bul_peso (0,73)
print &ume clave (0,82)
print 'Largo' (+2,1)
print &got dim_largo (0,15)
print 'Ancho' (0,19)
print &got dim_ancho (0,24)
print 'Alto' (0,42)
print &got dim_alto (0,+2)
print 'U.Mēdidā' (0,63)
print &ot_ume_solic (0,+2)
print 'Remitenē' (+6,1)
print &gt; nombre (0,15)
print 'DomIcilio' (+2,1)
print &gt; calle (0,15,35)
print 'Colōnia' (+2,1)
print &gt; colonia (0,15,19)
print 'Estado' (0,40)
print &gt; estado (0,+2,4)
print 'Paīs' (0,67)
print &gt; pais (0,+2,4)
print 'Cod.Postal' (+2,1)
print &gt; cod_posta (0,15)
print 'Recōger en' (0,36)
print &ot_dom_agent (0,+2,25)
print 'Hor Carġa' (0,62)
print &ot_rem_horar (0,+2)
print 'Atenciōn' (+2,1)
print &gt; atencion (0,15,20)
print 'DesEinatorio' (+6,1)
print &cia nombre (0,15)
print 'Entregar en' (+2,1)
print &dom nombre (0,15,30)
print 'Tel' (+2,1)
print &dom_tel (0,15)
print 'Ext' (0,43)
print 'Hor descarga' (0,59)
print &ot_des_horar (0,+2)
print 'Atenciōn' (+2,1)
print &alm_nom_respo (0,15,30)
print 'Observaciones' (+2,1)
print &ot_notas (0,15)
print 'Linēa Transp' (+6,1)
print &ot_lin_trans (0,15,40)
print 'Talōn' (0,66)
print &ot_gul_talon (0,+2)
print 'Placa TFactor' (+2,1)
print &ot_tra_placa (0,15)
print 'Eco' (0,43)
print &ot_num_econo (0,+2)
print 'Cajā num' (0,63)
print &ot_caj_numero (0,+2)
print 'Fec Embār' (+2,1)
print &ot_fec_embar (0,15)
print 'Fec Cruċe' (0,37)
print &ot_fec_cruce (0,+2)
print 'Fec Colōc' (0,62)
print &ot_fec_coloc (0,+2)
print 'OpeFadoĠ' (+2,1)

```

```
print &not_operator (0,15)
end-procedure
```

```

|*****
|** Programa que hace una Tabla Comparativa por Requisiciu:n **
|** y una Tabla Comparativa por Proveedor **
|** Realizado por: Sistemas Logística y Abastecimiento (GCN) **
|** Fecha de elaboraci:u:n: 26/02/93 **
|*****

begin-report
do principal
end-report

begin-setup
page-size 75 132
end-setup

begin-heading 8
print 'FECHA DE PROCESO: ' (1,2)
date-time (0,20) DD/MM/YY
page-number (1,119) 'HOJA: '
print $nom rep (2,2) center
print 'SERVICIOS INDUSTRIALES PEIOLES, S.A. DE C.V.' (2) center
print 'Direcci:u:n de Logística y Abastecimiento' (3) center
print 'Subdirecci:u:n de Abastecimiento' (4) center
print $titulo (5) center
print 'Almac:u:n:' (6,4)
print $xalmacen (0,13)
print $nom alm (0,17)
print 'Comprador:' (0,89)
print $xcomp (0,100)
print $xcomprador (0,103,26)
print 'Requisici:u:n:' (7,4)
print $xrequisicion (0,17) edit 99999
print ' ' (8,2,128) fill
end-heading

|*****
|** Procedimiento que obtiene los datos correspondientes a cada **
|** proveedor que cotiza la requisici:u:n. **
|*****

begin-procedure obten_provs

! Se crea el arreglo "PROVS", el cual almacenar:u: informaci:u:n sobre
! los proveedores.
create-array name=provs size=10
field=ar_prv_rfc:char ! RFC Proveedor
field=ar_rzo_soc:char ! Raz:u:n Social Proveedor
field=ar_cve_cot:char ! Clave de Cotizaci:u:n.
field=ar_fec_cot:char ! Fecha vigencia cotizaci:u:n
field=ar_tipo_mon:char ! Tipo Moneda
field=ar_tipo_camb:number ! Tipo Cambio

! Se define el arreglo "COTIZ" para almacenar cotizaci:u:n de las partidas
create-array name=cotiz size=10
field=ar_pje_dsto:number ! Porcentaje Descuento
field=ar_descuento:number ! Descuento
field=ar_conv_pu:number ! Conversi:u:n precio unitario
field=ar_conv_pt:number ! Conversi:u:n precio total
field=ar_conv_dsto:number ! Conversi:u:n descuento
field=ar_cot_prov:number ! Cotizaci:u:n por Proveedor
field=ar_des_prov:number ! Descuento por Proveedor
field=ar_tot_cot:number ! Total cotizaci:u:n
field=ar_tot_reco:number ! Total cotizaci:u:n econ:mica

! Se define el arreglo "COT_ECON" para almacenar la cotizaci:u:n m:u:lti
! econ:mica por partida.
create-array name=cot_econ size=100 ! 100 partidas por requisici:u:n
field=ar_pu_moncot:number:10 ! Precio unitario mon. cot.

```

```

field=ar_pt_moncot:number:10 | Precio total mon. cot.
field=ar_pja_des:number:10 | Porcentaje de descuento
field=ar_pt_fondes:number:10 | Precio total con descuento
field=ar_conv_pu:number:10 | Conversi,un precio unitario
field=ar_conv_pt:number:10 | Conversi,un precio total
field=ar_conv_ptcd:number:10 | Conv. pre.tot. c/descuento
field=ar_mas_econ:number:3 | Cotizacion mAs economica
field=ar_cot_selec:number:10 | Cotizaci,un seleccionada

```

```

move 0 to #nprov
move 'f' to $si_esta
begin-select
partida cotizacion.prv_rfc,prv_rzo_socia,partida_cotizacion.cot_clave,cot_fec_limit
convert(char(12),cot_fec_cotiz,3) &wk_fecha_cot
if #nprov > 0
do chk_prove
! 50 ya hay proveedores en el arreglo
! checa si el proximo ya estA.
end-if
if $si_esta = 'f'
add 1 to #nprov
if &cotizacion.mon_clave != 'MN'
do obten_tip_camb
end-if
put &partida_cotizacion.prv_rfc &prv_rzo_socia &partida_cotizacion.cot_clave &
move 0 to #tip_cam
else
move 'f' to $si_esta
end-if
from partida_cotizacion,proveedor,cotizacion
where alm_clave = $xalmacan
and convert(varchar(5),req_num_conse) = $xrequisicion
and req_num_conse = #xrequisicion
and proveedor.prv_rfc = partida_cotizacion.prv_rfc
and cotizacion.prv_rfc = partida_cotizacion.prv_rfc
and cotizacion.cot_clave = partida_cotizacion.cot_clave
order by prq_num_conse, partida_cotizacion.prv_rfc, partida_cotizacion.cot_clave
end-select
show (1,1) clear-line
if #nprov = 0
show clear-screen (23,1) bold reverse '>' No hay partidas cotizadas para esta re
input $wait type=char noprompt
stop
end-if
do imp_prov
end-procedure

```

```

|*****
|** Procedimiento que checa que los proveedores no se repitan **
|** A momento de meterlos al arreglo. **
|*****

```

```

begin-procedure chk_prove
move 1 to #prv
while #prv <= #nprov
get $rfc $cve coti from provs(#prv) ar_prv_rfc ar_cve_cot
if &partida_cotizacion.prv_rfc = $rfc and
&partida_cotizacion.cot_clave = $cve_coti
move 't' to $si_esta
let #prv = #nprov + 1
else
add 1 to #prv
end-if
end-while
end-procedure

```

```

|*****
|** Procedimiento que obtiene el tipo de cambio para la moneda **
|** en la que se cotiza la requisici,un. **
|*****

```

```

begin-procedure obten_tip_camb
begin-select
tca tip_cambi
move &Eca_tip_cambi to #tip_cam
from tipo_cambio

```

```

where mon_clave = 'US'
and convert(char(12),tca_fec_cotiz,3) = &wk_fecha_cot
end-select

begin-select
tca_tip_cambi &tc otra_mon
move &cotizacion.mon_clave to $moneda
if $moneda != 'US'
let #tip_cam = trunc((#tip_cam * &tc_otra_mon),4)
end-if
from tipo_cambio
where tipo_cambio.mon_clave = &cotizacion.mon_clave
and convert(char(12),tca_fec_cotiz,3) = &wk_fecha_cot
end-select

```

```
end-procedure
```

```

!*****
!** Procedimiento que pone las leyendas para los proveedores **
!*****

```

```

begin-procedure Pon_leyendas
print 'PROVEEDOR:' (1,23) | Imprime leyendas
print 'CLAVE DE COTIZACION:' (2,13)
print 'FECHA VIGENCIA COTIZACION:' (3,7)
print 'MONEDA/TIPO CAMBIO:' (4,14)
print ' ' (+1,2,128) fill
print 'PARTIDA CANTIDAD UNIDAD' (+1,2)
print 'DESCRIPCION' (+1,12)
print ' ' (+1,2,31) fill
end-procedure

```

```

!*****
!** Procedimiento que imprime el resumen de las cotizaciones economicas por **
!** proveedor. Llamado por "Imp_pcot_eco" **
!*****

```

```

begin-procedure imp_coeco_prov
move 1 to #rc
print ' ' (+1,2,128) fill
while #rc <= 2
if #rc = 1
print 'ALTERNATIVA ECONOMICA POR PROVEEDOR' (+2,2)
end-if
move 1 to #rcp
while #rcp <= 3
if #rcp = 1
move 51 to #col
let #c = #col - 3
end-if
if #rcp = 2
move 83 to #col
let #c = #col - 3
end-if
if #rcp = 3
move 115 to #col
let #c = #col - 3
end-if
get #tot_coeco_prov from cotiz(#rcp) ar_tot_ceco
if #rc = 1
if #tot_coeco_prov != 0
let #cotiz_ekon_tot = #cotiz_ekon_tot + #tot_coeco_prov
print 'NS' (0,#c)
print #tot_coeco_prov (0,#col,14) edit 999,999,999.99
put 0 into cotiz(#rcp) ar_tot_ceco
end-if
end-if
if #rc = 2 | No funciona en este procedimiento
if #rcp <= #nprov
if #tot_coeco_prov != 0
print ' ' (0,#col,14) fill
end-if
end-if
end-if
add 1 to #rcp
end-while

```

```

add 1 to #rc
end-while
if #np = #nprov
print 'TOTAL ALTERNATIVA ECONOMICA' (+1,2,128) fill
print '<<<<<' (+2,2)
print 'NS' (0,42)
print #cotiz_econ_tot (0,48)
print '>>>>>' (0,51,14) edit 999,999,999.99
end-if
end-procedure

```

```

|*****
|** Procedimiento que imprime el resumen de las cotizaciones por proveedor.**
|** Llamado por "imp_part_cot"
|*****

```

```

begin-procedure imp_coti_prov
move 1 to #rc
print ' ' (+1,2,128) fill
while #rc <= 4
if #rc = 1
print 'COTIZACION POR PROVEEDOR' (+2,2)
end-if
if #rc = 2
print 'DESCUENTO POR PROVEEDOR' (+1,2)
end-if
if #rc = 3
print ' ' (+1)
end-if
if #rc = 4
print 'TOTAL COTIZACION' (+2,2)
end-if
move 1 to #rcp
while #rcp <= 3
if #rcp = 1
move 51 to #col
end-if
if #rcp = 2
move 83 to #col
end-if
if #rcp = 3
move 115 to #col
end-if
get #tot_cot_prov #tot_des_prov #tot_cotiz from cotiz(#rcp) ar_cot_prov ar_des_
if #rc = 1
if #tot_cot_prov
let #tot_cotiz = #tot_cot_prov - #tot_des_prov
put #tot_cotiz into cotiz(#rcp) ar_tot_cot
print #tot_cotiz (0,#col,14) edit 999,999,999.99
put 0 into cotiz(#rcp) ar_cot_prov
end-if
end-if
if #rc = 2
if #tot_des_prov != 0
print #tot_des_prov (0,#col,14) edit 999,999,999.99
put 0 into cotiz(#rcp) ar_des_prov
end-if
end-if
if #rc = 3
if #rcp <= #nprov
print ' ' (0,#col,14) fill
end-if
end-if
if #rc = 4
if #tot_cotiz != 0
let #c = #col - 3
print 'NS' (0,#c)
print #tot_cotiz (0,#col,14) edit 999,999,999.99
put 0 into cotiz(#rcp) ar_tot_cot
end-if
end-if
add 1 to #rcp
end-while
add 1 to #rc

```

```
end-while
end-procedure
```

```
!*****
!** Procedimiento que imprime la conversio del descuento **
!** a moneda nacional . Llamado por "imp convera" . **
!*****
```

```
begin-procedure imp_conv_des
```

```
move 1 to #ncon
```

```
print ' ' (+1)
```

```
while #ncon <= #prov_pag
```

```
if #ncon = 1
```

```
move 35 to #c1
```

```
move 51 to #c2
```

```
end-if
```

```
if #ncon = 2
```

```
move 67 to #c1
```

```
move 83 to #c2
```

```
end-if
```

```
if #ncon = 3
```

```
move 99 to #c1
```

```
move 115 to #c2
```

```
end-if
```

```
get #conv_des from cotiz(#ncon) ar_conv_dsto
```

```
if #conv_des != 0
```

```
lprint #conv_pu (0,#c1,14) edit 999,999,999.99
```

```
print #conv_des (0,#c2,14) edit 999,999,999.99
```

```
end-if
```

```
add 1 to #ncon
```

```
end-while
```

```
end-procedure
```

```
!*****
!** Procedimiento que imprime las conversiones del precio unitario **
!** y del precio total a moneda nacional. Llamado por "imp desctos" **
!*****
```

```
begin-procedure imp_convera
```

```
move 1 to #ncon
```

```
print ' ' (+1)
```

```
while #ncon <= #prov_pag
```

```
if #ncon = 1
```

```
move 35 to #c1
```

```
move 51 to #c2
```

```
end-if
```

```
if #ncon = 2
```

```
move 67 to #c1
```

```
move 83 to #c2
```

```
end-if
```

```
if #ncon = 3
```

```
move 99 to #c1
```

```
move 115 to #c2
```

```
end-if
```

```
get #conv_pu #conv_pt from cotiz(#ncon) ar_conv_pu ar_conv_pt
```

```
if #conv_pu != 0
```

```
print #conv_pu (0,#c1,14) edit 999,999,999.99
```

```
print #conv_pt (0,#c2,14) edit 999,999,999.99
```

```
end-if
```

```
add 1 to #ncon
```

```
end-while
```

```
do imp_conv_des
```

```
end-procedure
```

```
!*****
!** Procedimiento que imprime los descuentos **
!** Llamado por "imp part cot" **
!*****
```

```
begin-procedure imp_desctos
```

```
move 1 to #nd
```

```
print ' ' (+1)
```

```
while #nd <= #prov_pag
```

```
if #nd = 1
```

```
move 47 to #c3
```

```
move 51 to #c4
```

```

end-if
if #nd = 2
  move 79 to #c3
  move 83 to #c4
end-if
if #nd = 3
  move 111 to #c3
  move 115 to #c4
end-if
get #pje des #desc from cotiz(#nd) ar_pje_dsto ar_descuento
if #pje des != 0
  print #pje des      (0,#c3,2) edit 99
  print #desc        (0,#c4,14) edit 999,999,999.99
end-if
add 1 to #nd
end-while
do imp_convers
end-procedure

```

```

|*****
|** Procedimiento que imprime la cotizaci#n m#s econ#mica por **
|** partida. (Llamado por "imp_prov")
|*****

```

```

begin-procedure imp_pcot_eco
move 0 to #part_req_ant
move 0 to #num_cot_ant
move 0 to #non_tra#
move 1 to #prov_pag
begin-select
pco_pre unita &pre unita
pco_ptj_dseto &pje_des
if #npart != #part_req_ant
  if #part_req_antE != 0
    print ' '(+1)
  end-if
  if #prov_pag > 3
    move 3 to #prov_pag
  end-if
prg_num conse &npart      (+1,5)
pco_cantidad &cant      (0,13,9) edit 9,999,999 ! Num. partida
ume_clave &unidad      (0,29,2) ! Cantidad
des_nombre &descrip    (+1,2,31) ! Unidad Medida
! Descripci#n
move &npart to #part_req_ant
move &npart to #npart
move 1 to #prov_pag
move #ncot to #cp
end-if
while #prov_pag <= 3
get #pu #pt #pd #ptcd #cpu #cpt #cptcd #cot_eco from cot_econ(#npart) ar_pu_monco
if #cot_eco = 1
  if #prov_pag = 1
    move 35 to #c1
    move 51 to #c2
  end-if
  if #prov_pag = 2
    move 57 to #c1
    move 83 to #c2
  end-if
  if #prov_pag = 3
    move 99 to #c1
    move 115 to #c2
  end-if
  let #c=#c1+12
  print #pu      (0,#c1,14) edit 999,999,999.99
  print #pt      (0,#c2,14) edit 999,999,999.99
  print #pd      (+1,#c,2) edit 99
  print #ptcd    (0,#c2,14) edit 999,999,999.99
  if #cpu != 0
    print #cpu    (+1,#c1,14) edit 999,999,999.99
    print #cpt    (0,#c2,14) edit 999,999,999.99
    if #pd != 0
      print #cptcd (+1,#c2,14) edit 999,999,999.99
    end-if
    array-add #cptcd to cotiz(#prov_pag) ar_tot_ceco

```

```

else
  array-add #ptcd to cotiz(#prov_pag) ar_tot_ceco
end-if
move 4 to #prov_pag
else
  add 1 to #prov_pag
  add 1 to #cp
end-if
end-while
from partida cotizacion,descr_temporal
where alm clave = $xalmacen
and convert(varchar(5),req_num conse) = $xrequisicion
and req_num conse = $xrequisicion
and descr_temporal.des_clave = partida_cotizacion.des_clave
and des_num_conse = 1
order by prq_num_conse, prv_rfc, cot_clave
end-select
do imp_coeo procv          !Imprime resumen de cotizaciun economica por proveedor
end-procedure

!*****
!** Procedimiento que imprime las partidas de la requisiciun **
!** cotizadas por cada proveedor. (Llamado por "imp_prov") **
!*****

begin-procedure imp_part_cot
move 999999999999 to #cot_mas_eco
move 0 to #part_req_ant
move 0 to #num_cot_ant
move 0 to #son_tres
move 0 to #cotizo_part
move 0 to #px
move 1 to #prov_pag
begin-select
prv_rfc,cot_clave,pco_pre_unita,pco_ptj_dacto
if #prq_num_conse != #part_req_ant
  if #ya_paso = 1
    move 4 to #nl
    get #cme from cot_econ(#px) ar_mas_econ(1)
    if #cot_mas_eco < #cma
      put #cot_mas_eco #prv_eco into cot_econ(#px) ar_mas_econ(1) ar_mas_econ(
    end-if
  else
    put #cot_mas_eco #prv_eco into cot_econ(#px) ar_mas_econ(1) ar_mas_econ(2)
    move 1 to #nl
  end-if
if #prov_pag > 3
  move 3 to #prov_pag
end-if
if #part_req_ant != 0
  if #son_tres = 0
    let #prov_pag = #prov_pag - 1
    do imp_desctos
  end-if
end-if
prq_num_conse          (+1,5)          ! Num. partida
pco_cantidad &cantidad (0,13,9) edit 9,999,999 ! Cantidad
ume_clave             (0,29,2)          ! Unidad Medida
des_nombre            (+1,2,31)         ! Descripciun
if #prov_pag > #num_cot_ant
  move #prov_pag to #num_cot_ant
end-if
move #prq_num_conse to #part_req_ant
move #prq_num_conse to #px
move 1 to #prov_pag
move 0 to #son_tres
move 999999999999 to #cot_mas_eco
end-if
move 0 to #cotizo
move 1 to #cont
move #ncot to #cp
while #cont<=3 and #cotizo=0
  get #rfc #cve coti #mon #tipo cambio from provs(#cp) ar_prv_rfc ar_cve_cot ar_tipo
  if #prv_rfc =-#rfc and #cot_clave = #cve_coti
    move 1 to #cotizo
  else

```

```

add 1 to #cont
add 1 to #cp
end-if
end-while
if #cont <= 3
  if #cont > #prov_pag
    move 0 to #dscto
    move 0 to #conv_pu
    move 0 to #conv_pt
    move 0 to #conv_des
    put 0 #dscto #conv_pu #conv_pt #conv_des into cotiz(#prov_pag)
    move #cont to #prov_pag
    move #cp to #nl
    put #pre_uni #pre_tot #potje_des #pt_cdes #conv_pu #conv_pt #conv_ptcd into
  end-if
end-if
if #cotizo = 1
  move #pco_pre_unita to #pre_uni
  move #pco_ptj_dscto to #potje_des
  if #prov_pag = 1
    move 35 to #c1
    move 51 to #c2
  end-if
  if #prov_pag = 2
    move 57 to #c1
    move 83 to #c2
  end-if
  if #prov_pag = 3
    move 99 to #c1
    move 115 to #c2
  end-if
  if #prov_pag <= 3
    print #pco_pre_unita (0,#c1,14) edit 999,999,999.99
    let #pre_tot = trunc((#cantidad * #pco_pre_unita),2) |Precio Total
    let #dscto = trunc((#pre_tot * #pco_ptj_dscto)/100,2) |Descuento
    print #pre_tot (0,#c2,14) edit 999,999,999.99
    let #pt_cdes = trunc((#pre_tot-#dscto),2)
    if #mon_1 = 'MN'
      let #conv_pu = trunc((#pco_pre_unita * #tipo_cambio),2)
      let #conv_pt = trunc((#canElda# * #conv_pu),2)
      let #conv_des = trunc((#conv_pt * #pco_ptj_dscto)/100,2)
      let #conv_ptcd = trunc((#conv_pt-#conv_des),2)
      move #conv_ptcd to #ptcd
      array-add #conv_pt #conv_des to cotiz(#prov_pag) ar_cot_prov ar_des_prov
    else
      move #pt_cdes to #ptcd
      move 0 to #conv_pu
      move 0 to #conv_pt
      move 0 to #conv_des
      array-add #pre_tot #dscto to cotiz(#prov_pag) ar_cot_prov ar_des_prov
    end-if
    put #pco_ptj_dscto #dscto #conv_pu #conv_pt #conv_des into cotiz(#prov_pag)
  end-if
  if #ptcd < #cot_mas_eco
    move #ptcd to #cot_mas_eco
    move #nl to #prv_eco
  end-if
else
  move 0 to #pre_uni
  move 0 to #pre_tot
  move 0 to #potje_des
  move 0 to #dscto
  move 0 to #conv_pu
  move 0 to #conv_pt
  move 0 to #conv_des
  put 0 #dscto #conv_pu #conv_pt #conv_des into cotiz(#prov_pag)
end-if
if #prov_pag=3
  do imp_dscptos
  move 1 to #son_tres
end-if
if #ya_paso=1 and #cotizo=0
else
  add 1 to #prov_pag
  put #pre_uni #pre_tot #potje_des #pt_cdes #conv_pu #conv_pt #conv_ptcd into cot_
  add 1 to #nl

```

```

end-if
from partida_cotizacion,descr_temporal
where alm_clave = $xalmacen
and convert(varchar(5),req_num_conse) = $xrequisicion
!and req_num_conse = $xrequisicion
and descr_temporal.des_clave = partida_cotizacion.des_clave
and des_num_conse = 1
order by prq_num_conse, prv_rfc, cot_clave
and-select
if #ya_paso = 1
get %cme from cot_econ(#px) ar_mas_econ(1)
if #cot_mas_eco < %cme
put %cot_mas_eco #prv_eco into cot_econ(#px) ar_mas_econ(1) ar_mas_econ(2)
end-if
else
put %cot_mas_eco #prv_eco into cot_econ(#px) ar_mas_econ(1) ar_mas_econ(2)
end-if
if #prov_pag > 1 and #non_tres = 0
let #prov_pag = #prov_pag - 1
do imp_desctos
end-if
if #prov_pag > 3
move 3 to #prov_pag
end-if
move 1 to #ya_paso
do imp_coti_prov !Imprime resumen de cotizaciun por proveedor
end-procedure

!*****
!** Procedimiento que imprime los datos de cada proveedor que **
!** cotiza la requisiciun. **
!*****

begin-procedure imp_prov
move 1 to #n ! Contador de partidas/requisiciun
move 0 to #cotiz_econ_tot ! Cotizaciun economica total
move 0 to #ya_paso
move 1 to #prim_cot
move 0 to #menos_de_tres
move 1 to #xtabla
move 'abtabcom01' to $nom_rep
move 'TABLA COMPARATIVA POR REQUISICION' to $titulo
if #nprov < 3
move 1 to #menos_de_tres
end-if
while #xtabla <= 2
move 1 to #cont_cot
move 1 to #prv_pag
move 0 to #nc
move 1 to #np
do Pon_leyendas
while #np <= #nprov
if #prv_pag > 3
new-page
do Pon_leyendas
move 1 to #prv_pag
end-if
if #prv_pag = 1
move 38 to #col
end-if
if #prv_pag = 2
move 70 to #col
end-if
if #prv_pag = 3
move 102 to #col
end-if
get $rfc $rzo_soc $cve_coti $fec_vig_cot $moneda #tip_cam from provs(#np)
if $moneda != 'MH'
move #tip_cam to $tip_cam 9.9999
string $moneda $tip_cam by '/' into $mon_tip_cam
else
move $moneda to $mon_tip_cam
end-if
print $rzo_soc (1,#col,27)
print $cve_coti (2,#col,12)
print $fec_vig_cot (3,#col,8) edit dd/mm/yy

```

```

print $mon tip cam (4,#col,9)
print 'P. UNITARIO P. TOTAL' (+2,#col)
print ' % DESCTO. DESCTO.' (+1,#col)
print ' ' (+1,#col,27) fill
if #prv_pag = 3 or #menos_de_tres = 1
  if #xtabla = 1
    move #cont cot to #ncot
    do imp parE cot
    if #prov pag > #num cot ant
      move #prov_pag to #num_cot_ant
    end-if
    let #cont_cot = #cont_cot + #num_cot_ant
  else
    move #cont cot to #ncot
    do imp pcoE eco
    move 4 to #Cont_cot
  end-if
end-if
add 1 to #np
add 1 to #prv_pag
end-while
move 1 to #j
while #j <= #px
  move 1 to #k
  while #k <= #nprov
    get #pu #pt #pd #pcd #eco #ma from cot econ(#j) ar_pu_moncot(#k) ar_pt_moncot(#
    put 1 into cot_econ(#j) ar_cot_selec(#me)
    add 1 to #k
  end-while
  add 1 to #j
end-while
add 1 to #xtabla
new-report 'abtabcom02_1.lis'
move 'abtabcom02_1' to $nom_rep
move 'ALTERNATIVA ECONOMICA' to $titulo
end-while
end-procedure

|*****
|** Procedimiento que obtiene de la clave y el nombre del **
|** comprador, así como el nombre del almacén. **
|*****

begin-procedure obt_alm_comp

begin-select
requisición.com clave,com nombre,alm nombre
  move &requisición.com clave to $cve_comp
  move &com nombre to $xcomprador
  move &alm nombre to $nom_alm
from requisición,comprador,almacen
where requisición.alm clave = $xalmacen
and convert(varchar(5),req_num conse) = $xrequisición
and req_num conse = #xrequisición
and comprador.com clave = requisición.com clave
and almacen.alm clave = $xalmacen
end-select
if $cve_comp = ''
  move 1 to #err
end-if
end-procedure

|*****
|** Procedimiento que valida la clave del almacén y el **
|** número de requisición, así como también chequea que **
|** exista dicha requisición en la tabla "REQUISICION" **
|*****

begin-procedure valida
  move 0 to #err
if $xalmacen='' or $xrequisición=''
  move 1 to #xstat1
  show (23,1) clear-line
  show (23,1) reverse '='> Introduzca la clave del almacén y el número de requisici
  input $pausa noprompt type=char
else

```

```

if #xstat1 != 0 or #xstat2 != 0
show (23,1) clear-line
if #xstat1 != 0
if #xstat1=1
show (23,1) reverse '='> La clave del almacen fue mal proporcionada !!
input $wait type=char noprompt maxlen=1
end-if
if #xstat1=2
show (23,1) reverse '='> La clave del almacen rebasa la longitud permit
input $wait type=char noprompt maxlen=1
end-if
else
if #xstat2=1
show (23,1) reverse '='> El número de requisición debe ser numérico !!
input $wait type=char noprompt maxlen=1
end-if
if #xstat2=2
show (23,1) reverse '='> El número de requisición rebasa la longitud pe
input $wait type=char noprompt maxlen=1
end-if
end-if
else
do obt_alm comp                ! Busca y obtiene la clave y el nombre del
if #erf = 1                    ! comprador, así como el nombre del almacen
move 1 to #xstat1
show (23,1) clear-line
show (23,1) reverse bold '='> La requisición requerida no existe !!' beep
input $wait type=char noprompt maxlen=1
end-if
end-if
end-if
end-procedure

```

```

!*****
!** Procedimiento que captura la clave del almacen y el **
!** número de la requisición de la cual se desea obtener **
!** La Tabla Comparativa. **
!*****

```

```

begin-procedure captura_req
move 1 to #xstat1
move 1 to #xstat2
move 0 to #ok
!while #ok = 0
!show clear-screen (11,20) bold reverse 'Obtener Tabla Comparativa <S/N>?'
!input $opcion maxlen=1 noprompt type=char
!show normal
!uppercase $opcion
!if $opcion='N'
! show clear-screen normal
! stop
!end-if
!if $opcion='S'
!move 1 to #ok
!while #xstat1 != 0 or #xstat2 != 0
! move 7 to #ren
! show clear-screen normal
!show (7,14) bold '*****'
! show (9,14) bold '*****'
! show (13,14) bold '*****'
!while #ren < 14
! show (#ren,14) bold '*'
! show (#ren,69) bold '*'
! add 1 to #ren
!end-while
!show (8,15) reverse bold ' TABLA COMPARATIVA POR REQUISICION
!show (11,22) normal bold 'Almacen: ' reverse
input $xalmacen maxlen=4 status=#xstat1 type=char noprompt
!uppercase $xalmacen
!show (11,42) normal bold 'Requisición No: ' reverse
input $xrequisicion maxlen=5 status=#xstat2 type=integer noprompt
move $xrequisicion to #xrequisicion
do valida
!end-while
!end-if
!show clear-screen (1,5) bold blink 'Espere un momento ...' normal

```

```
! end-while  
end-procedure
```

```
|*****  
|** Procedimiento Principal **  
|*****
```

```
begin-procedure Principal  
do captura req  
do obten pfova  
end-procedure
```

```
! Captura el No. de requisicion  
! Obtiene datos de los proveedores
```

C.3. Programas en lenguaje "C"

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sybfront.h>
#include <sybdb.h>
#include <syberror.h>
#include "/users/logistica/fuentes_c/include/sybutil.h"
#include "/users/logistica/fuentes_c/include/printer.h"
#include "/users/logistica/fuentes_c/include/util.h"

/*****
Fecha: Ago 21 1992
Programa: abpeped01_1.c
Descripcion: Programa que genera el formato de impresion de los Pedidos.
*****/
char      uname[20], upass[20], server[20], mycommand[80], file_name[36],
file_name2[40], option[10], letr0[50], letr1[52], letr2[3][16],
letr3[8] = "PEÑALES", letr4[7] = "México", letr5[10] = "recepcion";

struct tm *today_date, *localtime();
time_t    time(), seconds;
size_t    strftime();
int  MAX_LINES_1 = 19, MAX_LINES_2 = 57;
int  cnt_page, cnt_line, limit_page,
par_not, f_not,
acuse;

char  date[33], y_o[4], dom_ofna_grl[10][35],
dom_corres[10][35], dom_factur[10][43], dom_consig[10][35],
des_parte[1000][37], dom_nom_esta[61], dom_nom_pais[61],
nom_vla_conse[61], nom_ond_pago[61], nom_pri[14],
com_nomBre[41], sppg_num_conse[3], sppg_cantidad[31],
nom_lmpto[17], ped_nom_autor[41], srag_num_conse[6],
srag_num_conse[3], des_not[1000][73], filier[81],
no_cnv[5], dom_fisca[10][43], partel[9],
paEte2[10], num_ample[5], tel_agt_adu[5],
cp_agt_adu[6], num_change[3], fec_eiaBo[9];

char  *des_nombre2;
DBPROCESS *sqlEmp, *sqlped, *sqlppe,
*sqlreq, *sqlpro, *sqlppr;
LOGINREC *login;
DBINT count_office;
FILE *file_output;

/**** Secuencias de control de impresion ****/
char  device[20];
int  copies;

unsigned char Q[4], W[4], A[4], S[4],
/* For Pc's .=218, .=191, .=192, .=217 */
/* For HP-9000 Q, W, A, S */
LV[4], LH[4],
/* For Pc's .=179, D=196 */
/* For HP-9000 ; */
LVGI[4], LVGD[4], LHGS[4], LHGI[4], CRS[4];
/* For Pc's .=180, .=195, .=194, .=193, .=197 */
/* For HP-9000 1, 2, 3, 4, 0 */

/**** Datos principales del Pedido ****/
DBCHAR alm_clave[7];
DBINT ped_clave;
DBTINYINT ped_num_conse;

/**** Totales en importes ****/
DBREAL sub_total, total_desc, total_imp,
total_1, tca_tip_cambi, total_nn,
total_partidas, tca_tip_cambi_ped, tca_tip_cambi_US,
sub_total_prog, total_partidas_prog;

```

```

/**** Datos de La oficina regional */
DBCHAR      ofi_clave[5],      ofi_nombre[41],      ofi_dom_regio[109],
            ofi_dom_corre[109];

/**** Datos del Almacén */
DBCHAR      alm_nombre[41],      alm_nom_respo[41],      cia_nombre[41],
            uni_clave[5],      ald_nombre_fac[256],      ald_nombre_con[256],
            cia_dom_fisca[109],      cia_nom_fisca2[109],      cia_nombre2[41];
DBSMALLINT  cia_clave;

/**** Datos del Pedido */
DBCHAR      prv_rfc[21],      prv_rfc_repre[21],      pdo_atencion[41],
            com_clave[3],      mon_clave[3],      prv_cnd_pago[3],
            prv_cnd_entre[4],      prv_cve_lugar_entre[3],      ped_nom_lugar_entre[36],
            ped_mat_orige[2],      agt_clave[3],      ped_tip_segur[3],
            ped_via_consi[3],      ped_per_autor[3],      ped_estado[3],
            com_ofi_clave[5],      nta_clave[15],      break_nta_clave[15],
            sprv_dia_condi_pago[4],      cot_clave[13],      dia_pront_pago[4];
DBTINYINT   pdo_num_conse,      prv_dia_pront_pago,      pdo_cve_domic_consi,
            ped_tpo_pedido,      prv_num_modif,      prv_dia_condi_pago,
            ped_cve_domic_factu;
DBSMALLINT  ped_num_conse;
DBREAL      ped_pje_impto,      ped_pje_dcto,      prv_dto_pront_pago,
            ped_comision;
DBREAL      ped_imp_total,      ped_imp_pagad;
DBCHAR      ped_fec_elabo[9],      ped_fec_entre[9],      ped_fec_ultma_repro[9],
            ped_fec_autor[9],      ped_fec_modif[9],      ped_fec_statu[9],
            cot_fec_cotiz[9];

/**** Datos de las partidas del Pedido */
DBCHAR      uma_unimed[6],      ppe_cve_prog[2],      mun_clave[13],
            mco_clave[15],      des_nombre[256];
DBREAL      ppe_cantidad,      ppe_pre_unita,      ppe_ptj_dcto;
DBINT       des_clave,      req_num_conse;
DBSMALLINT  ppe_num_conse,      prq_num_conse;

/**** Datos de las partidas programadas */
DBSMALLINT  emb_num_conse;
DBREAL      emb_canEldad,      emb_pre_unita;
DBCHAR      emb_fec_entre[9];

/**** Datos de la Requisición */
DBCHAR      req_aut_plant[3],      req_des_gener[256],      req_fec_elabo[9],
            req_fec_recep[9],      req_fec_asign[9],      req_fec_reque[9],
            req_tip_servi[2],      req_cod_prior[2],      req_nom_solic[26],
            req_ord_traba[7],      req_num_equip[13],      req_dep_nombre[21],
            req_folio[11],      req_mon_clave[3],      lin_sub_linea[7];
DBSMALLINT  alm_cta_cuent,      alm_cta_conce;
DBTINYINT   alm_cta_subct,      alm_cta_subc,      req_aut_perio,
            lin_clave;
DBINT       req_aut_folio,      req_num_emple;
DBREAL      req_imp_estim;

/**** Datos del Tipo de moneda */
DBCHAR      mon_nom_espan[21];

/**** Datos del Proveedor */
DBCHAR      prv_rzo_socia[41],      prv_rzo_socia_r[41],      prv_nacional[2],
            prv_teléfono[15],      prv_fax[15];

/**** Datos del Domicilio del proveedor u representante */
DBCHAR      pdo_calle[41],      pdo_colonia[41],      pdo_dlg_munic[41],
            pdo_estado[5],      pdo_pais[5],      pdo_cod_posta[14];

/**** Datos del Domicilio del Agente Aduanal */
DBCHAR      agt_nombre[41],      agt_colonia[41],      agt_calle[41],
            agt_fax[15],      agt_telefono[15],      agt_pais[5],
            agt_estado[5];
DBINT       agt_cod_posta;
char        pais_agE[16],      estado_agt[16];

int ctoi(char *cve, int no_arg)
{
char x;
int r_value=0;
while(7x = *cve++) != NULL)

```



```

    {
        fprintf(file_output, "%sIT%s QUANT %sUNIT          DESCRIPTION          %7.7-%sCNT%s
        fprintf(file_output, "%sEM%s ITY %sSUPPLIER PART No.          CODE%SRCB%
    }
    fprintf(file_output, "%s%2.2%s%7.7%s%36.36%s%3.3%s%13.13%s%14.14%s\n", LVGI, fi
    cnt_line += 7;
}

void footing_partidas(void)
{
    fprintf(file_output, "%s%2.2%s%7.7%s%36.36%s%3.3%s%13.13%s%14.14%s\n", A, file
}

void heading_page(void)
{
    if (strcmp(prv_nacionali, "N") == 0)
        sprintf(letr0, "SUBSIDIARIA DE INDUSTRIAS %7.7-s, S. A. DE C. V.", letr3);
    else
        sprintf(letr0, " SUBSIDIARY OF INDUSTRIAS %7.7-s, S. A. DE C. V.", letr3);
    if (acuse == 1)
    {
        letr1[0] = NULL;
        letr2[0][0] = NULL;
        letr2[1][0] = NULL;
        letr2[2][0] = NULL;
    }
    else
    {
        if (strcmp(prv_nacionali, "N") == 0)
        {
            strcpy(letr1, "          -OFICINAS GENERALES-          CORRESPONDENCIA A:");
            strcpy(letr2[0], "CITENSE ESTOS");
            strcpy(letr2[1], "NUMEROS EN TODA");
            strcpy(letr2[2], "CORRESPONDENCIA");
        }
        else
        {
            strcpy(letr1, "          -MAIN OFFICE-          MAIL TO:          ");
            strcpy(letr2[0], "ALWAYS REFER TO");
            strcpy(letr2[1], "THIS NUMBER ON");
            strcpy(letr2[2], "ALL DOCUMENTS ");
        }
    }
    if (cnt_page++ > 0)
        fprintf(file_output, "\f\r");
    else
    {
        if (export_to == 0)
        {
            reset_256X;
            lp1;
            tb;
            lp18;
            enable_draw;
        }
        cnt_line = 0;
        if (export_to == 0)
        {
            fprintf(file_output, "\n          ");
            lbig_letter;
        }
        else
        {
            fprintf(file_output, "          ");
            lbig_letter;
        }
        fprintf(file_output, "%40.40-s\n", cia_nombre2);
        if (export_to == 0)
            normal_letter;
        else
            lnormal_letter;
        if (acuse == 1)
            fprintf(file_output, "\n");
        fprintf(file_output, "          %40.40-s\n", alm_nombre);
        if (acuse == 1)
            fprintf(file_output, "\n          %49.49-s\n", letr0);
        else
            fprintf(file_output, "          %49.49-s\n", letr0);
        if (acuse == 0 && export_to == 1)
    }
}

```

```

lnormal letter;
fprintf(file_output, "%51.51-s\n", let1);
if (acuse == 1)
  fprintf(file_output, " ");
fprintf(file_output, "%32.32-s %32.32-s %15.15-s\n", dom_ofna_grl[0], dom_corres[0]
if (acuse == 1)
  fprintf(file_output, " ");
fprintf(file_output, "%32.32-s %32.32-s %15.15-s\n", dom_ofna_grl[1], dom_corres[1]
if (acuse == 1)
  fprintf(file_output, " ");
fprintf(file_output, "%32.32-s %32.32-s %15.15-s\n", dom_ofna_grl[2], dom_corres[2]
if (acuse == 0)
  {
    if (strcmp(prv_nacionali, "N") == 0)
      {
        if (strcmp(ped_status, "PC") == 0)
          {
            if (export to == 0)
              big_letter;
            else
              {
                fprintf(file_output, "\n          ");
                fprintf(file_output, "          PEDIDO CANCELADO\n\n");
                if (export to == 0)
                  normal_letter;
                else
                  lnormal_letter;
              }
            }
          }
        else
          {
            if (ped_num modif == 0)
              fprintf(file_output, "          PEDIDO SUJETO A CONDICIO
            else
              {
                if (export to == 0)
                  big_letter;
                else
                  {
                    fprintf(file_output, "\n          ");
                    fprintf(file_output, "          %2d%5 MODIFICACION AL PEDIDO\n", ped_
                    if (export to == 0)
                      normal_letter;
                    else
                      lnormal_letter;
                    fprintf(file_output, "          ESTE PEDIDO SUBSTITUYE AL
                  }
                }
              }
            }
          }
        }
      }
    else
      {
        if (strcmp(ped_status, "PC") == 0)
          {
            if (export to == 0)
              big_letter;
            else
              {
                fprintf(file_output, "\n          ");
                fprintf(file_output, "          ORDER CANCELLATION\n\n");
                if (export to == 0)
                  normal_letter;
                else
                  lnormal_letter;
              }
            }
          }
        else
          {
            if (ped_num modif == 0)
              fprintf(file_output, "          THIS ORDER IS SUBJECT TO TERMS AN
            else
              {
                if (export to == 0)

```

```

        big_letter;
    else
        {
            fprintf(file_output, "\n
            xbig_letter;
        }
        fprintf(file_output, "
        %2ds CHANGE ORDER\n", ped_num_mod
        if (export_to == 0)
            normal_letter;
        else
            Inormal letter;
        fprintf(file_output, "
        THIS ISSUE SUBSTITUTE THE OR
    }
}
}
}
}
else
{
    fprintf(file_output, "\n\n\n\n\n\n");
    return;
}
fprintf(file_output, "%s%44.44s%s%8.8s%s%8.8s%s%14.14s\n", Q, filler, W, Q, fill
y_o[0] = NULL;
if (strcmp(prv_rfc, prv_rfc_repre) != 0)
    strcpy(y_o, "y/o");
if (strcmp(prv_nacionali, "N") == 0)
    fprintf(file_output, "%sPROVEEDOR:
    %20.20-s%s%F.PEDIDO%s%EMBARQU
else
    fprintf(file_output, "%sSUPPLIER:
    %20.20-s%s%D.ISSUED%s%SHIPPIN
fprintf(file_output, "%s %39.39-s %3.3-s%s%8.8s%s%14.14s\n", LV, prv_r
fprintf(file_output, "%s %40.40-s %s%8.8-s%s%8.8-s%s%2.2-s%4.4-s%5.5d-%2.2d
if (strcmp(parte2, "OR SOONER") != 0)
{
    if (strcmp(prv_nacionali, "N") == 0)
        fprintf(file_output, "%s %40.40-s %s%8.8s%s%8.8-s%s%HOJA: %5d %s\n"
    else
        fprintf(file_output, "%s %40.40-s %s%8.8s%s%8.8-s%s%PAGE: %5d %s\n"
}
else
    fprintf(file_output, "%s %40.40-s %s%8.8s%s%9.9-s%s%PAGE: %5d %s\n", LV, pd
    fprintf(file_output, "%s%44.44s", A, filler, S);
else
    fprintf(file_output, "%s %40.40-s %s", LV, pdo_colonia, LV);
fprintf(file_output, "%s%8.8s%s%14.14s\n", A, filler, S, A, filler, S);
if (cnt_page > 1)
    goto head_detail;
fprintf(file_output, "%s %32.32-s C.P. %5.5-s%s%34.34s\n", LV, pdo_dlg_munic, pdo
fprintf(file_output, "%s %20.20-s %20.20-s %s", LV, dom_nom_esta, dom_nom_pais, LV
if (strcmp(prv_nacionali, "N") == 0)
{
    fprintf(file_output, "CONVENIO DE SUMINISTRO No.: %5.5-s%s\n", no_conv, LV);
    fprintf(file_output, "%sATTN: %38.38-s%s%CUENTA CARGO: %4.4d-%4.4d-%4.4d-%4.4
    fprintf(file_output, "%sTELEFONO: %14.14-s FAX: %14.14-s%s
}
else
{
    fprintf(file_output, "PROCUREMENT AGREEMENT No.: %5.5-s%s\n", no_conv, LV);
    fprintf(file_output, "%sATTN: %38.38-s%s%CHARGE ACCOUNT: %4.4d-%4.4d-%4.4d-%4.4
    fprintf(file_output, "%sPHONE: %14.14-s FAX: %14.14-s%s
}
if (strcmp(prv_nacionali, "N") == 0)
    fprintf(file_output, "%sCOTZ:
    %12.12-s %8.8-s %s%AUT. DESEMBOLS
else
    fprintf(file_output, "%sQUOTATION:
    %12.12-s %8.8-s %s%PAYMENT AUT.:
    fprintf(file_output, "%s%44.44s%s%34.34s\n", A, filler, S, A, filler, S);
    fprintf(file_output, "%s%44.44s%s%34.34s\n", Q, filler, W, Q, filler, W);
    if (strcmp(prv_nacionali, "N") == 0)
        fprintf(file_output, "%sFACTURAR A: %32.32-s%s%CONSIGNAR VIA: %19.19-s%s\n", LV,
    else
        fprintf(file_output, "%sINVOICE TO: %32.32-s%s%CONSIGN VIA: %21.21-s%s\n", LV, ci
    fprintf(file_output, "%s %42.42-s %s%34.34-s%s\n", LV, dom_fisca[0], LV, LV, cia_nombr
    fprintf(file_output, "%s %42.42-s %s%34.34-s%s\n", LV, dom_fisca[1], LV, LV, alm_nombr
    fprintf(file_output, "%s %42.42-s %s%34.34-s%s\n", LV, dom_fisca[2], LV, LV, dom_consi
    if (strcmp(prv_nacionali, "N") == 0)
        fprintf(file_output, "%sENVIAR FACTURAS AUTOGRAFAS Y DOCTOS. EMB: %s%34.34-s

```



```

void check_page(void)
{
int max_lines_1, max_lines_2;
max_lines_1 = MAX_LINES_1;
max_lines_2 = MAX_LINES_2;
if (ped_num_modif == 0)
{
max_lines_1 += 2;
max_lines_2 += 2;
}
if ((cnt_page == 1 && cnt_line > max_lines_1) || (cnt_page > 1 && cnt_line > max_1
{
footing_page();
heading_page();
}
cnt_line++;
}

void check_limits(void)
{
int max_lines_1, max_lines_2;
max_lines_1 = MAX_LINES_1;
max_lines_2 = MAX_LINES_2;
if (ped_num_modif == 0)
{
max_lines_1 += 2;
max_lines_2 += 2;
}
if ((cnt_page == 1 && cnt_line < max_lines_1) ||
(cnt_page > 1 && cnt_line < max_lines_2))
{
if (cnt_page == 1)
limit_page = max_lines_1;
else
limit_page = max_lines_2;
for(; cnt_line <= limit_page; ++cnt_line)
{
if (par_not == 0)
blank_detail_1();
else
blank_detail_2();
}
}
}

void check_page2(void)
{
int max_lines_1, max_lines_2;
max_lines_1 = MAX_LINES_1;
max_lines_2 = MAX_LINES_2;
if (ped_num_modif == 0)
{
max_lines_1 += 2;
max_lines_2 += 2;
}
if ((cnt_page == 1 && cnt_line > (max_lines_1 - 7)) || (cnt_page > 1 && cnt_line >
{
if (cnt_page == 1)
limit_page = max_lines_1;
else
limit_page = max_lines_2;
for(; cnt_line <= limit_page; ++cnt_line)
blank_detail_1();
footing_page();
par_not = 1;
heading_page();
}
else
{
footing_partidas();
cnt_line++;
heading_notes();
}
}

void accuse_recibo(void)

```



```

}
fprintf(file_output, " %40.40-s\n", cia_nombre);
fprintf(file_output, " %40.40-s\n", dom_corres[0]);
fprintf(file_output, " %40.40-s\n", dom_corres[1]);
fprintf(file_output, " %40.40-s\n", dom_corres[2]);
fprintf(file_output, " Atn. Depto. Corp. de Abastecimiento{f\r"};
}

```

```

main(int argc, char *argv[ ])

```

```

{
DBINT detect_office(void);
void cia_direccion(void);
void almacen_compania(void);
void pedido(void);
void requisicion(void);
void moneda(void);
void proveedor(void);
void calc_sub(void);
void direction_office(void);
void direction_corres(void);
void def_char_ctl(void);
void partida_pedido(void);
int x;
uname[0] = NULL;
upass[0] = NULL;
server[0] = NULL;
device[0] = NULL;
mycommand[0] = NULL;
file_name[0] = NULL;
file_name2[0] = NULL;
opcion[0] = NULL;
copies = 1;
ped_clave = 999999;
ped_num_conse = 999;
export_Eo = 0;
acuse = 0;
for(x=1;x<=argc;x++)
{
if (strstr(argv[x], "-d") != NULL)
{
strcpy(device, argv[x]);
kill_substring(device,0,2);
}
if (strstr(argv[x], "-n") != NULL)
{
kill_substring(argv[x],0,2);
if ((copies = (DBTINYINT) atoi(argv[x], 3)) > 99)
{
printf("El Valor maximo de las Copias es 99.\n");
exit(1);
}
}
if (strstr(argv[x], "-t") != NULL)
{
kill_substring(argv[x],0,2);
export_to = (DBTINYINT) atoi(argv[x], 4) + 1;
}
if (strstr(argv[x], "-U") != NULL)
{
strcpy(uname, argv[x]);
kill_substring(uname,0,2);
}
if (strstr(argv[x], "-P") != NULL)
{
strcpy(upass, argv[x]);
kill_substring(upass,0,2);
}
if (strstr(argv[x], "-S") != NULL)
{
strcpy(server, argv[x]);
kill_substring(server,0,2);
}
if (strstr(argv[x], "-R") != NULL)
{
strcpy(alm_clave, argv[x]);
kill_substring(alm_clave,0,2);
}
}
}

```

```

to upper(alm_clave);
if (strlen(alm_clave) > 4)
{
printf("La clave del almacen no debe exeder los 4 caracteres.\n");
exit(2);
}
if (strchr(argv[x], "-C") != NULL)
{
kill_substring(argv[x], 0, 2);
if ((ped_clave = (DBINT) atoi(argv[x], 2)) > 99999)
{
printf("El Valor maximo del Numero del Pedido es 99999.\n");
exit(3);
}
}
if (strchr(argv[x], "-A") != NULL)
{
kill_substring(argv[x], 0, 2);
if ((ped_num_conse = (DBTINYINT) atoi(argv[x], 3)) > 99)
{
printf("El Valor maximo del Anexo del Pedido es 99.\n");
exit(4);
}
}
if (strchr(argv[x], "-F") != NULL)
{
strcpy(file_name, argv[x]);
kill_substring(file_name, 0, 2);
strcat(file_name, "-rpt");
}
}
if (strlen(device) == 0)
{
printf("El dispositivo de salida es requerido...\n");
exit(7);
}
if (export_to == 0)
{
printf("El tipo de Impresora (-t) es requerido...\n");
exit(7);
}
export_to--;
if (export_to == 1)
{
MAX_LINES_1 == 5;
MAX_LINES_2 == 5;
}
if (strlen(uname) == 0)
{
printf("El usuario es requerido...\n");
exit(6);
}
if (strlen(upass) == 0)
{
printf("El password es requerido...\n");
exit(7);
}
if (strlen(alm_clave) == 0)
{
printf("El Requisitor es requerido...\n");
exit(8);
}
if (ped_clave > 99999)
{
printf("La Clave del Pedido es requerido...\n");
exit(9);
}
if (ped_num_conse > 99)
{
printf("El Anexo del Pedido es requerido...\n");
exit(10);
}
}
par_not = 0;
f_not = 0;
if (dbinit() == FAIL)

```

```

    exit (ERREXIT);
dberrhandle(err_handler);
dbmsghandle(msg_handler);
login = dblogin();
DBSETLUSER(login,uname);
DBSETLPWD(login,upass);
sqltemp = dbopen(login, server);
sqlped = dbopen(login, server);
sqlppe = dbopen(login, server);
sqlreq = dbopen(login, server);
sqlpro = dbopen(login, server);
sqlppr = dbopen(login, server);
dbloginfree(login);
if ((count_office = detect_office()) != 1)
{
    if (count_office == 0)
        printf("No hay Oficina operando.\n");
    else
        printf("Hay mas de una Oficina operando.\n");
    exit(2);
}
if (strlen(file_name) == 0)
    sprintf(file_name, "%s_d_d.rpt.%s",alm_clave,ped_clave,ped_num_conse,uname);
almacen_compania();
pedido();
if ((strcmp(ofi_clave, com_ofi_clave)) != 0)
{
    printf("El pedido no corresponde a la oficina operando.\n");
    exit(7);
}
roman_PC(letr3);
roman_PC(letr4);
roman_PC(letr5);
requisicion();
proveedor();
strcpy(fec_elabo,ped_fec_elabo);
if (strcmp(prv_nacionali, "N") != 0)
{
    cia_dom_fisca[0] = NULL;
    strcpy(cia_dom_fisca, cia_nom_fisca2);
    if (ped_tpo_pedid == 1)
    {
        strcpy(partel, "SEE");
        strcpy(parte2, "SCHEDULE");
    }
    else
    {
        strcpy(partel, ped_fec_entre);
        strcpy(parte2, "OR-SOONER");
    }
    if (ped_num_modif != 0)
    {
        strcpy(fec_elabo, ped_fec_modif);
        strcpy(num_change, "th");
        if (ped_num_modif == 1)
            strcpy(num_change, "st");
        if (ped_num_modif == 2)
            strcpy(num_change, "nd");
        if (ped_num_modif == 3)
            strcpy(num_change, "rd");
    }
}
}
else
{
    if (ped_tpo_pedid == 1)
    {
        strcpy(partel, "VER");
        strcpy(parte2, "PROGRAMA");
    }
    else
    {
        strcpy(partel, ped_fec_entre);
        strcpy(parte2, "O ANTES");
    }
    if (ped_num_modif != 0)
    {

```

```

        strcpy(fec_elabo,ped_fec_modif);
        strcpy(num_change,"a.");
    }
}
cia direction();
moneda();
calc_sub();
if ((strlen(ofi_dom_corre) == 0)
    strcpy(ofi_dom_corre, ofi_dom_regio);
direction_office();
direction_corres();
cnt_page = 0;
def_char_ctl();
seconds = time(NULL);
today_date = localtime(&seconds);
strftime(date, (size_t)32, "FECHA: %b %d %Y, HORA: %H:%M:%S", today_date);
if ((file_output = fopen(file_name,"w")) == NULL)
{
    printf("Error al Abrir archivo %s\n",file_name);
    exit(8);
}
heading_page();
partida_pedido();
check_limits();
footing_page();
if (fclose(file_output) != NULL)
{
    printf("Error al Cerrar archivo %s\n",file_name);
    exit(9);
}
sprintf(mycommand,"lp -d%-s -n%-s -s -onb %s",device,copies,file_name);
system(mycommand);
for(x=1;x<=argc;x++)
{
    if (strstr(argv[x], "-O") != NULL)
    {
        strcpy(option, argv[x]);
        to_upper(option);
        if (strcmp(option, "-OREMOVE") == 0)
        {
            kill_substring(argv[x],0,2);
            remove(file_name);
            continue;
        }
    }
    if (strcmp(argv[x], "-OAR") == 0)
    {
        strcpy(file_name2, file_name);
        strcat(file_name2, ".AR");
        if ((file_output = fopen(file_name2,"w")) == NULL)
        {
            printf("Error al Abrir archivo %s\n",file_name2);
            exit(8);
        }
        acuse = 1;
        cnt_page = 0;
        heading_page();
        acuse_fecibo();
        if (fclose(file_output) != NULL)
        {
            printf("Error al Cerrar archivo %s\n",file_name2);
            exit(9);
        }
        sprintf(mycommand,"lp -d%-s -s -onb %s",device,file_name2);
        system(mycommand);
        remove(file_name2);
        break;
    }
}
}
printf("\n");
dbexit();
exit(0);
}
DBINT detect_office(void)
{

```

```

dbfcmd(sqltemp, " select offi_clave, " "});
dbfcmd(sqltemp, " offi_nombre, " "});
dbfcmd(sqltemp, " offi_dom_regio, " "});
dbfcmd(sqltemp, " offi_dom_corr, " "});
dbfcmd(sqltemp, " from ofna_regional " "});
dbfcmd(sqltemp, " where offi_ind_opera = 1 " "});
dbsqlxexac(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
  dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, offi_clave);
  dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, offi_nombre);
  dbbind(sqltemp, 3, STRINGBIND, (DBINT)0, offi_dom_regio);
  dbbind(sqltemp, 4, STRINGBIND, (DBINT)0, offi_dom_corr);
  while(dbnextrow(sqltemp) != NO_MORE_ROWS)
  {
    roman_PC(offi_nombre);
    roman_PC(offi_dom_regio);
    roman_PC(offi_dom_corr);
  }
  return DBCOUNT(sqltemp);
}
}

```

```
void almacen_compania(void)
```

```

{
  dbfcmd(sqltemp, " select alm_clave, " "});
  dbfcmd(sqltemp, " alm_nombre, " "});
  dbfcmd(sqltemp, " alm_nom_respo, " "});
  dbfcmd(sqltemp, " cia_nombre, " "});
  dbfcmd(sqltemp, " unidad.uni_clave, " "});
  dbfcmd(sqltemp, " compania.cia_clave, " "});
  dbfcmd(sqltemp, " cia_dom_fisca2, " "});
  dbfcmd(sqltemp, " cia_nom_fisca2, " "});
  dbfcmd(sqltemp, " from almacen, unidad, compania " "});
  dbfcmd(sqltemp, " where alm_clave = upper('%s') " alm_clave " "});
  dbfcmd(sqltemp, " and unidad.uni_clave = almacen.uni_clave " "});
  dbfcmd(sqltemp, " and compania.cia_clave = unidad.cia_clave " "});
  dbsqlxexac(sqltemp);
  if (dbresults(sqltemp) == SUCCEED)
  {
    dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, alm_clave);
    dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, alm_nombre);
    dbbind(sqltemp, 3, STRINGBIND, (DBINT)0, alm_nom_respo);
    dbbind(sqltemp, 4, STRINGBIND, (DBINT)0, cia_nombre);
    dbbind(sqltemp, 5, STRINGBIND, (DBINT)0, uni_clave);
    dbbind(sqltemp, 6, SMALLBIND, (DBINT)0, (BYTE *) &cia_clave);
    dbbind(sqltemp, 7, STRINGBIND, (DBINT)0, cia_dom_fisca);
    dbbind(sqltemp, 8, STRINGBIND, (DBINT)0, cia_nom_fisca2);
    while(dbnextrow(sqltemp) != NO_MORE_ROWS)
    {
      roman_PC(alm_nombre);
      roman_PC(alm_nom_respo);
      roman_PC(cia_nombre);
      roman_PC(uni_clave);
      roman_PC(cia_dom_fisca);
      roman_PC(cia_nom_fisca2);
      strcpy(cia_nombre2, cia_nombre);
    }
    if (DBCOUNT(sqltemp) == 0)
    {
      printf("No existe el Almacen o la estructura Compa$ia-Unidad-Almacen est
      exit(3);
    }
  }
}

```

```
void pedido(void)
```

```

{
  void alm_direction_fact(void);
  void alm_direction_cons(void);
  void nom_fom(char *eve_com, char *nom_compr, char *clave_ofi);
  int long_dom;
  cot.clave[0] = NULL; cot.fec_cotiz[0] = NULL;
  dbfcmd(sqlped, " select prv_rfc = rtrim(prv_rfc),
  dbfcmd(sqlped, " prv_rfc_repre = rtrim(prv_rfc_repre),
  dbfcmd(sqlped, " pdo_num_conse,

```

```

dbfcmd(sqlped, " pdo_atencion,
dbfcmd(sqlped, " pedido.com_clave,
dbfcmd(sqlped, " ped_fec_elabo = convert(char(8), ped_fec_elabo,
dbfcmd(sqlped, " ped_fec_entra = convert(char(8), ped_fec_entra,
dbfcmd(sqlped, " ped_fec_ultima_repro = convert(char(8), ped_fec_ultima_rep
dbfcmd(sqlped, " mon_clave,
dbfcmd(sqlped, " ped_imp_total,
dbfcmd(sqlped, " ped_pje_lmpto,
dbfcmd(sqlped, " ped_pje_dacto,
dbfcmd(sqlped, " prv_dto_pront_pago,
dbfcmd(sqlped, " prv_dia_pront_pago,
dbfcmd(sqlped, " prv_cnd_pago,
dbfcmd(sqlped, " prv_dia_condi_pago,
dbfcmd(sqlped, " prv_cnd_entra,
dbfcmd(sqlped, " prv_cve_lugar_entra,
dbfcmd(sqlped, " ped_nom_lugar_entra,
dbfcmd(sqlped, " pdo_cve_domic_cons1,
dbfcmd(sqlped, " ped_via_cons1,
dbfcmd(sqlped, " ped_num_conve = isnull(ped_num_conve,0),
dbfcmd(sqlped, " ped_mat_orige,
dbfcmd(sqlped, " agt_clave,
dbfcmd(sqlped, " ped_tip_segur,
dbfcmd(sqlped, " ped_fec_autor = convert(char(8), ped_fec_autor,
dbfcmd(sqlped, " ped_per_autor,
dbfcmd(sqlped, " ped_num_modif,
dbfcmd(sqlped, " ped_fec_modif = convert(char(8), ped_fec_modif,
dbfcmd(sqlped, " ped_imp_pagad,
dbfcmd(sqlped, " ped_tpo_pedid,
dbfcmd(sqlped, " ped_comision,
dbfcmd(sqlped, " ped_status,
dbfcmd(sqlped, " ped_fec_statu = convert(char(8), ped_fec_statu,
dbfcmd(sqlped, " ofi_clave,
dbfcmd(sqlped, " pdo_cve_domic_factu,
dbfcmd(sqlped, " cot_clave
dbfcmd(sqlped, " from pedIdo, comprador
dbfcmd(sqlped, " where alm_clave = 'as', alm_clave
dbfcmd(sqlped, " and ped_clave = 'd', ped_clave
dbfcmd(sqlped, " and ped_num_conse = 'd', ped_num_conse
dbfcmd(sqlped, " and comprador.com_clave = pedido.com_clave
dbsqlxec(sqlped);
if (dbresults(sqlped) == SUCCEED)
{
dbbind(sqlped, 1, STRINGBIND, (DBINT)0, prv_rfc);
dbbind(sqlped, 2, STRINGBIND, (DBINT)0, prv_rfc_repre);
dbbind(sqlped, 3, TINYBIND, (DBINT)0, (BYTE *) &pdo_num_conse);
dbbind(sqlped, 4, STRINGBIND, (DBINT)0, pdo_atencion);
dbbind(sqlped, 5, STRINGBIND, (DBINT)0, com_clave);
dbbind(sqlped, 6, STRINGBIND, (DBINT)0, ped_fec_elabo);
dbbind(sqlped, 7, STRINGBIND, (DBINT)0, ped_fec_entra);
dbbind(sqlped, 8, STRINGBIND, (DBINT)0, ped_fec_ultima_repro);
dbbind(sqlped, 9, STRINGBIND, (DBINT)0, mon_clave);
dbbind(sqlped, 10, REALBIND, (DBINT)0, (BYTE *) &ped_imp_total);
dbbind(sqlped, 11, REALBIND, (DBINT)0, (BYTE *) &ped_pje_lmpto);
dbbind(sqlped, 12, REALBIND, (DBINT)0, (BYTE *) &ped_pje_dacto);
dbbind(sqlped, 13, REALBIND, (DBINT)0, (BYTE *) &prv_dto_pront_pago);
dbbind(sqlped, 14, TINYBIND, (DBINT)0, (BYTE *) &prv_dia_pront_pago);
dbbind(sqlped, 15, STRINGBIND, (DBINT)0, prv_cnd_pago);
dbbind(sqlped, 16, TINYBIND, (DBINT)0, (BYTE *) &prv_dia_condi_pago);
dbbind(sqlped, 17, STRINGBIND, (DBINT)0, prv_cnd_entra);
dbbind(sqlped, 18, STRINGBIND, (DBINT)0, prv_cve_lugar_entra);
dbbind(sqlped, 19, STRINGBIND, (DBINT)0, ped_nom_lugar_entra);
dbbind(sqlped, 20, TINYBIND, (DBINT)0, (BYTE *) &pdo_cve_domic_cons1);
dbbind(sqlped, 21, STRINGBIND, (DBINT)0, ped_via_cons1);
dbbind(sqlped, 22, SHALBIND, (DBINT)0, (BYTE *) &ped_num_conve);
dbbind(sqlped, 23, STRINGBIND, (DBINT)0, ped_mat_orige);
dbbind(sqlped, 24, STRINGBIND, (DBINT)0, agt_clave);
dbbind(sqlped, 25, STRINGBIND, (DBINT)0, ped_tip_segur);
dbbind(sqlped, 26, STRINGBIND, (DBINT)0, ped_fec_autor);
dbbind(sqlped, 27, STRINGBIND, (DBINT)0, ped_per_autor);
dbbind(sqlped, 28, TINYBIND, (DBINT)0, (BYTE *) &ped_num_modif);
dbbind(sqlped, 29, STRINGBIND, (DBINT)0, ped_fec_modif);
dbbind(sqlped, 30, MONEYBIND, (DBINT)0, (BYTE *) &ped_imp_pagad);
dbbind(sqlped, 31, TINYBIND, (DBINT)0, (BYTE *) &ped_tpo_pedid);
dbbind(sqlped, 32, REALBIND, (DBINT)0, (BYTE *) &ped_comision);
dbbind(sqlped, 33, STRINGBIND, (DBINT)0, ped_status);
dbbind(sqlped, 34, STRINGBIND, (DBINT)0, ped_fec_statu);
}

```

```

dbbind(sqlped, 35, STRINGBIND, (DBINT)0, com_ofi_clave);
dbbind(sqlped, 36, TINYBIND, (DBINT)0, (BYTE *) &ped_cve_domic_factu);
dbbind(sqlped, 37, STRINGBIND, (DBINT)0, cot_clave);
while(dbnextrrow(sqlped) != NO_MORE_ROWS);
if (DBCOUNT(sqlped) == 0)
{
printf("No existe el Pedido...\n");
exit(6);
}
else
{
roman_PC(pdo_atencion);
roman_PC(ped_nom lugar entro);
if (ped_imp Total == 0)
strcpy(ped status, "PC");
if (prv dia pront_pago != 0)
sprintf(dia_pront_pago, "%3d", prv_dia_pront_pago);
else
dia_pront_pago[0] = NULL;
if (strlen(cot_clave) > 0)
{
dbfcmd(sqltemp, " select cot_fec cotiz=convert(char(8),cot_fec_cotiz,
dbfcmd(sqltemp, " from cotizacion ");
dbfcmd(sqltemp, " where prv_rfc = '%s' ", prv_rfc );
dbfcmd(sqltemp, " and cot_clave = '%s' ", cot_clave );
dbsqlxex(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, cot_fec_cotiz);
while(dbnextrrow(sqltemp) != NO_MORE_ROWS);
}
}
nom_com( com_clave, com_nombre, com_ofi_clave );
nom_com(ped_per_autor, ped_nom_autor1, com_ofi_clave );
dbfcmd(sqltemp, " select dom_nombre ");
if (ped_cve_domic_factu != 9)
{
dbfcmd(sqltemp, " from almacen domicilio, domicilio ");
dbfcmd(sqltemp, " where alm_clave = '%s' ", alm_clave );
dbfcmd(sqltemp, " and ald_tipo = '1' ");
dbfcmd(sqltemp, " and ald_num conse = %d ", ped_cve_domic_factu );
dbfcmd(sqltemp, " and dom_clave = ald_clave ");
}
else
{
dbfcmd(sqltemp, " from domicilio temporal ");
dbfcmd(sqltemp, " where alm_clave = '%s' ", alm_clave );
dbfcmd(sqltemp, " and ped_clave = %d ", ped_clave );
dbfcmd(sqltemp, " and ped_num conse = %d ", ped_num_conse );
dbfcmd(sqltemp, " and dom_tipo = 1 ");
}
dbsqlxex(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, ald_nombre_fac);
while(dbnextrrow(sqltemp) != NO_MORE_ROWS)
roman_PC(ald_nombre_fac);
alm_dirction_Fact();
}
dbfcmd(sqltemp, " select dom_nombre ");
if (pdo_cve_domic_consi != 9)
{
dbfcmd(sqltemp, " from almacen domicilio, domicilio ");
dbfcmd(sqltemp, " where alm_clave = '%s' ", alm_clave );
dbfcmd(sqltemp, " and ald_tipo = '2' ");
dbfcmd(sqltemp, " and ald_num conse = %d ", pdo_cve_domic_consi );
dbfcmd(sqltemp, " and dom_clave = ald_clave ");
}
else
{
dbfcmd(sqltemp, " from domicilio temporal ");
dbfcmd(sqltemp, " where alm_clave = '%s' ", alm_clave );
dbfcmd(sqltemp, " and ped_clave = %d ", ped_clave );
dbfcmd(sqltemp, " and ped_num conse = %d ", ped_num_conse );
dbfcmd(sqltemp, " and dom_tipo = 2 ");
}
}
}

```

```

dbsqlxec(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
    dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, ald_nombre_con);
    while(dbnextrow(sqltemp) != NO_MORE_ROWS)
        roman_PC(ald_nombre_con);
    alm_dirSection_Cons();
}
if (ped_num_conve == 0)
    no_Conv[0] = NULL;
else
{
    sprintf(no_conv, "%4d", ped_num_conve);
    insert_substring(no_conv, "-", 2);
}
}
}
}

```

void partida_pedido(void)

```

{
void print notes(void);
int descripcion(void);
int w, x, y, z;
spps num_conse[0] = NULL;
dbfcmd(sqlppe, " select ppe_num_conse, ");
if (strcmp(prv_nacionali, "N") == -0)
    dbfcmd(sqlppe, " ume_unimed, ");
else
    dbfcmd(sqlppe, " ume_unimed_ing, ");
dbfcmd(sqlppe, " ppe_cantidad, ");
dbfcmd(sqlppe, " ppe_pre_unita, ");
dbfcmd(sqlppe, " mun_clave, ");
dbfcmd(sqlppe, " ppe_ptj_dcto, ");
dbfcmd(sqlppe, " pp_des_clave, ");
dbfcmd(sqlppe, " ppe_cve_prog, ");
dbfcmd(sqlppe, " mco_clave, ");
dbfcmd(sqlppe, " pp_req_num_conse, ");
dbfcmd(sqlppe, " pp_prq_num_conse, ");
dbfcmd(sqlppe, " from partida_pedido pp, ");
dbfcmd(sqlppe, " partida_requisicion pr, ");
dbfcmd(sqlppe, " unidad_medida um ");
dbfcmd(sqlppe, " where pp.alm_clave = 's' ", alm_clave );
dbfcmd(sqlppe, " and ped_clave = d ", ped_clave );
dbfcmd(sqlppe, " and ped_num_conse = d ", ped_num_conse );
dbfcmd(sqlppe, " and pr.alm_clave = pp.alm_clave ");
dbfcmd(sqlppe, " and pr.req_num_conse = pp.req_num_conse ");
dbfcmd(sqlppe, " and pr.prq_num_conse = pp.prq_num_conse ");
dbfcmd(sqlppe, " and um.ume_clave = pr.ume_clave ");
dbfcmd(sqlppe, " order by ppe_num_conse ");
dbsqlxec(sqlppe);
if (dbresults(sqlppe) == SUCCEED)
{
    dbbind(sqlppe, 1, SMALLBIND, (DBINT)0, (BYTE *) &ppe_num_conse);
    dbbind(sqlppe, 2, STRINGBIND, (DBINT)0, ume_unimed);
    dbbind(sqlppe, 3, REALBIND, (DBINT)0, (BYTE *) &ppe_cantidad);
    dbbind(sqlppe, 4, REALBIND, (DBINT)0, (BYTE *) &ppe_pre_unita);
    dbbind(sqlppe, 5, STRINGBIND, (DBINT)0, mun_clave);
    dbbind(sqlppe, 6, REALBIND, (DBINT)0, (BYTE *) &ppe_ptj_dcto);
    dbbind(sqlppe, 7, INTBIND, (DBINT)0, (BYTE *) &des_clave);
    dbbind(sqlppe, 8, STRINGBIND, (DBINT)0, ppe_cve_prog);
    dbbind(sqlppe, 9, STRINGBIND, (DBINT)0, mco_clave);
    dbbind(sqlppe, 10, INTBIND, (DBINT)0, (BYTE *) &req_num_conse);
    dbbind(sqlppe, 11, SMALLBIND, (DBINT)0, (BYTE *) &prq_num_conse);
    while(dbnextrow(sqlppe) != NO_MORE_ROWS)
    {
        if (strcmp(ppe_cve_prog, "S") != 0)
            sprintf(pppe_num_conse, "%2d", ppe_num_conse);
        else
        {
            strcpy(spps num_conse, " ");
            dbfcmd(sqltemp, " select sum(emb_cantidad) ");
            dbfcmd(sqltemp, " from embarque ");
            dbfcmd(sqltemp, " where alm_clave = 's' ", alm_clave );
            dbfcmd(sqltemp, " and ped_clave = d ", ped_clave );
            dbfcmd(sqltemp, " and ped_num_conse = d ", ped_num_conse );
        }
    }
}
}
}

```

```

dbfcmd(sqltemp, " and ppe_num_conse = %d ", ppe_num_conse );
dbsexec(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
    dbbind(sqltemp, 1, REALBIND, (DBINT)0, &ppe_cantidad);
    dbnextrow(sqltemp);
}
sprintf(appe_cantidad,"%f",ppe_cantidad);
sprintf(areq_num_conse,"%d",req_num_conse);
sprintf(sprq_num_conse,"%d",prq_num_conse);
check_page();
for(w = strlen(appe_cantidad);ppe_cantidad[w] != '.';w--)
{
    if (ppe_cantidad[strlen(appe_cantidad) - 1] != '0')
        break;
    kill_substring(appe_cantidad,(strlen(appe_cantidad) - 1),1);
}
if (ppe_cantidad[strlen(appe_cantidad) - 1] == '.')
    kill_substring(appe_cantidad,(strlen(appe_cantidad) - 1),1);
if (strcmp(ped_status,"PC") == 0)
    ppe_cantidad = 0;
if (strcmp(ppe_cve_progrr,"S") != 0)
{
    fprintf(file_output,"%s2.2-s%s7.7s%s5.5-s %4.4-s %5.5-s %2.2-s %1
    fprintf(file_output,"%14.14s\n",format_money((ppe_pre_unita * ppe_can
}
else
{
    fprintf(file_output,"%s2.2-s%s7.7s%s5.5-s %4.4-s %5.5-s %2.2-s %1
    fprintf(file_output,"%14.14s\n",format_money(0.0, NULL, 14, 2),LV);
}
dbfcmd(sqltemp, " select des_nombre ");
dbfcmd(sqltemp, " from descr temporal ");
dbfcmd(sqltemp, " where des_clave = %d ",des_clave );
dbfcmd(sqltemp, "order by des_num_conse ");
dbsexec(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
    dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, des_nombre);
    w = 0;
    while(dbnextrow(sqltemp) != NO_MORE_ROWS)
    {
        roman_PC(des_nombre);
        if (w > 0)
        {
            w += strlen(des_nombre);
            des_nombre2 = realloc(des_nombre2,(w * sizeof(char)));
            strcat(des_nombre2,des_nombre);
        }
        else
        {
            w = strlen(des_nombre) + 1;
            des_nombre2 = malloc(w * sizeof(char));
            strcpy(des_nombre2,des_nombre);
        }
    }
    if (DBCOUNT(sqltemp) == 0)
    {
        des_nombre2 = malloc(35 * sizeof(char));
        strcpy(des_nombre2,"*** PARTIDA SIN DESCRIPCION ***");
    }
    z = descripcion();
    for(y=0;y<=z;y++)
    {
        check_page();
        fprintf(file_output,"%s %s %s36.36-ats %s %s
    }
    free(des_nombre2);
}
if (ppe_ptj_dscto > 0)
{
    check_page();
    if (strcmp(prv_nacionali,"N") == 0)
        fprintf(file_output,"%s %s %s DESCUENTO PARTIDA %5.
    else

```



```

    fprintf(file_output,"%s %s %s\n",LV,LV,LV,format_mo
if }ped_pje_imppto != 0)
{
    if (strcmp(mon_clave, "MN") == 0)
        strcpy(nom_imppto, "I.V.A.");
    else
        if (strcmp(prv_nacionali, "N") == 0)
            strcpy(nom_imppto, "IMPUESTO");
        else
            strcpy(nom_imppto, "TAX");
    check_page();
    blank_detail_1();
    check_page();
    fprintf(file_output,"%s %s %s\n",LV,LV,LV);
    fprintf(file_output,"%5.5s %s",nom_imppto,format_money(
    fprintf(file_output,"%s %s %s\n",LV,LV,LV,format_mo
}
dbfcmd(sqltemp, " select nta_descripcion, " ");
dbfcmd(sqltemp, " nta_clave " ");
dbfcmd(sqltemp, " from notas_pedido " ");
dbfcmd(sqltemp, " where alm_clave = '%s' ",alm_clave );
dbfcmd(sqltemp, " and ped_clave = %d ",ped_clave );
dbfcmd(sqltemp, " and ped_num_conse = %d ",ped_num_conse );
dbfcmd(sqltemp, " order by nta_num_secue " ");
dbfcmd(sqltemp, " nta_num_conse " ");
dbsqlxex(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
    dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, des_nombre);
    dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, nta_clave);
    strcpy(break_nta_clave, "XXXXXXXXXXXXXX");
    while(dbnxtrow(sqltemp) != NO_MORE_ROWS)
    {
        roman_PC(des_nombre);
        if (strcmp(break_nta_clave, nta_clave) != 0)
        {
            if (strcmp(break_nta_clave, "XXXXXXXXXXXXXX") != 0)
                print_notes();
            else
                check_page2();
            w = 0;
            strcpy(break_nta_clave, nta_clave);
        }
        if (w > 0)
        {
            w += strlen(des_nombre);
            des_nombre2 = ralloc(des_nombre2, w * sizeof(char));
            strcat(des_nombre2,des_nombre);
        }
        else
        {
            w = strlen(des_nombre) + 1;
            des_nombre2 = malloc(w * sizeof(char));
            strcpy(des_nombre2,des_nombre);
        }
    }
    if (DBCOUNT(sqltemp) != 0)
        print_notes();
}
}
}

void print_notes(void)
{
    int y, z;
    par_not = 1;
    check_page();
    blank_detail_2();
    z = descripcion2();
    for(y=0;y<=z;y++)
    {
        check_page();
        fprintf(file_output,"%s %72.72-s %s\n",LV,des_nota[y],LV);
    }
    free(des_nombre2);
}

```

```

des_nombre2[0] = NULL;
}

void requisicion(void)
{
int x;
dbfcmd(sqlreq, " select alm_cta_cuent,                                ");
dbfcmd(sqlreq, "     alm_cta_subct,                                ");
dbfcmd(sqlreq, "     alm_cta_ssubc,                                ");
dbfcmd(sqlreq, "     alm_cta_conce,                                ");
dbfcmd(sqlreq, "     req_aut_plant,                                ");
dbfcmd(sqlreq, "     req_aut_folio,                                ");
dbfcmd(sqlreq, "     req_aut_perio,                                ");
dbfcmd(sqlreq, "     req_des_gener,                                ");
dbfcmd(sqlreq, "     req_fec_elabo = convert(char(8), req_fec_elabo, 3), ");
dbfcmd(sqlreq, "     req_fec_recep = convert(char(8), req_fec_recep, 3), ");
dbfcmd(sqlreq, "     req_fec_asign = convert(char(8), req_fec_asign, 3), ");
dbfcmd(sqlreq, "     req_fec_reque = convert(char(8), req_fec_reque, 3), ");
dbfcmd(sqlreq, "     req_imp_estim = isnull(req_imp_estim, $0.00), ");
dbfcmd(sqlreq, "     req_tip_servi,                                ");
dbfcmd(sqlreq, "     req_cod_prior,                                ");
dbfcmd(sqlreq, "     req_nom_solic,                                ");
dbfcmd(sqlreq, "     req_ord_traba,                                ");
dbfcmd(sqlreq, "     req_num equip,                                ");
dbfcmd(sqlreq, "     req_dep_nombr,                                ");
dbfcmd(sqlreq, "     req_num_emple,                                ");
dbfcmd(sqlreq, "     req_folio,                                    ");
dbfcmd(sqlreq, "     mon_clave,                                    ");
dbfcmd(sqlreq, "     pr.lin_clave,                                    ");
dbfcmd(sqlreq, "     lin_sub_linea                                ");
dbfcmd(sqlreq, " from requisicion r, partida_requisicion pr ");
dbfcmd(sqlreq, " where r.alm_clave = 'ts' ",alm_clave ");
dbfcmd(sqlreq, "     and r.req_num_conse = 'd' ",ped_clave ");
dbfcmd(sqlreq, "     and pr.alm_clave = r.alm_clave ");
dbfcmd(sqlreq, "     and pr.req_num_conse = r.req_num_conse ");
dbfcmd(sqlreq, "     and prq_num_conse = 1 ");
dbqlexec(sqlreq);
if (dbresults(sqlreq) == SUCCEEDED)
{
dbbind(sqlreq, 1, SMALLBIND, (DBINT)0, (BYTE *) &alm_cta_cuent);
dbbind(sqlreq, 2, TINYBIND, (DBINT)0, (BYTE *) &alm_cta_subct);
dbbind(sqlreq, 3, TINYBIND, (DBINT)0, (BYTE *) &alm_cta_ssubc);
dbbind(sqlreq, 4, SMALLBIND, (DBINT)0, (BYTE *) &alm_cta_conce);
dbbind(sqlreq, 5, STRINGBIND, (DBINT)0, req_aut_plant);
dbbind(sqlreq, 6, INTBIND, (DBINT)0, (BYTE *) &req_aut_folio);
dbbind(sqlreq, 7, TINYBIND, (DBINT)0, (BYTE *) &req_aut_perio);
dbbind(sqlreq, 8, STRINGBIND, (DBINT)0, req_des_gener);
dbbind(sqlreq, 9, STRINGBIND, (DBINT)0, req_fec_elabo);
dbbind(sqlreq, 10, STRINGBIND, (DBINT)0, req_fec_recep);
dbbind(sqlreq, 11, STRINGBIND, (DBINT)0, req_fec_asign);
dbbind(sqlreq, 12, STRINGBIND, (DBINT)0, req_fec_reque);
dbbind(sqlreq, 13, REALBIND, (DBINT)0, (BYTE *) &req_imp_estim);
dbbind(sqlreq, 14, STRINGBIND, (DBINT)0, req_tip_servi);
dbbind(sqlreq, 15, STRINGBIND, (DBINT)0, req_cod_prior);
dbbind(sqlreq, 16, STRINGBIND, (DBINT)0, req_nom_solic);
dbbind(sqlreq, 17, STRINGBIND, (DBINT)0, req_ord_traba);
dbbind(sqlreq, 18, STRINGBIND, (DBINT)0, req_num equip);
dbbind(sqlreq, 19, STRINGBIND, (DBINT)0, req_dep_nombr);
dbbind(sqlreq, 20, INTBIND, (DBINT)0, (BYTE *) &req_num_emple);
dbbind(sqlreq, 21, STRINGBIND, (DBINT)0, req_folio);
dbbind(sqlreq, 22, STRINGBIND, (DBINT)0, req_mon_clave);
dbbind(sqlreq, 23, TINYBIND, (DBINT)0, (BYTE *) &lin_clave);
dbbind(sqlreq, 24, STRINGBIND, (DBINT)0, lin_sub_linea);
while(dbnextrow(sqlreq) != NO_MORE_ROWS);
if (DBCOUNT(sqlreq) != 0)
{
roman_PC(req_des_gener);
roman_PC(req_nom_solic);
roman_PC(req_dep_nombr);
if (req_num_emple != 0)
printf(num_emple, "%4d", req_num_emple);
else
strcpy(num_emple, " ");
for(x=0;req_des_gener[x] != NULL;x++)
if (req_des_gener[x] == '\n')
req_des_gener[x] = ' ';
}
}

```

```

if (x > 81)
  req_des_gener[81] = NULL;
switch (req_cod_prior[0])
{
  case 'A': strcpy(nom_pri, "Extra Urgente");
            break;
  case 'B': strcpy(nom_pri, "Urgente");
            break;
  case 'C': strcpy(nom_pri, "Normal");
            break;
  default: strcpy(nom_pri, "INVALIDA");
}
}
}
}

```

```

void moneda(void)
{
  nom_via_cons[0] = NULL;
  if (strlen(ped_via_cons) != 0)
    tabla_detalle(4, ped_via_cons, nom_via_cons);
  mon_nom_espan[0] = NULL;
  if (strcmp(prv_nacionali, "N") == 0)
    dbfcmd(sqltemp, " select mon_nom_espan");
  else
    dbfcmd(sqltemp, " select mon_nom_ingl");
  dbfcmd(sqltemp, " from moneda");
  dbfcmd(sqltemp, " where mon_clave = '%s'", mon_clave);
  dbsqlxec(sqltemp);
  if (dbresults(sqltemp) == SUCCEED)
  {
    dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, mon_nom_espan);
    while(dbnextraw(sqltemp) != NO_MORE_ROWS);
    if (DBCOUNT(sqltemp) != 0)
      if (strcmp(mon_clave, "MN") != 0)
      {
        dbfcmd(sqltemp, " select tca_tip_cambi");
        dbfcmd(sqltemp, " from tip5_cambio, pedido");
        dbfcmd(sqltemp, " where tipo_cambio.mon_clave = pedido.mon_clave");
        dbfcmd(sqltemp, " and alm_clave = '%s'", alm_clave);
        dbfcmd(sqltemp, " and ped_clave = '%d'", ped_clave);
        dbfcmd(sqltemp, " and ped_num_conse = '%d'", ped_num_conse);
        dbfcmd(sqltemp, " and tca_fec_cotiz = convert(char(12), ped_fec_elabo);
        dbsqlxec(sqltemp);
        if (dbresults(sqltemp) == SUCCEED)
        {
          dbbind(sqltemp, 1, REALBIND, (DBINT)0, (BYTE *) &tca_tip_cambi_ped);
          while(dbnextraw(sqltemp) != NO_MORE_ROWS);
          if (DBCOUNT(sqltemp) == 0)
            tca_tip_cambi = 0;
          else
            if (strcmp(mon_clave, "US") != 0)
            {
              dbfcmd(sqltemp, " select tca_tip_cambi");
              dbfcmd(sqltemp, " from tip5_cambio, pedido");
              dbfcmd(sqltemp, " where tipo_cambio.mon_clave = 'US'");
              dbfcmd(sqltemp, " and alm_clave = '%s'", alm_clave);
              dbfcmd(sqltemp, " and ped_clave = '%d'", ped_clave);
              dbfcmd(sqltemp, " and ped_num_conse = '%d'", ped_num_conse);
              dbfcmd(sqltemp, " and tca_fec_cotiz = convert(char(12), ped_fec_elabo);
              dbsqlxec(sqltemp);
              if (dbresults(sqltemp) == SUCCEED)
              {
                dbbind(sqltemp, 1, REALBIND, (DBINT)0, (BYTE *) &tca_tip_cambi_ped);
                while(dbnextraw(sqltemp) != NO_MORE_ROWS);
                if (DBCOUNT(sqltemp) == 0)
                  tca_tip_cambi = 0;
                else
                  tca_tip_cambi = tca_tip_cambi_ped * tca_tip_cambi_US;
              }
            }
          }
        }
      }
    }
  }
  tca_tip_cambi = tca_tip_cambi_ped;
}
}
}

```

```

}
void nom_com(char *cve_com, char *nom_compr, char *clave_ofi)
{
    nom_compr[0] = NULL;
    dbfcmd(sqltemp, " select com nombre                ";
    dbfcmd(sqltemp, "   from comprador                ";
    dbfcmd(sqltemp, "   where ofi_clave = '%s' ",clave_ofi );
    dbfcmd(sqltemp, "   and com_clave = '%s' ",cve_com );
    dbsqlxex(sqltemp);
    if (dbresults(sqltemp) == SUCCEED)
    {
        dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, nom_compr);
        while(dbnextrrow(sqltemp) != NO_MORE_ROWS)
            roman_PC(nom_compr);
    }
}

void proveedor(void)
{
    void domicilio(DBCHAR *rfc_pro, DBTINYINT num_dom);
    void telefono(DBCHAR *rfc_pro);
    int ind_c;
    prv_rzo_socia[0] = NULL; prv_rzo_socia_r[0] = NULL;
    pdo_calfe[0] = NULL; pdo_colonia[0] = NULL;
    pdo_dlg_munic[0] = NULL; pdo_estado[0] = NULL;
    pdo_pais[0] = NULL; pdo_cod_posta[0] = NULL;
    dbfcmd(sqlpro, " select prv_rzo_socia,                ";
    dbfcmd(sqlpro, "   prv_nacionali                ");
    dbfcmd(sqlpro, "   from proveedor                ");
    dbfcmd(sqlpro, "   where prv_rfc = '%s' ",prv_rfc );
    dbsqlxex(sqlpro);
    if (dbresults(sqlpro) == SUCCEED)
    {
        dbbind(sqlpro, 1, STRINGBIND, (DBINT)0, prv_rzo_socia);
        dbbind(sqlpro, 2, STRINGBIND, (DBINT)0, prv_nacionali);
        while(dbnextrrow(sqlpro) != NO_MORE_ROWS);
        if (DBCOUNT(sqlpro) != 0)
        {
            roman_PC(prv_rzo_socia);
            if (strcmp(prv_nacionali, "L") == 0)
                strcpy(prv_nacionali, "N");
            if ((strcmp(prv_rfc, prv_rfc_repre) != 0)
            {
                dbfcmd(sqlpro, " select prv_rzo_socia                ");
                dbfcmd(sqlpro, "   from proveedor                ");
                dbfcmd(sqlpro, "   where prv_rfc = '%s' ",prv_rfc_repre );
                dbsqlxex(sqlpro);
                if (dbresults(sqlpro) == SUCCEED)
                {
                    dbbind(sqlpro, 1, STRINGBIND, (DBINT)0, prv_rzo_socia_r);
                    while(dbnextrrow(sqlpro) != NO_MORE_ROWS);
                    if (DBCOUNT(sqlpro) != 0)
                    {
                        roman_PC(prv_rzo_socia_r);
                        domicilio(prv_rfc_repre, pdo_num_conse);
                        telefono(prv_rfc_repre);
                    }
                }
            }
        }
    }
    else
    {
        domicilio(prv_rfc, pdo_num_conse);
        telefono(prv_rfc);
    }
    nom_cnd_pago[0] = NULL;
    if (strlen(prv_cnd_pago) != 0)
    {
        tabla_detalle(1,prv_cnd_pago, nom_cnd_pago);
        if ((ind_c = find_substring(nom_cnd_pago, "XXX")) != -1)
        {
            kill_substring(nom_cnd_pago, ind_c, 3);
            sprintf(sprv_dia_condi_pago, "%3d",prv_dia_condi_pago);
            insert_substring(nom_cnd_pago,sprv_dia_condi_pago,ind_c);
        }
    }
}
}

```



```

while(dbnextrrow(sqltemp) != NO_MORE_ROWS);
}
if {strcmp(ped_status, "PC") == 0}
{
sub total = 0.00;
total_partidas = 0.00;
}
dbfcmd(sqltemp, " select sub_total=sum((isnull(emb_cantidad,0.00) * ");
dbfcmd(sqltemp, " isnull(emb_pre_unita,$0.00)) - ");
dbfcmd(sqltemp, " (isnull(emb_cantidad,0.00) * ");
dbfcmd(sqltemp, " isnull(emb_pre_unita,$0.00)) * ");
dbfcmd(sqltemp, " (isnull(ppe_ptj_dcto,0.00) / 100)), ");
dbfcmd(sqltemp, " total_part=sum(isnull(emb_cantidad,0.00) * ");
dbfcmd(sqltemp, " isnull(emb_pre_unita,$0.00)) ");
dbfcmd(sqltemp, " from partida_pedido pp, embarque em ");
dbfcmd(sqltemp, " where em.alm_clave = 'A' ", alm_clave ");
dbfcmd(sqltemp, " and em.ped_clave = 'd' ", ped_clave ");
dbfcmd(sqltemp, " and em.ped_num_conse = 'd' ", ped_num_conse ");
dbfcmd(sqltemp, " and pp.alm_clave = em.alm_clave ");
dbfcmd(sqltemp, " and pp.ped_clave = em.ped_clave ");
dbfcmd(sqltemp, " and pp.ped_num_conse = em.ped_num_conse ");
dbfcmd(sqltemp, " and pp.ppe_num_conse = em.ppe_num_conse ");
dbfcmd(sqltemp, " and ppe_cve_progr = 'S' ");
}
}
dbsexec(sqltemp);
if {dbresults(sqltemp) == SUCCEED}
{
dbbind(sqltemp, 1, REALBIND, (DBINT)0, &sub_total_prog);
dbbind(sqltemp, 2, REALBIND, (DBINT)0, &total_partidas_prog);
while(dbnextrrow(sqltemp) != NO_MORE_ROWS);
}
if {strcmp(ped_status, "PC") == 0}
{
sub total_prog = 0.00;
total_partidas_prog = 0.00;
}
sub total += sub total_prog;
total_partidas += total_partidas_prog;
total_desc = sub total * (ped_pje_dcto / 100.00);
total_imp = (sub total - total_desc) * (ped_pje_impto / 100.00);
total_l = (sub total - total_desc) + total_imp;
if {strcmp(mon_clave, "MN") == 0}
{
tca_tip_cambi = 1.0;
total_mn = total_l * tca_tip_cambi;
if {strcmp(mon_clave, "MN") == 0}
{
tca_tip_cambi = 0.0;
}
}
}

```

```

void domicilio(DBCHAR *rfc_pro, DBTINYINT dom_num)

```

```

{
dbfcmd(sqltemp, " select pdo_calle, ");
dbfcmd(sqltemp, " pdo_colonia, ");
dbfcmd(sqltemp, " pdo_dlg_munic, ");
dbfcmd(sqltemp, " pdo_cod_posta, ");
dbfcmd(sqltemp, " pdo_estado, ");
dbfcmd(sqltemp, " pdo_pais ");
dbfcmd(sqltemp, " from domicilio_proveedor ");
dbfcmd(sqltemp, " where prv_rfc = 'A' ", rfc_pro );
dbfcmd(sqltemp, " and pdo_num_conse = 'd' ", dom_num );
}
dbsexec(sqltemp);
if {dbresults(sqltemp) == SUCCEED}
{
dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, pdo_calle);
dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, pdo_colonia);
dbbind(sqltemp, 3, STRINGBIND, (DBINT)0, pdo_dlg_munic);
dbbind(sqltemp, 4, STRINGBIND, (DBINT)0, pdo_cod_posta);
dbbind(sqltemp, 5, STRINGBIND, (DBINT)0, pdo_estado);
dbbind(sqltemp, 6, STRINGBIND, (DBINT)0, pdo_pais);
while(dbnextrrow(sqltemp) != NO_MORE_ROWS);
if {DBCOUNT(sqltemp) != 0}
{
roman_PC(pdo_calle);
roman_PC(pdo_colonia);
roman_PC(pdo_dlg_munic);
roman_PC(pdo_cod_posta);
strcpy(dom_nom_esta, " ");
strcpy(dom_nom_pais, " ");
}
}
}

```

```

        if ((strlen(pdo_pais) != 0)
            tabla_detalle(16, pdo_pais, dom_nom_pais);
        if ((strlen(pdo_estado) != 0)
            tabla_detalle(19, pdo_estado, dom_nom_esta);
    }
}

void telefono(DBCHAR *rfc_pro)
{
    dbfcmd(sqltemp, " select prv_telefono,                ");
    dbfcmd(sqltemp, "     prv_fax                        ");
    dbfcmd(sqltemp, "   from telefono proveedor        ");
    dbfcmd(sqltemp, "     where   prv_rfc = '%s' ", rfc_pro);
    dbfcmd(sqltemp, "     and   prv_cve_prove = 'P' ");
    dbqlxec(sqltemp);
    if (dbresults(sqltemp) == SUCCEED)
    {
        dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, prv_telefono);
        dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, prv_fax);
        while(dbnextrow(sqltemp) != NO_MORE_ROWS);
        if (DBCOUNT(sqltemp) != 0)
        {
            roman_PC(prv_telefono);
            roman_PC(prv_fax);
        }
    }
}

int descripcion(void)
{
    int i = 0, r = 0, c = 0;
    for(r = 0, c = 0; r < 1000; r++)
        des_parte[r][c] = NULL;
    r = 0;
    c = 0;
    do
    {
        /* if (c == 0 && des_nombre2[i] == ' ')
           continue;
           if (des_nombre2[i] == '\n')
           {
               des_parte[r][c] = NULL;
               r++;
               c = 0;
           }
           else
           {
               des_parte[r][c] = des_nombre2[i];
               des_parte[r][c + 1] = NULL;
               if (strlen(des_parte[r]) > 35)
               {
                   r++;
                   c = 0;
               }
               else
                   c++;
           }
           } while (des_nombre2[i++] != NULL && r <= 1000);
           if (strlen(des_parte[r]) == 0)
               r--;
           return r;
       }

int descripcion2(void)
{
    int i=0, r=0, c=0;
    for(r = 0, c = 0; r < 1000; r++)
        des_notas[r][c] = NULL;
    r = 0;
    c = 0;
    do
    {
        /* if (c == 0 && des_nombre2[i] == ' ')
           continue;
           if (des_nombre2[i] == '\n')
           */

```

```

    {
        des_nota[r][c] = NULL;
        r++;
        c = 0;
    }
    else
    {
        des_nota[r][c] = des_nombre2[i];
        des_nota[r][c + 1] = NULL;
        if (strlen(des_nota[r]) > 71)
        {
            r++;
            c = 0;
        }
        else
            c++;
    }
} while(des_nombre2[i++] != NULL && r <= 1000);
if (strlen(des_nota[r]) == 0)
    r--;
return r;
}

void direction_office(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)
        dom_ofna_grl[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (dom_ofna_grl[r][c] = ofi_dom_regio[i]) != NULL && r < 10;
        {
            if (ofi_dom_regio[i] == '\n')
            {
                dom_ofna_grl[r++][c] = NULL;
                c = -1;
                continue;
            }
            if (c > 34)
            {
                dom_ofna_grl[r++][++c] = NULL;
                c = -1;
            }
        }
    for(r = 0; r < 10; r++)
        if ((strlen(dom_ofna_grl[r])) > 32)
            dom_ofna_grl[r][33] = NULL;
}

void direction_corres(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)
        dom_corres[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (dom_corres[r][c] = ofi_dom_corre[i]) != NULL && r < 10; i+
        {
            if (ofi_dom_corre[i] == '\n')
            {
                dom_corres[r++][c] = NULL;
                c = -1;
                continue;
            }
            if (c > 34)
            {
                dom_corres[r++][++c] = NULL;
                c = -1;
            }
        }
    for(r = 0; r < 10; r++)
        if ((strlen(dom_corres[r])) > 32)
            dom_corres[r][33] = NULL;
}

void alm_direction_fact(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)

```

```

dom factur[r][c] = NULL;
for(i = 0, r = 0, c = 0; (dom_factur[r][c] = ald_nombre_fac[i]) != NULL && r < 10; i)
{
    if (ald_nombre_fac[i] == '\n')
    {
        dom_factur[r++][c] = NULL;
        c = -1;
        continue;
    }
    if (c > 40)
    {
        dom_factur[r++][++c] = NULL;
        c = -1;
    }
}
}

```

```
void cia_direction(void)
```

```

{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)
        dom_fisca[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (dom_fisca[r][c] = cia_dom_fisca[i]) != NULL && r < 10; i++)
    {
        if (cia_dom_fisca[i] == '\n')
        {
            dom_fisca[r++][c] = NULL;
            c = -1;
            continue;
        }
        if (c > 40)
        {
            dom_fisca[r++][++c] = NULL;
            c = -1;
        }
    }
}
}

```

```
void alm_direction_cons(void)
```

```

{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)
        dom_consigr[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (dom_consigr[r][c] = ald_nombre_con[i]) != NULL && r < 10; i)
    {
        if (ald_nombre_con[i] == '\n')
        {
            dom_consigr[r++][c] = NULL;
            c = -1;
            continue;
        }
        if (c > 33)
        {
            dom_consigr[r++][++c] = NULL;
            c = -1;
        }
    }
}
}

```

```
void def_char_ctl(void)
```

```

{
    int i;
    if (export_to == 0)
    {
        Q[2] = NULL; W[2] = NULL; A[2] = NULL; S[2] = NULL;
        LV[3] = NULL; LH[1] = NULL;
        LVGI[2] = NULL; LVGD[2] = NULL; LHGS[1] = NULL; LHGI[1] = NULL; CRS[1] = NULL;
        Q[0] = 14; A[0] = 14;
        LV[0] = 14;
        LVGI[0] = 14;

        LV[2] = 15; W[1] = 15; S[1] = 15;

        Q[1] = 'R'; W[0] = 'T'; A[1] = 'F'; S[0] = 'G';
        LV[1] = '.'; LH[0] = ',';
        LVGI[1] = '5'; LVGD[0] = '6'; LHGS[0] = '7'; LHGI[0] = '8'; CRS[0] = '/';
    }
}

```

```

}
else
{
  Q[1] = NULL; W[1] = NULL; A[1] = NULL; S[1] = NULL;
  LV[1] = NULL; LH[1] = NULL;
  LVGI[1] = NULL; LVGD[1] = NULL; LHGS[1] = NULL; LHGI[1] = NULL; CRS[1] = NULL;
  Q[0] = 218; W[0] = 191; A[0] = 192; S[0] = 217;
  LV[0] = 179; LH[0] = 196;
  LVGI[0] = 195; LVGD[0] = 180; LHGS[0] = 194; LHGI[0] = 193; CRS[0] = 197;
}
for(i=0;i<80;i++)
  filler[i] = LH[0];
filler[80] = NULL;
}

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sybfront.h>
#include <sybdb.h>
#include <syberror.h>
#include "users/logistica/fuentes_c/include/sybutil.h"
#include "users/logistica/fuentes_c/include/printer.h"
#include "users/logistica/fuentes_c/include/util.h"

/*****
Fecha: Noviembre 11 de 1992.
Programa: abctsol01_1.c
Descripcion: Genera el formato de la impresion de la Solicitud de
Cotizacion en español e inglés.
*****/
char uname[20], upase[20], server[20], mycommand[20], file_name[51],
option[10], letr0[50], letr1[52], letr2[3][16], file_name2[51],
letr3[8] = "PEÑALES", letr4[7] = "México",
letr5[10] = "recopición";

struct tm *today_date, *localtime();
time_t time_t, seconds;
size_t strftime();
int cnt_page, cnt_line, limit_page,
export to, par_not, f_not,
acuse;

int MAX_LINES 1 = 44, MAX_LINES 2 = 44;
char daFe[33], y_o[4], dom_ofna_gri[10][35],
dom_corras[10][35], dom_factur[10][43], dom_consIg[10][35],
des_parte[1000][37], dom_nom_esta[61], dom_nom_pais[61],
nom_via_cons[61], nom_cnd_pago[61], nom_pfi[14],
com_nombre[41], sppe_num_conse[3], sppe_cantidad[31],
sppp_pre_unita[31], sppe_total_partida[31], asub_total[31],
sped_pje_imppto[41], stotal_imp[31], nom_imppto[17],
ped_nom_autori[41], sreq_num_conse[6], sprq_num_conse[3],
com_ofi_clave[5], des_nota[500][73], filfer[81];

char *des_nombre2;
DBPROCESS *sqltemp, *sqlped, *sqlppe,
*sqlreq, *sqlpro, *sqlppr;
LOGINREC *login;
DBINT count_office;
FILE *file_output;

/**** Secuencias de control de impresion ****/
char device[20];
int copies;
unsigned char Q[4], W[4], A[4], S[4],
/* For Pc's . =218, . =191, . =192, . =217 */
/* For HP-9000 Q, W, A, S */
LV[4], LH[4],
/* For Pc's . =179, - =196 */
/* For HP-9000 , , , ,
LVGI[4], LVGD[4], LHGS[4], LHGI[4], CRS[4];
/* For Pc's . =180, . =195, . =194, . =193, . =197 */
/* For HP-9000 1, 2, 3, 4, 0 */

/**** Totales en importes ****
DBREAL sub_total, total_desc, total_imp,
total_1, tca_tIp_cambi, total_mn,
total_partidas;
DBCHAR total_mo[31], valor_mn[31], tip_cam[31],
valor_me[31];
DBSHALLINT partida;

```

```
/*** Datos de La oficina regional */
```

```
DBCHAR ofi_clave[5], ofi_nombre[41], ofi_dom_regio[109],  
ofi_dom_corre[109];
```

```
/*** Datos del Almacen */
```

```
DBCHAR alm_nombre[41], alm_nom_respo[41], cia_nombre[41],  
alm_clave[5], alm_clave[5], cia_nombre2[41],  
uni_clave[5], alm_nombre_fac[109], alm_nombre_con[109];
```

```
/*** Datos de la Solicitud */
```

```
DBCHAR sco_fec_sol[9], sco_fec_lim_cotiz[9];  
DBINT sco_num_conse;
```

```
/*** Datos de la Requisicion */
```

```
DBCHAR req_aut_plant[3], req_des_gener[256], req_fec_elabo[9],  
req_fec_recep[9], req_fec_asign[9], req_fec_reque[9],  
req_tip_servi[2], req_cod_prior[2], req_nom_solic[26],  
req_ord_traba[7], req_num equip[13], req_dep_nombre[21],  
nta_clave[15], com_clave[3], break_nra_clave[15];  
DBSMALLINT alm_cta_cuent, alm_cta_conce,  
DBTINYINT alm_cta_subct, alm_cta_ssubc, req_aut_perio;  
DBINT req_aut_folio, req_num_conse, rreq_num_conse;  
DBREAL req_imp_estim;
```

```
/*** Datos de las Partidas de la Requisicion */
```

```
DBCHAR mun_clave[13], lin_sub_linea[7], mco_clave[15],  
ume_unimed[6], des_nombre[256],  
DBREAL prq_can_reque, prq_can_coloc;  
DBSMALLINT prq_num_conse;  
DBTINYINT lin_clave, prq_num_desc;  
DBINT des_clave;
```

```
/*** Datos del Programa de la Partidas de la Requisicion */
```

```
DBCHAR emb_fec_entre[9];  
DBREAL emb_canEidad;  
DBSMALLINT emb_num_conse;
```

```
/*** Datos del Proveedor */
```

```
DBCHAR prv_rzo_socia[41], prv_rzo_socia_r[41], prv_nacionali[2],  
prv_rfc[21], prv_rfc_repre[21], pdo_atencion[41];  
DBTINYINT pdo_num_conse;
```

```
/*** Datos del Domicilio del proveedor u representante */
```

```
DBCHAR pdo_calle[41], pdo_colonia[41], pdo_dig_munic[41],  
pdo_estado[5], pdo_pais[5], pdo_cod_posta[14],  
pdo_fax[15], pdo_telefono[15];
```

```
int ctoi(char *cve, int no_arg)
```

```
{  
    char x;  
    int r_value=0;  
    while((x = *cve++) != NULL)  
        if (x >= '0' && x <= '9')  
            r_value = r_value * 10 + x - '0';  
        else  
        {  
            if (no_arg == 2)  
                printf("El Numero de la Req. debe de ser numerico...\n");  
            if (no_arg == 3)  
                printf("El Numero de la Req. debe de ser numerico...\n");  
            exit(4);  
        }  
    return r_value;  
}
```

```
void tabla_detalle(int num_tab, char *cve_tab, char *elemento)
```



```

{
    let1[0] = NULL;
    let2[0][0] = NULL;
    let2[1][0] = NULL;
    let2[2][0] = NULL;
}
else
if (strcmp(prv_nacional1, "N") == 0)
{
    strcpy(let1, "                -OFICINAS GENERALES-
CORRESPONDENCIA A:");
    strcpy(let2[0], "CITENSE ESTOS");
    strcpy(let2[1], "NUMEROS EN TODA");
    strcpy(let2[2], "CORRESPONDENCIA");
}
else
{
    strcpy(let1, "                -MAIN OFFICE-
MAIL TO:                ");
    strcpy(let2[0], "ALWAYS REFER TO");
    strcpy(let2[1], "THIS NUMBER ON");
    strcpy(let2[2], "ALL DOCUMENTS ");
}
if (cnt page++ > 0)
printf(file_output, "\f\r");
else
if (export_to == 0)
{
    lpl;
    tl;
    lpi8;
    enable_draw;
}
cnt line = 0;
if (export_to == 0)
{
    fprintf(file_output, "\n
big_letter;
}
else
{
    fprintf(file_output, "
big_letter;
}
fprintf(file_output, "%40.40-s", cia_nombre2);
if (export_to == 0)
    normal_letter;
else
    lnormal_letter;
if (acuse == 1)
    fprintf(file_output, "\n");
fprintf(file_output, "
%40.40-s\n", alm_nombre);
if (acuse == 1)
    fprintf(file_output, "\n
%49.49-s\n", let1[0]);
else
    fprintf(file_output, "
%49.49-s\n", let1[0]);
if (acuse == 0 && export_to == 1)
    lnormal_letter;
fprintf(file_output, "%51.51-s\n", let1);
if (acuse == 1)
    fprintf(file_output, "
");
fprintf(file_output, "%32.32-s %32.32-s %15.15-s\n", dom_ofna_grl[0], dom_corres[0]);
if (acuse == 1)
    fprintf(file_output, "
");
fprintf(file_output, "%32.32-s %32.32-s %15.15-s\n", dom_ofna_grl[1], dom_corres[1]);
if (acuse == 1)
    fprintf(file_output, "
");
fprintf(file_output, "%32.32-s %32.32-s %15.15-s\n", dom_ofna_grl[2], dom_corres[2]);
if (acuse == 0)
{
    if (export_to == 0)
        big_letter;
    else
    {
        xbig_letter;
        fprintf(file_output, "\n
");
    }
}
if (strcmp(prv_nacional1, "N") == 0)

```

```

        fprintf(file_output, "
else
    fprintf(file_output, "
    if (export to == 0)
        normal_letter;
    else
        lnormal_letter;
else
    {
        fprintf(file_output, "\n\n\n\n");
        return;
    }
fprintf(file_output, "%s44.44%s%s8.8%s%s8.8%s%s14.14%s\n", Q, filler, W, Q, fill
strcpy(y_o, " ");
if (strcmp(prv_rfc, prv_rfc_repre) != 0)
    strcpy(y_o, "y/o");
/* Renglon 1 */
if (strcmp(prv_nacionali, "N") == 0)
    fprintf(file_output, "%sPROVEEDOR:                %20.20-%s%sFECHA DE%sLIHITE F
else
    fprintf(file_output, "%sSUPPLIER:                %20.20-%s%sDATE OF %sQUOTING
/* Renglon 2 */
if (strcmp(prv_nacionali, "N") == 0)
    fprintf(file_output, "%s %39.39-%s %3.3-%s%sEMISION %sCOTIZAR. %s14.14%s\n"
else
    fprintf(file_output, "%s %39.39-%s %3.3-%s%sISSUE %sD. LIMIT%s14.14%s\n"
/* Renglon 3 */
fprintf(file_output, "%s %40.40-%s %s8.8%s%s8.8%s %4.4-%s %5.5d %s\n", LV,
/* Renglon 4 */
if (strcmp(prv_nacionali, "N") == 0)
    fprintf(file_output, "%s %40.40-%s %s8.8-%s%s8.8-%s%s5.5d HOJA: %2d%s\n"
else
    fprintf(file_output, "%s %40.40-%s %s8.8-%s%s8.8-%s%s5.5d PAGE: %2d%s\n"
/* Renglon 5 */
fprintf(file_output, "%s %40.40-%s %s8.8%s%s8.8%s%s14.14%s\n", LV, pdo_colon
if (strcmp(prv_nacionali, "N") == 0)
    {
/* Renglon 6 */
    fprintf(file_output, "%s %32.32-%s C.P. %5.5-%s%s8.8%s%s8.8%sFAVOR COTIZAR
/* Renglon 7 */
    fprintf(file_output, "%s %20.20-%s %20.20-%s %sENTREGA %sENTREGA %sLAB FABRI
/* Renglon 8 */
    fprintf(file_output, "%sATTN: %38.38-%s%sREQUERID%sOPRECIDA%sUS MEJORES:
/* Renglon 9 */
    fprintf(file_output, "%sTelefono: %14.14-%s Fax: %14.14-%s%s8.8%s%s8.8-%s -P
/* Renglon 10 */
    fprintf(file_output, "%s
    fprintf(file_output, "%s44.44%s%s8.8%s%s8.8-%s -TIEMPO ENTREGA\n", A, filler
    }
else
/* Renglon 6 */
    fprintf(file_output, "%s %32.32-%s C.P. %5.5-%s%s8.8%s%s8.8%sPLEASE QUOTE
/* Renglon 7 */
    fprintf(file_output, "%s %20.20-%s %20.20-%s %sMATERIAL%sR BEST %s\n", LV, do
/* Renglon 8 */
    fprintf(file_output, "%sATTN: %38.38-%s%sREQD. BY: %sDEL. DATE %sYOUR BEST:
/* Renglon 9 */
    fprintf(file_output, "%sPhone: %14.14-%s Fax: %14.14-%s %s8.8%s%s8.8-%s-PR
/* Renglon 10 */
    fprintf(file_output, "%s
    fprintf(file_output, "%s44.44%s%s8.8%s%s8.8-%s-DELIVERY TIME\n", A, filler,
    }
fprintf(file_output, "%s26.26%s%s16.16%s%s34.34%s\n", Q, filler, W, Q, filler, W, Q,
if (strcmp(prv_nacionali, "N") == 0)
    {
    fprintf(file_output, "%sCONDICIONES DE PAGO: %sDESC. POR P/PAGO%sCONDI
    fprintf(file_output, "%s %s % A DIAS %s
    }
else
    {
    fprintf(file_output, "%sPAYMENT TERMS: %sPROMPT PAY. DSCT%sDELIV
    fprintf(file_output, "%s %s % AT DAYS %s
    }
fprintf(file_output, "%s26.26%s%s16.16%s%s34.34%s\n", A, filler, S, A, filler, S, A,

```



```

    }
}
void check_page2(void)
{
  if ((cnt_page == 1 && cnt_line > 12) || (cnt_page > 1 && cnt_line > 49))
  {
    if (cnt_page == 1)
      limit_page = MAX_LINES_1;
    else
      limit_page = MAX_LINES_2;
    for(;cnt_line <= limit_page;++cnt_line)
      if (paF not == 0)
        blank_detail_1();
      else
        blank_detail_2();
    footing_page();
    heading_page();
  }
  else
  {
    footing_partidas();
    cnt_line++;
    heading_notes();
  }
}

```

```

void acuse_recibo(void)
{
  char mesx[13][11], part[3];
  int dia, mes, alo;
  part[2] = NULL;
  part[0] = sco_fec_sol[0];
  part[1] = sco_fec_sol[1];
  dia = (DBTINYINT) atoi(part, 0);
  part[0] = sco_fec_sol[3];
  part[1] = sco_fec_sol[4];
  mes = (DBTINYINT) atoi(part, 0);
  part[0] = sco_fec_sol[6];
  part[1] = sco_fec_sol[7];
  alo = (DBTINYINT) atoi(part, 0);
  strcpy(mesx[0], "");
  if (strcmp(prv_nacionali, "N") == 0)
  {
    strcpy(mesx[1], "ENERO");
    strcpy(mesx[2], "FEBRERO");
    strcpy(mesx[3], "MARZO");
    strcpy(mesx[4], "ABRIL");
    strcpy(mesx[5], "MAYO");
    strcpy(mesx[6], "JUNIO");
    strcpy(mesx[7], "JULIO");
    strcpy(mesx[8], "AGOSTO");
    strcpy(mesx[9], "SEPTIEMBRE");
    strcpy(mesx[10], "OCTUBRE");
    strcpy(mesx[11], "NOVIEMBRE");
    strcpy(mesx[12], "DICIEMBRE");
  }
  else
  {
    strcpy(mesx[1], "JANUARY");
    strcpy(mesx[2], "FEBRUARY");
    strcpy(mesx[3], "MARCH");
    strcpy(mesx[4], "APRIL");
    strcpy(mesx[5], "MAY");
    strcpy(mesx[6], "JUNE");
    strcpy(mesx[7], "JULY");
    strcpy(mesx[8], "AUGUST");
    strcpy(mesx[9], "SEPTEMBER");
    strcpy(mesx[10], "OCTOBER");
    strcpy(mesx[11], "NOVEMBER");
    strcpy(mesx[12], "DECEMBER");
  }
  if (export to == 1)
    l6normal letter;
  if (strcmp(prv_nacionali, "N") == 0)

```



```

kill substrng(argv[x],0,2);
if ({Copias = (DBTINYINT) atoi(argv[x], 3) > 99)
    {
        printf("El Valor maximo de las Copias es 99.\n");
        exit(1);
    }
}
if {strchr(argv[x], "-U") != NULL}
    {
        strcpy(uname, argv[x]);
        kill_substrng(uname,0,2);
    }
if {strchr(argv[x], "-P") != NULL}
    {
        strcpy(upass, argv[x]);
        kill_substrng(upass,0,2);
    }
if {strchr(argv[x], "-S") != NULL}
    {
        strcpy(server, argv[x]);
        kill_substrng(server,0,2);
    }
if {strchr(argv[x], "-R") != NULL}
    {
        strcpy(prv_rfc, argv[x]);
        kill_substrng(prv_rfc,0,2);
        to_upper(prv_rfc);
        if (strlen(prv_rfc) > 20)
            {
                printf("El R.F.C del proveedor no debe exeder de 20 caracteres.\n");
                exit(2);
            }
    }
if {strchr(argv[x], "-C") != NULL}
    {
        kill_substrng(argv[x],0,2);
        if ({sco_num_conse = (DBINT)          atoi(argv[x], 1) > 99999}
            {
                printf("El Valor maximo del Numero del Pedido es 99999.\n");
                exit(3);
            }
    }
if {strchr(argv[x], "-F") != NULL}
    {
        strcpy(file_name, argv[x]);
        kill_substrng(file_name,0,2);
        strcat(file_name, ".rpt");
    }
}
if {strlen(device) == 0}
    {
        printf("El dispositivo de salida es requerido...\n");
        exit(7);
    }
if {export_to == 0}
    {
        printf("El tipo de Impresora (-t) es requerido...\n");
        exit(7);
    }
}
export to--;
if {export_to == 1}
    {
        MAX_LINES_1 -- 4;
        MAX_LINES_2 -- 4;
    }
if {strlen(uname) == 0}
    {
        printf("El usuario es requerido...\n");
        exit(5);
    }
if {strlen(upass) == 0}
    {
        printf("El password es requerido...\n");
        exit(7);
    }
}
if {strlen(prv_rfc) == 0}

```

```

{
    printf("El R.F.C. del proveedor es requerido...\n");
    exit(8);
}
if {sco_num_conse > 99999}
{
    printf("La Clave del Pedido es requerido...\n");
    exit(9);
}
par not = 0;
f not = 0;
if (dbinit() == FAIL)
    exit (ERREXIT);
dberrhandle(orr_handler);
dbmsghandle(msg_handler);
login = dblogin();
DBSETLPWD(login,upass);
DBSETLUSER(login,uname);
sqltemp = dbopen(login, server);
sqlped = dbopen(login, server);
sqlppe = dbopen(login, server);
sqlreq = dbopen(login, server);
sqlpro = dbopen(login, server);
sqlppr = dbopen(login, server);
dbloginfree(login);
if ((count_office = detect_office()) != 1)
{
    if (count_office == 0)
        printf("No hay Oficina operando.\n");
    else
        printf("Hay mas de una Oficina operando.\n");
    exit(2);
}
if {strlen(file_name) == 0}
{
    strcpy(rfc,prv_rfc);
    while(rfc[strlen(rfc) - 1] == ' ')
        kill_substring(rfc,strlen(rfc) - 1, 1);
    pos subs = find_substring(rfc," ");
    kill_substring(rfc,pos subs,1);
    insert_substring(rfc,"-",pos subs);
    sprintf(file_name,"%s_d.rpt.%s",rfc,sco_num_conse,uname);
}
lee_solicitud();
detect_almacen();
detect_proveedor();
almacen_compania();
roman_PC(letr3);
roman_PC(letr4);
roman_PC(letr5);
requisicion();
proveedor();
/* if ((strlen(ofi_dom_corre) == 0)
    strcpy(ofi_dom_corre, ofi_dom_regio); */
direction_office();
direction_corres();
cnt_page = 0;
def_char_ctl();
seconds = time(NULL);
today_date = localtime(5seconds);
strftime(date, (size_t)32, "FECHA: %b %d %Y, HORA: %H:%M:%S", today_date);
if ((file_output = fopen(file_name,"w")) == NULL)
{
    printf("Error al Abrir archivo %s\n",file_name);
    exit(8);
}
heading_page();
parida_requisicion();
check_limits();
footing_page();
/* fprintf(file_output,"%f"); */
if {fclose(file_output) != NULL}
{
    printf("Error al Cerrar archivo %s\n",file_name);
    exit(9);
}
}

```

```

sprintf(mycommand,"lp -d%-s -n%-d -s -onb %s",device,copies,file_name);
system(mycommand);
for(x=1;x<=argc;x++)
{
if (strstr(argv[x], "-O") != NULL)
{
strcpy(option, argv[x]);
to_upper(option);
if (strcmp(option, "-OREMOVE") == 0)
{
kill_substring(argv[x],0,2);
remove(file_name);
continue;
}
}
if (strcmp(argv[x], "-OAR") == 0)
{
strcpy(file_name2, file_name);
strcat(file_name2, ".AR");
if ((file_output = fopen(file_name2,"w")) == NULL)
{
printf("Error al Abrir archivo %s\n",file_name2);
exit(8);
}
acuse = 1;
cnt_page = 0;
heading_page();
acuse_recibo();
if (fclose(file_output) != NULL)
{
printf("Error al Cerrar archivo %s\n",file_name2);
exit(9);
}
sprintf(mycommand,"lp -d%-s -s -onb %s",device,file_name2);
system(mycommand);
remove(file_name2);
break;
}
}
}
printf("\a");
dbexit();
exit(0);
}
void lee_solicitud(void)
{
dbfcmd(sqltemp, " select sco_fec_sol = convert(char(8), sco_fec_sol, 3), ");
dbfcmd(sqltemp, " sco_fec_lim_cotiz = convert(char(8), sco_fec_lim_cotiz, 3) ");
dbfcmd(sqltemp, " from solicitud_cotizacion ");
dbfcmd(sqltemp, " where sco_num_conse = %d ", sco_num_conse );
dbsqlxec(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, sco_fec_sol);
dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, sco_fec_lim_cotiz);
while(dbnextrow(sqltemp) != NO_MORE_ROWS);
}
}
void detect_almacen(void)
{
dbfcmd(sqltemp, " select distinct alm_clave ");
dbfcmd(sqltemp, " from solicitud_partida_requisicion ");
dbfcmd(sqltemp, " where sco_num_conse = %d ", sco_num_conse );
dbsqlxec(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, alm_clave);
while(dbnextrow(sqltemp) != NO_MORE_ROWS);
}
}
void detect_proveedor(void)
{
dbfcmd(sqltemp, " select prv_rfc repre, pdo_num_conse ");
dbfcmd(sqltemp, " from solicitud_proveedor ");
dbfcmd(sqltemp, " where sco_num_conse = %d ", sco_num_conse );
dbfcmd(sqltemp, " and prv_rfc = '%s' ",prv_rfc );
}

```

```

dbsqllexac(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
    dbbind(sqltemp, 1, STRINGBIND, (DBINT)0,                prv_rfc_repre);
    dbbind(sqltemp, 2, TINYBIND, (DBINT)0, (BYTE *) &pdo_num_conse);
    while(dbnextrow(sqltemp) != NO_MORE_ROWS);
}
}

DBINT detect_office(void)
{
    dbfcmd(sqltemp, " select ofi_clave, ofi_nombre, ofi_dom_regio, ";
    dbfcmd(sqltemp, " ofi_dom_corr ";
    dbfcmd(sqltemp, " from ofi regional ";
    dbfcmd(sqltemp, " where ofi_ind_opera = 1 ";
    dbsqllexac(sqltemp);
    if (dbresults(sqltemp) == SUCCEED)
    {
        dbbind(sqltemp, 1, STRINGBIND, (DBINT)0,                ofi_clave);
        dbbind(sqltemp, 2, STRINGBIND, (DBINT)0,                ofi_nombre);
        dbbind(sqltemp, 3, STRINGBIND, (DBINT)0,                ofi_dom_regio);
        dbbind(sqltemp, 4, STRINGBIND, (DBINT)0,                ofi_dom_corre);
        while(dbnextrow(sqltemp) != NO_MORE_ROWS)
        {
            roman_PC(ofi_nombre);
            roman_PC(ofi_dom_regio);
            roman_PC(ofi_dom_corre);
        }
        return DBCOUNT(sqltemp);
    }
}

void almacan_compania(void)
{
    dbfcmd(sqltemp, " select alm_clave, alm_nombre, alm_nom_respo, ";
    dbfcmd(sqltemp, " cia_nombre, unidad.uni_clave ";
    dbfcmd(sqltemp, " from almacan, unidad, compania ";
    dbfcmd(sqltemp, " where alm_clave = upper('a') ,alm_clave ";
    dbfcmd(sqltemp, " and unidad.uni_clave = almacan.uni_clave ";
    dbfcmd(sqltemp, " and compania.cia_clave = unidad.cia_clave ";
    dbsqllexac(sqltemp);
    if (dbresults(sqltemp) == SUCCEED)
    {
        dbbind(sqltemp, 1, STRINGBIND, (DBINT)0,                alm_clave);
        dbbind(sqltemp, 2, STRINGBIND, (DBINT)0,                alm_nombre);
        dbbind(sqltemp, 3, STRINGBIND, (DBINT)0,                alm_nom_respo);
        dbbind(sqltemp, 4, STRINGBIND, (DBINT)0,                cia_nombre);
        dbbind(sqltemp, 5, STRINGBIND, (DBINT)0,                uni_clave);
        while(dbnextrow(sqltemp) != NO_MORE_ROWS)
        {
            roman_PC(alm_nombre);
            roman_PC(alm_nom_respo);
            roman_PC(cia_nombre);
            strcpy(cia_nombre2, cia_nombre);
        }
        if (DBCOUNT(sqltemp) == 0)
        {
            printf("No existe el Almacen o la estructura Compania-Unidad-Almacen esta\n");
            exit(3);
        }
    }
}

void partida_requisicion(void)
{
    int descripcion(void);
    void print_notas(void);
    int w, x, y, z;
    dbfcmd(sqlppe, " select pr.prq_num_conse, ";
    if (strcmp(prv_nacional, "N") == 0)
    {
        dbfcmd(sqlppe, " use unimed, ";
    }
    else
    {
        dbfcmd(sqlppe, " use unimed_ing, ";
    }
    dbfcmd(sqlppe, " pr.req_num_conse, ";
    dbfcmd(sqlppe, " prq_can_reque, ";
    dbfcmd(sqlppe, " prq_can_coloc, ";
}

```

```

dbfcmd(sqlppe, "      prq_num_descr,      ");
dbfcmd(sqlppe, "      lin_clave,        ");
dbfcmd(sqlppe, "      lin_sub_linea,    ");
dbfcmd(sqlppe, "      mco_clave,        ");
dbfcmd(sqlppe, "      mun_clave,        ");
dbfcmd(sqlppe, "      pr.des_clave      ");
dbfcmd(sqlppe, "      from partida requisicion pr,");
dbfcmd(sqlppe, "      solicitud partida requisicion sp,");
dbfcmd(sqlppe, "      unidad medida um   ");
dbfcmd(sqlppe, "      where pr.alm_clave = sp.alm_clave ");
dbfcmd(sqlppe, "      and pr.req_num_conse = sp.req_num_conse ");
dbfcmd(sqlppe, "      and pr.prq_num_conse = sp.prq_num_conse ");
dbfcmd(sqlppe, "      and um.ume_clave = pr.ume_clave ");
dbfcmd(sqlppe, "      and sco_num_conse = 'd ',sco_num_conse );");
dbfcmd(sqlppe, "      order by pr.Feq_num_conse ");");
dbsqlxex(sqlppe);
if (dbresults(sqlppe) == SUCCEED)
{
  dbbind(sqlppe, 1, SMALLBIND, (DBINT)0, (BYTE *) &prq_num_conse);
  dbbind(sqlppe, 2, STRINGBIND, (DBINT)0, ume_unimed);
  dbbind(sqlppe, 3, INTBIND, (DBINT)0, (BYTE *) &req_num_conse);
  dbbind(sqlppe, 4, REALBIND, (DBINT)0, (BYTE *) &prq_can_reque);
  dbbind(sqlppe, 5, REALBIND, (DBINT)0, (BYTE *) &prq_can_coicoc);
  dbbind(sqlppe, 6, TINYBIND, (DBINT)0, (BYTE *) &prq_num_descr);
  dbbind(sqlppe, 7, TINYBIND, (DBINT)0, (BYTE *) &lin_clave);
  dbbind(sqlppe, 8, STRINGBIND, (DBINT)0, lin_sub_linea);
  dbbind(sqlppe, 9, STRINGBIND, (DBINT)0, mco_clave);
  dbbind(sqlppe, 10, STRINGBIND, (DBINT)0, mun_clave);
  dbbind(sqlppe, 11, INTBIND, (DBINT)0, (BYTE *) &des_clave);
  partida = 0;
  while (dbnextrow(sqlppe) != NO_MORE_ROWS)
  {
    partida += 1;
    if (prq_num_descr == 0)
    {
      strcpy(sppe_num_conse, " ");
    }
    else
    {
      sprintf(sppe_num_conse, "%2d", partida);
    }
    sprintf(sppe_num_conse, "%2d", partida);
    sprintf(sreq_num_conse, "%d", req_num_conse);
    sprintf(sprq_num_conse, "%d", prq_num_conse);
    check_page();
    fprintf(file_output, "%s%2.2-s%7.7s%5.5-s %4.4-s %5.5-s %2.2-s %12.12
    ,LV,sppe_num_conse,LV,format money(prq_can_reque, NULL, 7, -4),LV,ume_unime
    dbfcmd(sqltemp, " select des_nombre ");
    dbfcmd(sqltemp, " from descr temporal ");
    dbfcmd(sqltemp, " where des_clave = 'd ',des_clave ");
    dbfcmd(sqltemp, "order by des_num_conse ");
    dbsqlxex(sqltemp);
    if (dbresults(sqltemp) == SUCCEED)
    {
      dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, des_nombre);
      des_nombre2 = malloc(256 * sizeof(char));
      x = 1; w = 256;
      while (dbnextrow(sqltemp) != NO_MORE_ROWS)
      {
        roman_PC(des_nombre);
        if (x != 1)
          strcat(des_nombre2, des_nombre);
        else
          strcpy(des_nombre2, des_nombre);
        des_nombre2 = realloc(des_nombre2, ((w+255) * sizeof(char)));
        w += 255; x = 0;
      }
      if (DBCOUNT(sqltemp) == 0)
      {
        des_nombre2 = malloc(35 * sizeof(char));
        strcpy(des_nombre2, "**** PARTIDA SIN DESCRIPCION ****");
      }
      z = descripcion();
      for (y=0; y<z; y++)
      {
        check_page();
      }
    }
  }
}

```

```

        fprintf(file_output, "%s %s          %s36.36-s1s %s          %s
    }
    free(des_nombre2);
}
if (prq_num_descr == 0)
{
    dbfcmd(sqlppr, " select emb_num conse,                ");
    dbfcmd(sqlppr, " emb_cantidad,                        ");
    dbfcmd(sqlppr, " emb_fec_entre = convert(char(8), emb_fec_entre
    dbfcmd(sqlppr, " from partida_requisicion_programa ");
    dbfcmd(sqlppr, " where alm_clave = '%s', alm_clave ");
    dbfcmd(sqlppr, " and req_num_conse = %d ", req_num_conse );
    dbfcmd(sqlppr, " and prq_num_conse = %d ", prq_num_conse );
    dbfcmd(sqlppr, " order by emb_num_conse ");
    dbsqlxec(sqlppr);
    if (dbresults(sqlppr) == SUCCEED)
    {
        dbbind(sqlppr, 1, SMALLBIND, (DBINT)0, (BYTE *) &emb_num_conse);
        dbbind(sqlppr, 2, REALBIND, (DBINT)0, (BYTE *) &emb_cantidad);
        dbbind(sqlppr, 3, STRINGBIND, (DBINT)0, emb_fec_entre);
        while (dbnextrow(sqlppr) != NO_MORE_ROWS)
        {
            sprintf(appe_num_conse, "%2d", emb_num_conse);
            check_page();
            blank_detail_1();
            check_page();
            fprintf(file_output, "%s2.2-s1s%7.7s1s%5.5-s
        }
    }
}
check_page();
blank_detail_1();
}
dbfcmd(sqltemp, " select nta_nombre,                ");
dbfcmd(sqltemp, " nta_clave                        ");
dbfcmd(sqltemp, " from notas_requisicion ");
dbfcmd(sqltemp, " where alm_clave = '%s', alm_clave ");
dbfcmd(sqltemp, " and req_num_conse = %d ", req_num_conse );
dbfcmd(sqltemp, " order by nta_clave, nta_num_conse ");
dbsqlxec(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
    dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, des_nombre);
    dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, nta_clave);
    strcpy(break_nta_clave, "XXXXXXXXXXXXXXXX");
    while (dbnextrow(sqltemp) != NO_MORE_ROWS)
    {
        roman_PC(des_nombre);
        if (strcmp(break_nta_clave, nta_clave) != 0)
        {
            if (strcmp(break_nta_clave, "XXXXXXXXXXXXXXXX") != 0)
                print_notes();
            else
                check_page2();
            strcpy(break_nta_clave, nta_clave);
            des_nombre2 = malloc(256 * sizeof(char));
            x = 1; w = 256;
        }
        if (x != 1)
            strcat(des_nombre2, des_nombre);
        else
            strcpy(des_nombre2, des_nombre);
        des_nombre2 = realloc(des_nombre2, ((w+255) * sizeof(char)));
        w += 255; x = 0;
    }
    if (DBCOUNT(sqltemp) != 0)
        print_notes();
}
}
}

void print_notes(void)
{
    int y, z;
    par_not = 1;
    check_page();
}

```

```

blank detail 2();
z = descripcion2();
for(y=0;y<=z;y++)
{
    check_page();
    fprintf(file_output,"%s %72.72-s %s\n",LV,des_nota[y],LV);
}
free(des_nombre2);
}

```

```
void requisicion(void)
```

```

{
void nom_com(char *cve_com, char *nom_compr, char *clave_ofi);
int x;
dbfcmd(sqlreq, "select req_fec_reque = convert(char(8), req_fec_reque, 3), ");
dbfcmd(sqlreq, " com_clave, ");
dbfcmd(sqlreq, " req_des_gener, ");
dbfcmd(sqlreq, " req_alm_clave, ");
dbfcmd(sqlreq, " req.req_num_conse ");
dbfcmd(sqlreq, " from requisicion req, solicitud_partida_requisicion spr ");
dbfcmd(sqlreq, " where req.alm_clave = spr.alm_clave ");
dbfcmd(sqlreq, " and req.req_num_conse = spr.req_num_conse ");
dbfcmd(sqlreq, " and spr.sco_num_conse = %d ",sco_num_conse );
dbqlxexc(sqlreq);
if (dbresults(sqlreq) == SUCCEED)
{
    dbbind(sqlreq, 1, STRINGBIND, (DBINT)0, req_fec_reque);
    dbbind(sqlreq, 2, STRINGBIND, (DBINT)0, com_clave);
    dbbind(sqlreq, 3, STRINGBIND, (DBINT)0, req_des_gener);
    dbbind(sqlreq, 4, STRINGBIND, (DBINT)0, Falm_clave);
    dbbind(sqlreq, 5, INTBIND, (DBINT)0, (BYTE *) &rreq_num_conse);
    while(dbnextrow(sqlreq) != NO_MORE_ROWS)
        roman_PC(req_des_gener);
    if (DBCCOUNT(sqlreq) != 0)
    {
        for(x=0;req_des_gener[x] != NULL;x++)
            if (req_des_gener[x] == '\n')
                req_des_gener[x] = ' ';
            if (x > 81)
                req_des_gener[81] = NULL;
            nom_com(com_clave, com_nombre, com_ofi_clave);
    }
}
}
}

```

```
void nom_com(char *cve_com, char *nom_compr, char *clave_ofi)
```

```

{
nom_comp[0] = NULL;
dbfcmd(sqltemp, " select com_nombre ");
dbfcmd(sqltemp, " from comprador ");
dbfcmd(sqltemp, " where com_clave = '%s' ",cve_com );
dbqlxexc(sqltemp);
if (dbresults(sqltemp) == SUCCEED)
{
    dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, com_nombre);
    while(dbnextrow(sqltemp) != NO_MORE_ROWS)
        roman_PC(com_nombre);
}
}
}

```

```
void proveedor(void)
```

```

{
void domicilio(DBCHAR *rfc_pro, DBTINYINT num_dom);
prv_rzo_socia[0] = NULL; prv_rzo_socia_r[0] = NULL;
pdo_calle[0] = NULL; pdo_colonia[0] = NULL;
pdo_dlg_munic[0] = NULL; pdo_estado[0] = NULL;
pdo_pais[0] = NULL; pdo_cod_posta[0] = NULL;
dbfcmd(sqlpro, " select prv_rzo_socia, ");
dbfcmd(sqlpro, " prv_nacionall ");
dbfcmd(sqlpro, " from proveedor ");
dbfcmd(sqlpro, " where prv_rfc = '%s' ",prv_rfc );
dbqlxexc(sqlpro);
if (dbresults(sqlpro) == SUCCEED)
{
    dbbind(sqlpro, 1, STRINGBIND, (DBINT)0, prv_rzo_socia);
}
}
}

```

```

dbbind(sqlpro, 2, STRINGBIND, (DBINT)0, prv_nacional);
while(dbnextrrow(sqlpro) != NO_MORE_ROWS)
  roman_PC(prv_rzo_socia);
if (DBCOUNT(sqlpro) != 0)
  if ((strcmp(prv_rfc, prv_rfc_repre) != 0)
    {
      dbfcmd(sqlpro, " select prv_rzo_socia,                ");
      dbfcmd(sqlpro, "          prv_nacional                ");
      dbfcmd(sqlpro, "    from proveedor                ");
      dbfcmd(sqlpro, "    where prv_rfc = '%s' ",prv_rfc_repre );
      dbqlexec(sqlpro);
      if (dbresults(sqlpro) == SUCCEED)
        {
          dbbind(sqlpro, 1, STRINGBIND, (DBINT)0, prv_rzo_socia_r);
          dbbind(sqlpro, 2, STRINGBIND, (DBINT)0, prv_nacional);
          while(dbnextrrow(sqlpro) != NO_MORE_ROWS)
            roman_PC(prv_rzo_socia_r);
          if (DBCOUNT(sqlpro) != 0)
            domicilio(prv_rfc_repre, pdo_num_conse);
        }
    }
  else
    domicilio(prv_rfc, pdo_num_conse);
}
}

```

```

void domicilio(DBCHAR *rfc_pro, DBTINYINT dom_num)
{
  dbfcmd(sqltemp, " select pdo calle,      pdo colonia,      pdo_dlg_munic,  ");
  dbfcmd(sqltemp, "    pdo_cod_posEa = ltrim(convert(char(14),pdo_cod_posta)), ");
  dbfcmd(sqltemp, "    pdo_estado,      pdo_pais                ");
  dbfcmd(sqltemp, "    from domicilio proveedor                ");
  dbfcmd(sqltemp, "    where prv_rfc = '%s' ",rfc_pro );
  dbfcmd(sqltemp, "    and pdo_num_conse = %d ",dom_num );
  dbqlexec(sqltemp);
  if (dbresults(sqltemp) == SUCCEED)
    {
      dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, pdo_calle);
      dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, pdo_colonia);
      dbbind(sqltemp, 3, STRINGBIND, (DBINT)0, pdo_dlg_munic);
      dbbind(sqltemp, 4, STRINGBIND, (DBINT)0, pdo_cod_posta);
      dbbind(sqltemp, 5, STRINGBIND, (DBINT)0, pdo_estado);
      dbbind(sqltemp, 6, STRINGBIND, (DBINT)0, pdo_pais);
      while(dbnextrrow(sqltemp) != NO_MORE_ROWS)
        {
          roman_PC(pdo_calle);
          roman_PC(pdo_colonia);
          roman_PC(pdo_dlg_munic);
        }
      if (DBCOUNT(sqltemp) != 0)
        {
          if ((strlen(pdo_cod_posta) == 1) && (pdo_cod_posta[0] == '0'))
            pdo_cod_posta[0] = NULL;
          strcpy(dom_nom_pais, " ");
          if ((strlen(pdo_pais) != 0)
            tabla_detalle(16, pdo_pais, dom_nom_pais);
            if ((strlen(pdo_estado) != 0)
              tabla_detalle(19, pdo_estado, dom_nom_estado);
            dbfcmd(sqltemp, " select atn_atencion                ");
            dbfcmd(sqltemp, "    from atencion proveedor                ");
            dbfcmd(sqltemp, "    where prv_rfc = '%s' ",rfc_pro );
            dbfcmd(sqltemp, "    and atn_num_conse = 1                ");
            dbqlexec(sqltemp);
            if (dbresults(sqltemp) == SUCCEED)
              {
                dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, pdo_atencion);
                dbnextrrow(sqltemp);
                roman_PC(pdo_atencion);
              }
            dbfcmd(sqltemp, " select prv_telefono, prv_fax ");
            dbfcmd(sqltemp, "    from telefono proveedor ");
            dbfcmd(sqltemp, "    where prv_rfc = '%s' ",rfc_pro );
            dbfcmd(sqltemp, "    and prv_num_conse = 1 ");
            dbfcmd(sqltemp, "    and prv_cve_prove = 'P' ");
            dbqlexec(sqltemp);

```

```

        if (dbresults(sqltemp) == SUCCEED)
        {
            dbbind(sqltemp, 1, STRINGBIND, (DBINT)0, pdo_telefono);
            dbbind(sqltemp, 2, STRINGBIND, (DBINT)0, pdo_fax);
            dbnextrow(sqltemp);
        }
    }
}

int descripcion(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 1000; r++)
        des_parte[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (des_parte[r][c] = des_nombre2[i]) != NULL && r < 1000; i++)
    {
        if (des_nombre2[i] == '\n')
        {
            des_parte[r++][c] = NULL;
            c = -1;
            continue;
        }
        if (c > 34)
        {
            des_parte[r++][++c] = NULL;
            c = -1;
        }
    }
    if (strlen(des_parte[r]) == 0)
        r--;
    return r;
}

int descripcion2(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 500; r++)
        des_notas[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (des_notas[r][c] = des_nombre2[i]) != NULL && r < 500; i++)
    {
        if (des_nombre2[i] == '\n')
        {
            des_notas[r++][c] = NULL;
            c = -1;
            continue;
        }
        if (c > 72)
        {
            des_notas[r++][++c] = NULL;
            c = -1;
        }
    }
    if (strlen(des_notas[r]) == 0)
        r--;
    return r;
}

void direction_office(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)
        dom_ofna_gr1[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (dom_ofna_gr1[r][c] = ofi_dom_regio[i]) != NULL && r < 10; i++)
    {
        if (ofi_dom_regio[i] == '\n')
        {
            dom_ofna_gr1[r++][c] = NULL;
            c = -1;
            continue;
        }
        if (c > 34)
        {
            dom_ofna_gr1[r++][++c] = NULL;
        }
    }
}

```

```

    c = -1;
}

}
for(r = 0; r < 10; r++)
    if ((strlen(dom_ofna_grl[r])) > 32)
        dom_ofna_grl[r][33] = NULL;
}

void direction_corres(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)
        dom_corres[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (dom_corres[r][c] = ofi_dom_corre[i]) != NULL && r < 10; i)
        {
            if (ofi_dom_corre[i] == '\n')
                {
                    dom_corres[r++][c] = NULL;
                    c = -1;
                    continue;
                }
            if (c > 34)
                {
                    dom_corres[r++][++c] = NULL;
                    c = -1;
                }
        }
    for(r = 0; r < 10; r++)
        if ((strlen(dom_corres[r])) > 32)
            dom_corres[r][33] = NULL;
}

void alm_direction_fact(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)
        dom_factor[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (dom_factor[r][c] = ald_nombre_fac[i]) != NULL && r < 10; i)
        {
            if (ald_nombre_fac[i] == '\n')
                {
                    dom_factor[r++][c] = NULL;
                    c = -1;
                    continue;
                }
            if (c > 40)
                {
                    dom_factor[r++][++c] = NULL;
                    c = -1;
                }
        }
}

void alm_direction_cons(void)
{
    int i, r, c;
    for(r = 0, c = 0; r < 10; r++)
        dom_consigt[r][c] = NULL;
    for(i = 0, r = 0, c = 0; (dom_consigt[r][c] = ald_nombre_con[i]) != NULL && r < 10; i)
        {
            if (ald_nombre_con[i] == '\n')
                {
                    dom_consigt[r++][c] = NULL;
                    c = -1;
                    continue;
                }
            if (c > 32)
                {
                    dom_consigt[r++][++c] = NULL;
                    c = -1;
                }
        }
}

void def_char_ctl(void)

```

```

{
  int i;
  if (export_to == 0)
  {
    Q[2] = NULL; W[2] = NULL; A[2] = NULL; S[2] = NULL;
    LV[3] = NULL; LH[1] = NULL;
    LVGI[2] = NULL; LVGD[2] = NULL; LHGS[1] = NULL; LHGI[1] = NULL; CRS[1] = NULL;
    Q[0] = 14; A[0] = 14;
    LV[0] = 14;
    LVGI[0] = 14;
    LV[2] = 15; W[1] = 15; S[1] = 15;
    Q[1] = 'R'; W[0] = 'T'; A[1] = 'F'; S[0] = 'G';
    LV[1] = '.'; LH[0] = '.';
    LVGI[1] = 'S'; LVGD[0] = '6'; LHGS[0] = '7'; LHGI[0] = '8'; CRS[0] = '/';
  }
  else
  {
    Q[1] = NULL; W[1] = NULL; A[1] = NULL; S[1] = NULL;
    LV[1] = NULL; LH[1] = NULL;
    LVGI[1] = NULL; LVGD[1] = NULL; LHGS[1] = NULL; LHGI[1] = NULL; CRS[1] = NULL;
    Q[0] = 218; W[0] = 191; A[0] = 192; S[0] = 217;
    LV[0] = 179; LH[0] = 196;
    LVGI[0] = 195; LVGD[0] = 180; LHGS[0] = 194; LHGI[0] = 193; CRS[0] = 197;
  }
  for (i=0; i<80; i++)
    filler[i] = LH[0];
  filler[80] = NULL;
}

```

APENDICE D.

DEFINICION DE FORMAS DEL SISTEMA

NOTA: En este apéndice sólo se incluye una parte de la información referente al mismo, debido a lo extenso del sistema.

Text: Ped. Cerr/Cancel por Div/Cia
Value: 136
Key: d

Text: Remisión de Pedidos y Modif.
Value: 128
Key: r

Text: Req. y Pedidos de Proyectos
Value: 975
Key: q

Text: Control Costos Proyecto
Value: 976
Key: c

Text: Pedidos Monto Proyecto
Value: 977
Key: m

Text: Reportes de Requisiciones
Value: 954
Key: p

Text: Req/Colocadas por Area/comp
Value: 955
Key: r

Text: Req/Colocadas por Div/Cia
Value: 956
Key: c

Text: Req/Recibidas por Lin/Material
Value: 958
Key: l

Text: Req/Pend/Coloc por Area/Comp
Value: 959
Key: p

Text: Req/Pend/Coloc por Div/Cia
Value: 960
Key: d

Text: Tabla Comparativa Requisición
Value: 961
Key: t

Text: Trafico
Value: 63
Key: t

Text: Movimientos de Material
Value: 970
Key: m

Text: Consulta de Pedidos
Value: 971
Key: c
APTSQL Varname: C

Text: Reportes
Value: 972
Key: r

Text: Situacion de Traficos
Value: 973
Key: s

Text: Infraestructura
Value: 65
Key: i

Text: Proveedoras
Value: 110
Key: p

Text: Requisitores
Value: 66
Key: r

Text: Division
Value: 96
Key: d

Text: Compania
Value: 97
Key: c

Text: Unidad
Value: 98
Key: u

Text: Centro
Value: 99
Key: e

Text: Compradores
Value: 70
Key: c

Text: Oficinas
Value: 100
Key: o

Text: Compradoras
Value: 101
Key: c

Text: C. Password
Value: 978
Key: p

Text: Materiales
Value: 72
Key: m

Text: Cod.Almacen
Value: 103
Key: a

Text: Cod.Comercial
Value: 104
Key: c

Text: U/Medida
Value: 105
Key: m

Text: Linea/Sublinea
Value: 106
Key: l

Text: A. Aduanales
Value: 974
Key: a

Text: Otros
Value: 93
Key: o

Text: Tipo de Cambio
Value: 108
Key: c

Text: Catlgo. de Monedas
Value: 118
Key: m

Text: Catlgo. de Notas
Value: 107
Key: n

Text: Domicilio
Value: 113
Key: d

Text: Catlgo. de Tablas
Value: 119
Key: t

Text: /
Value: 942
Key: /

Text: Close
Value: 943
Key: c

Text: Customize
Value: 927
Key: u

Text: Database
Value: 914
Key: d

Text: Properties
Value: 917
Key: p

Text: Character
Value: 267
Key: c

Text: Integer
Value: 278
Key: i

Text: Float
Value: 282
Key: f

Text: Money
Value: 277
Key: m

Text: Datetime
Value: 268
Key: d

Text: Binary
Value: 266
Key: b

Text: Bit
Value: 273
Key: t

Text: Help
Value: 945
Key: h

No groups.

GROUPS

ID: 1
Prompt: SELECCIONE UNA OPCION <F1, F2...> FIELD: field_1
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 2
Prompt: FIELD: fecha_sis
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 27
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 3
Prompt: wrk formname FIELD: wrk_formname
In Group:
Desc: Campo para asignar nombre de forma a cargar
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 9
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 4
Prompt: FIELD: wrk_dummy
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: yes
Stop On: yes
Highlight: no

ID: 5
Prompt: com_cve_usuar FIELD: com_cve_usuar
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 7
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 8
Prompt: Usuario
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: uname

ID: 9
Prompt: Password
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: upassword

ID: 10
Prompt: Server
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: server

ID: 11
Prompt: Reporte
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: rep_clave

ID: 12
Prompt: Impresora
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: imp_clave

ID: 13
Prompt: Coplas
FIELD: ncopias
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 14
Prompt: Tipo Imp.
FIELD: tipo
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 15
Prompt: Ubicacion
FIELD: ubica
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 51
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 16
Prompt: Print status
FIELD: print_status
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 18
Prompt: Ruta
FIELD: ruta
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 50
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 19
Prompt: opcion
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 50
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: opcion

ID: 20
Prompt: parametros
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 100
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: parametros

ID	Name	Routine Name	Class	Parameter
21	/	exitform	Supp	
62	Compras			
80	Act./Cons.			
83	Requisicion	abinmen02 1	FPL	
84	Sol:cotizacion	exit routine abinmen02 1	Ext FPL	
85	Cotizacion	exit routine abinmen02 1	Ext FPL	
86	Pedido:Colocar	exit routine abinmen02 1	Ext FPL	
87	Pedido:Modificar	exit routine abinmen02 1	Ext FPL	
95	Movimiento de Material	exit routine abinmen02 1	Ext FPL	
81	Consulta	exit_routine	Ext	
89	Busqueda de Req.	abinmen04 1	FPL	
111	Selec. x Fecha	exit routine abstFaq01 0	Ext Form	
112	Selec. x Unida	abinmen00 1	FPL	
966	Req. y Partidas	abinmen04 1 exit routine abinmen00 1	Ext FPL	
967	Req. y Pedidos	abreqco21 1 abinmen00 1	Form FPL	
90	Busqueda de Ped.	abreqco22 1 abinmen00 1 abstreq03 0	Form FPL Form	
91	Cotizaciones	abinmen00 1	FPL	
92	Movimientos			
114	Estadístico Requisición	abrqres01 1 abinmen00 1	Form FPL	
116	Estadístico Pedido	abrqres02 1 abinmen00 1	Form FPL	
968	Historico Requisiciones	abrqres11 1 abinmen01 1	Form FPL	
969	Historico Pedidos	abrqres12 1 abinmen01 1	Form FPL	
963	Catálogos			
964	Catálogo de domicilios	abindom01 1	FPL	
965	Relación Almacén Domicil	abinpped01 1	FPL	
82	Reportes de Pedidos			
124	Lista de Precios	ablispre01 1	FPL	
127	Pedidos Autorizados Diar	abpedaut01 1	FPL	
950	Pedidos a Expeditar a la	abpeped11 1	FPL	
126	Ped. Cerra/Cancel por Ar	abinmen101 1	FPL	
136	Ped. Cerr/Cancel por Div	abpecoldiv01 1	FPL	
128	Remisión de Pedidos y Mo	abrcmonv01 1	FPL	
975	Req. y Pedidos de Proyec	abinmen09 1	FPL	
976	Control Costos Proyecto	abinmen11 1	FPL	
977	Pedidos Monto Proyecto	abinmen13 1	FPL	

954	Reportes de Requisiciones		
955	Req/Colocadas por Area/c	abrqqcol2 1	FPL
956	Req/Colocadas por Div/Cl	abrqqcol3 1	FPL
958	Req/Recibidas por Lin/Ha	abrqpen01 1	FPL
959	Req/Pend/Coloc por Area/c	abrqpen01 1	FPL
960	Req/Pend/Coloc por Div/C	abrqpen02 1	FPL
961	Tabla Comparativa Requis	abrtabcom3 1	FPL
63	Trafico		
970	Movimientos de Material	abinmen06 1	FPL
		exit_routine	Ext
971	Consulta de Pedidos	abinmen06 1	FPL
		exit_routine	Ext
972	Reportes		
973	Situacion de Traficos	abinmen07 1	FPL
65	Infraestructura		
110	Proveedores	abinmen05 1	FPL
66	Requisitores	abindiv01 1	Form
96	Division	abinmen05 1	FPL
97	Compania	abinmen05 1	FPL
98	Unidad	abinmen05 1	FPL
99	Centro	abinmen05 1	FPL
70	Compradores	abinofr01 1	Form
100	Oficinas	abinmen05 1	FPL
101	Compradores	abinmen05 1	FPL
978	C. Password	abinmen05 1	FPL
72	Materiales	abinmat01 1	Form
103	Cod.Almacen	abinmen05 1	FPL
104	Cod.Comercial	abinmen05 1	FPL
105	U/Medida	abinmen05 1	FPL
106	Linea/Sublinea	abinmen05 1	FPL
974	A. Aduanales	abinmen05 1	FPL
93	Otros	abinnta01 1	Form
108	Tipo de Cambio	abinmen05 1	FPL
118	Catigo. de Monedas	abinmen05 1	FPL
107	Catigo. de Notas	abinmen05 1	FPL
113	Domicilio	abinmen05 1	FPL
119	Catigo. de Tablas	abinmen05 1	FPL
942 /			
943	Close	exitform	Supp
927	Customize		
914	Database	custdb	Supp
917	Properties		
267	Character	formatchar	Supp
278	Integer	formatnum	Supp
282	Float	formatfloat	Supp
277	Money	formatmoney	Supp
268	Datetime	formatdate	Supp
266	Binary	formatbin	Supp
273	Bit	formatbit	Supp
945	Help	help	Supp

ID	Object	When	Routine Name	Class	Parameter
	Form	enter	abinmen01 1	FPL	
		exit	abinmen91 1	FPL	
		exit	abinmen00 1	FPL	
2	fecha_sis				
3	wrk_fOrname				
4	wrk_dummy				
5	com_cve_usuar				
8	uname				
9	upassword				
10	server				
11	rep_clave	enter	abinmen08 1	FPL	
12	imp_clave				
13	ncopias				
14	tipo				
15	ubica				
16	print_status				
18	ruta				
19	opcion				
20	parametros				

APT Report completed.

10/01/93 17:56
Page 1

abreqin01 1
FORM

Form Name: abreqin01
Form Version: 1
Title: REQUISICIONES
Description: FORMA PARA CAPTURA DE REQUISICIONES (DATOS GRALES)
Initial Field: alm_clave
Readonly: no
Show Menu: yes
Character Move: no
Inherit Function Keys: no
X: 0
Y: 0
Width: 77
Height: 17

Form Parameters:	Parameter	Type	Direction
	alm_clave	field	input
	req_num_conse	field	input

MENU
Title: 'REQUISICIONES'

MENU

Text: /
Value: 942
Key: /

Text: Actualiza
Value: 61
Key: a

Text: Alta
Value: 75
Key: a

Text: Cambio
Value: 77
Key: c

Text: Elim./Cancela
Value: 76
Key: e

Text: Cancela
Value: 78
Key: c

Text: Reactiva
Value: 79
Key: r

Text: Elimina
Value: 80
Key: e

Text: Transferencia
Value: 944
Key: t

Text: Consulta
Value: 64
Key: c

Text: requisiciones
Value: 86
Key: r

Text: cotizaciones
Value: 87
Key: c

Text: pedidos
Value: 88
Key: p

Text: movimientos
Value: 97
Key: m

Text: Ptd
Value: 65
Key: p

Text: Notas
Value: 82
Key: n

Text: Menu
Value: 89
Key: m

Text: Actualizacion
Value: 92
Key: a

Text: sol:Cotizacion
Value: 94
Key: s

Text: cotizaciones
Value: 105
Key: z

Text: pedidos:Coloc.
Value: 106
Key: c

Text: pedidos:Modif.
Value: 110
Key: m

Text: movimto:import.
Value: 107
Key: i

Text: movimto:Nacional
Value: 117
Key: n

Text: Consulta
Value: 99
Key: c

Text: requisiciones
Value: 100
Key: r

Text: pedidos
Value: 101
Key: p

Text: cotizaciones
Value: 103
Key: c

Text: movimientos
Value: 104
Key: m

Text: Reportes
Value: 118
Key: r

Text: Salida
Value: 128
Key: s

Text: Limpia
Value: 124
Key: l

Text: E/Col
Value: 945
Key: e

No groups.

GROUPS

ID: 1
Prompt: FIELD: alm_clave

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Database: abaste.dbo.requisicion.alm_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 2
Prompt: FIELD: req_num_conse

In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Database: abaste.dbo.requisicion.req_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 57
Prompt: Cta C. FIELD: alm_cta_cuent

In Group:
Desc: Cuenta de cargo. Si toma de Almacen si existe.
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 4
Database: abaste.dbo.requisicion.req_cta_cuent
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 58
Prompt: FIELD: alm_cta_subct

In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 4
Database: abaste.dbo.requisicion.req_cta_subct
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 59
Prompt: FIELD: alm_cta_ssubc

In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 4
Database: abaste.dbo.requisicion.req_cta_ssubc
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 60
Prompt: FIELD: alm_cta_conce

In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 4
Database: abaste.dbo.requisicion.req_cta_conce
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 33
Prompt: Aut.D FIELD: cia_clave

In Group:
Desc: Autorizacion Desembolso (Existe en tabla autorizaciones)
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 3
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 61
Prompt: FIELD: req_aut_plant

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.requisicion.req_aut_plant
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 62 FIELD: req_aut_folio
Prompt:

In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 4
Database: abaste.dbo.requisicion.req_aut_folio
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes

ID: 65 FIELD: req_aut_perio
Prompt:

In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Database: abaste.dbo.requisicion.req_aut_perio
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 17 FIELD: req_imp_estim
Prompt:

Monto
In Group:
Desc: Importe estimado de la requisicion (informativo)
Appearance Definition:
Datatype: money
Field Type: entry
Max Len: 15
Database: abaste.dbo.requisicion.req_imp_estim
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 18 FIELD: mon_clave
Prompt:

In Group:
Desc: Moneda del importe estimado (informativo) verificar moneda
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.requisicion.req_imp_moned
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 40
Prompt: Depto
FIELD: req_dep_nombr
In Group:
Desc: Nombre de Departamento (informativo)
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 25
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 14
Prompt: Solic
FIELD: req_nom_solic
In Group:
Desc: Nombre de Solicitante (informativo)
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 25
Database: abaste.dbo.requisicion.req_nom_solic
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 55
Prompt: O.T.
FIELD: req_ord_traba
In Group:
Desc: Numero de Orden de Trabajo (Informativo)
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 6
Database: abaste.dbo.requisicion.req_ord_traba
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 56
Prompt: Equipo
FIELD: req_num equip
In Group:
Desc: Identificacion de equipo que usara el matr. (informativo)
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 12
Database: abaste.dbo.requisicion.req_num equip
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 39

FIELD: req_folio

Prompt: Folio
In Group:
Desc: Folio (solo por tipo de almacen) informativo.
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 10
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 6

FIELD: com_clave

Prompt: Comprador
In Group:
Desc: Comprador asignado por sistema segun Linea de material(Con)
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 4

FIELD: wrk_fec_recep

Prompt: Recb
In Group:
Desc: Fecha de recepcion de requisicion, default = fecha captura
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 6
Database: abaste.dbo.requisicion.req_fec_recep
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 3

FIELD: wrk_fec_elabo

Prompt: Elab
In Group:
Desc: Fecha de elaboracion(Requisitiior) de documento
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 6
Database: abaste.dbo.requisicion.req_fec_recep
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 5
Prompt: Reqd
FIELD: wrk_fec_reque
In Group:
Desc: Fecha que se requiere matr. debe ser > hoy
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 6
Database: abaste.dbo.requisicion.req_fec_reque
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID:119
Prompt: Dias
FIELD: wrk_dia_coloc
In Group:
Desc: Dias pronostico de colocacion de pedido --> fec-asignacion c
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID:123
Prompt: E/Coloc
FIELD: wrk_fec_asign
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 6
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 15
Prompt: Tipo
FIELD: req_tip_servi
In Group:
Desc: Tipo de servicio de matr. Reparacion, Adquis. Servicio
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion.req_tip_servi
Use Default: yes
Run Rule: yes
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 16 FIELD: req_cod_prior

Prompt: Prio
In Group:
Desc: Prioridad de requisicion: (A)alta (B)medio (C) baja
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion.req_cod_prior
Use Default: yes
Run Rule: yes
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 46 FIELD: req_act_fijo

Prompt: A.Fijo
In Group:
Desc: Indicador de si/no activo fijo (informativo)
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion.req_act_fijo
Use Default: yes
Run Rule: yes
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 41 FIELD: lin_clave

Prompt: Linea
In Group:
Desc: Linea de material de la requisicion, esta asigna comprador
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 45 FIELD: lin_sub_linea

Prompt: Sublinea
In Group:
Desc: Sublinea de matr. Usado en Catalogo Corporativo
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 6
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 19
Prompt: Descripción Generica
FIELD: req_des_gener
In Group:
Desc: Descripción de la Línea del Material (Informativo)
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 255
Database: abasto.dbo.requisicion.req_des_gener
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 31
Prompt:
FIELD: alm_nombre
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 30
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 34
Prompt: uni clave
FIELD: uni_clave
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 5
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 35
Prompt: req fec asign
FIELD: req_fec_asign
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 68
Prompt: ofi_clave
In Gröup:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: ofi_clave

ID: 13
Prompt: req_num_emple
In Gröup:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Database: abaste.dbo.requisicion.req_num_emple
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: req_num_emple

ID: 69
Prompt: com_area
In Gröup:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: com_area

ID: 70
Prompt: alm_num_emple
In Gröup:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 6
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: alm_num_emple

ID: 71 FIELD: alm_nom_respo
Prompt: alm nom_respo
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 25
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 72 FIELD: wrk_indexi
Prompt: wrk indexi
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 73 FIELD: req_num_ptdas
Prompt: Partidas
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 4
Database: abaste.dbo.requisicion.req_num_ptdas
Use Default: no
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 74 FIELD: req_num_coloc
Prompt: Colocadas
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 4
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 82
Prompt: wk nivel
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: wk_nivel

ID: 83
Prompt: com nivel
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: com_nivel

ID: 86
Prompt: wrk_num_conse
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: wrk_num_conse

ID: 89
Prompt: rca 1:
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_1
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: rca_1

ID: 90
Prompt: rca 2: FIELD: rca_2
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_2
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 91
Prompt: rca 3: FIELD: rca_3
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_3
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 92
Prompt: rca 4: FIELD: rca_4
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_4
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 93
Prompt: rca 5: FIELD: rca_5
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_5
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 94
Prompt: rca 6:
In GrGup:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_6
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: rca_6

ID: 95
Prompt: rca 7:
In GrGup:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_7
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: rca_7

ID: 96
Prompt: rca 8:
In GrGup:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_8
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: rca_8

ID: 97
Prompt: rca 9:
In GrGup:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_9
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: rca_9

ID: 98

FIELD: rca_10

Prompt: rca_10:
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_10
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 99

FIELD: rca_11

Prompt: rca_11:
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_11
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:100

FIELD: rca_12

Prompt: rca_12:
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_12
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:101

FIELD: rca_13

Prompt: rca_13:
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Database: abaste.dbo.requisicion_campo.rca_13
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:102

Prompt: wk_comprador
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: wk_comprador

ID:103

Prompt: req_num_pedid
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: req_num_pedid

ID:104

Prompt: (R,<A>,S)
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wrk_def_tipo

ID:105

Prompt: (A,B,<C>)
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wrk_def_prior

ID:106
Prompt: (S,<N>)
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wrk_def_activ

ID:107
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 6
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wrk_def_recep

ID:108
Prompt: req_fec_elabo
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: req_fec_elabo

ID:109
Prompt: req_fec_recep
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: req_fec_recep

ID:110
Prompt: req_fec_reque
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: req_fec_reque

ID:112
Prompt: lin_nombre
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 60
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: lin_nombre

ID:114
Prompt: Centro
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: field_114

ID:115
Prompt: Requisicion
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: field_115

ID:116

FIELD: wrk_formname

Prompt: wrk formname
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 9
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:117

FIELD: wrk_fechas

Prompt: Fechas de:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID:118

FIELD: wrk_notas

Prompt: Notas
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID:120

FIELD: wrk_ptd_prog

Prompt: Programadas
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 3
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID:122 FIELD: wrk_dia_seman

Prompt: wrk_dia_seman
In Group:
Desc: Campo para extraer el dia de la semana y calc. dias habiles
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:111 FIELD: req_fec_statu

Prompt: req_fec_statu
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:124 FIELD: com_clave_sup

Prompt: com_clave_sup
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:125 FIELD: wk_clave_sup

Prompt: wk_clave_sup
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:127

Prompt: wk com
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: wk_com

ID:128

Prompt: wk sup
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: wk_sup

ID:129

Prompt: com_cve_usuar
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 7
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: com_cve_usuar

ID:130

Prompt: Usuario
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 25
Attributes:
Visible: yes
Stop On: no
Highlight: yes

FIELD: wrk_usu_clave

ID:131 FIELD: wok_cia_clave
Prompt: wok_cia_clave:
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:132 FIELD: field_132
Prompt: field_132:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: yes
Stop On: yes
Highlight: no

ID:133 FIELD: wok_limpia
Prompt: wok_limpia
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:134 FIELD: wok_diasc
Prompt: wok_diasc
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

MENU ID	Name	Routine Name	Class	MENU Parameter
942	/	abreqin18 1	FPL	
		exit_routine	Ext	
61	Actualiza			
75	Alta	abreqin11 1	FPL	
77	Cambio	abreqin16 1	FPL	
76	Elim./Cancela			
78	Cancela	abreqin17 1	FPL	
79	Reactiva	abreqin10 1	FPL	
80	Elimina	abreqin14 1	FPL	
944	Transferencia	abreqin31 1	FPL	
64	Consulta			
86	requisiciones	abreqin32 1	FPL	
87	cotizaciones			
88	pedidos	abreqin34 1	FPL	
97	movimientos			
65	Ptd	abreqin29 1	FPL	
82	Notas	abreqin90 1	FPL	
89	Menu			
92	Actualizacion			
94	sol:Cotizacion	abreqin24 1	FPL	
		exit_routine	Ext	
105	cotizaciones	abreqin24 1	FPL	
		exit_routine	Ext	
106	pedidos:Coloc.	abreqin24 1	FPL	
		exit_routine	Ext	
110	pedidos:Modif.	abreqin24 1	FPL	
		exit_routine	Ext	
107	movimto:import.	abreqin24 1	FPL	
		exit_routine	Ext	
117	movimto:Nacional	abreqin24 1	FPL	
		exit_routine	Ext	
99	Consulta			
100	requisiciones	abreqin19 1	FPL	
		exit_routine	Ext	
101	pedidos			
103	cotizaciones			
104	movimientos			

FIELDS

118	Reportes			
128	Salida	cancelform	Supp	
124	Limpia	abreqin27 1	FPL	
945	E/Col	abreqin36 1	FPL	

ID	Object	When	Routine Name	Class	Parameter
	Form				
Func f03		enter	abreqin01 1	FPL	
Func f04			abreqin29 1	FPL	
Func f06			abreqin50 1	FPL	
Func f07			abreqin27 1	FPL	
1	alm_clave		abreqin36 1	FPL	
2	req_num_conse	exit	abreqin28 1	FPL	
57	alm_cta_cuent	exit	abreqin03 1	FPL	
58	alm_cta_subct	exit	abreqin04 1	FPL	
59	alm_cta_saubc				
60	alm_cta_conce				
33	cia_clave	enter	abreqin30 1	FPL	
61	req_aut_plant	exit	abreqin05 1	FPL	
62	req_aut_folio				
65	req_aut_perio				
17	req_imp_estim	exit	abreqin20 1	FPL	
18	mon_clave	exit	abreqin21 1	FPL	
40	req_dep_nombr	exit	abreqim01 1	FPL	
14	req_nom_solic				
55	req_ord_traba				
56	req_num equip				
39	req_folio	exit	abreqin22 1	FPL	
6	com_clave	exit	abreqin12 1	FPL	
4	wrk_fec_recep				
3	wrk_fec_elabo				
5	wrk_fec_reque				
119	wrk_dia_coloc	exit	abreqin78 1	FPL	
123	wrk_fec_asign	exit	abreqim01 1	FPL	
15	req_tip_servi	exit	abreqim01 1	FPL	
16	req_cod_prior	exit	abreqim01 1	FPL	
46	req_act_fijo	exit	abreqim01 1	FPL	
41	lin_clave	exit	abreqin07 1	FPL	
45	lin_sub_linea	exit	abreqin08 1	FPL	
19	req_des_gener	exit	abreqin09 1	FPL	
31	alm_nombre	exit	abreqin91 1	FPL	
			abreqin45 1	FPL	
			abreqin02 1	FPL	
			abreqin98 1	FPL	
34	uni_clave				
35	req_fec_asign				
43	alm_cve_indic				
48	com_nombre				
49	req_cve_statu				
50	wrk_fec_statu				
68	ofi_clave				
13	req_num_emple				
69	com_area				
70	alm_num_emple				
71	alm_nom_respo				
72	wrk_index1				
73	req_num_ptdas				
74	req_num_coloc				

82 wk nivel
83 com_nivel
86 wrk_num_conse
89 rca_1
90 rca_2
91 rca_3
92 rca_4
93 rca_5
94 rca_6
95 rca_7
96 rca_8
97 rca_9
98 rca_10
99 rca_11
100 rca_12
101 rca_13
102 wk Comprador
103 req_num_pedid
107 wrk_def_recap
108 req_fec_elabo
109 req_fec_recep
110 req_fec_reque
112 lin_nombre
116 wrk_formname
118 wrk_notas
120 wrk_ptd_progr
122 wrk_dia_seman
111 req_fec_statu
124 com_clave_sup
125 wk_clave_sup
127 wk_com
128 wk_sup
129 com_cve_usuar
130 wrk_usu_clave
131 wok_cia_clave
132 field_132
133 wok_limpia
134 wok_diaac

APT Report completed.

10/01/93 18:02
Page 1

abctsol01 1
FORM

Form Name: abctsol01
Form Version: 1
Title: SOLICITUD DE COTIZACION
Description:
Initial Field: sco_num_conse
Readonly: no
Show Menu: yes
Character Move: yes
Inherit Function Keys: yes

X: 0
Y: 0
Width: 77
Height: 17

Form Parameters:	Parameter	Type	Direction
	alm_clave	field	both
	wk_Feq_ini	field	both
	wk_req_fin	field	both
	total_Feq	field	input

MENU
Title: 'SOLICITUD DE COTIZACION'

MENU

Text: /
Value: 21
Key: /

Text: Otra Solicitud
Value: 92
Key: o

Text: Salida
Value: 93
Key: s

Text: Act
Value: 67

Text: Cambio
Value: 68

Text: Baja
Value: 70

Text: Movtos.
Value: 71

Text: Seleccion Requisicion
Value: 72

Text: Partidas Selectivo
Value: 73

Text: Partidas Automatico

Value: 83
Text: Alta
Value: 84
Text: Actual Req.
Value: 85
Text: Todas Req.
Value: 86
Text: Baja
Value: 88
Text: Actual Req.
Value: 89
Text: Todas Req.
Value: 90
Text: Prov.
Value: 77
Text: Asigna
Value: 78
Text: Consulta
Value: 79
Text: Req. Ant.
Value: 80
Text: Req. Sig.
Value: 81
Text: Sol Asoc.
Value: 96

No groups.

GROUPS

ID: 10
Prompt: Solicitud No. FIELD: sco_num_conse
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Database: abaste.dbo.partida_requisicion.req_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 9
Prompt: Centro FIELD: alm_clave
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Database: abaste.dbo.partida_requisicion.alm_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 50
Prompt: Requisicion FIELD: req_num
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 26
Prompt: FIELD: alm_nombre
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 38
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 3
Prompt: Fecha de Solicitud FIELD: sco_fac_sol
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 27
Database: abaste.dbo.solicitud_cotizacion.sco_fac_sol
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no

ID: 5
Prompt: Fec. Lim. Cot. FIELD: sco_fec_lim_cotiz
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 11
Database: abaste.dbo.solicitud_cotizacion.sco_fec_lim_cotiz
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 6
Prompt: Notas FIELD: sco_comentarios
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 9000
Database: abaste.dbo.solicitud_cotizacion.sco_comentarios
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 51
Prompt: FIELD: wok_cont
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: yes
Stop On: yes
Highlight: no

ID: 25
Prompt: Descripcion FIELD: req_des_gener
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 255
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 27
Prompt: de FIELD: req_de
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 5
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 35
Prompt: Requisicion Numero
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 5
Attributes:
Visible: yes
Stop On: no
Highlight: yes

FIELD: req_no

ID: 36
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wk_num_sol

ID: 37
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wk_req_ini

ID: 38
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wk_req_fin

ID: 39
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: total_req

ID: 43
Prompt:

FIELD: des_clave

In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 44
Prompt:

FIELD: des_nombre

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 255
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 45
Prompt:

FIELD: counter

In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 46
Prompt:

FIELD: wk_par_sol

Partidas
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 28
Prompt:

FIELD: wk_req_act

Requisicion en proceso
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 48
Prompt: en sollicitud
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Attributes:
Visible: yes
Stop On: no
Highlight: yes
FIELD: wk_par_req

ID: 49
Prompt: wrk formname
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 9
Attributes:
Visible: yes
Stop On: no
Highlight: no
FIELD: wrk_formname

ID: 52
Prompt: com clave:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: com_clave

ID: 53
Prompt: com nivel
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: com_nivel

ID: 54
Prompt: wok_comp_usu
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: wok_comp_usu

ID: 55 FIELD: com_clave_sup
Prompt: com_clave_sup
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 56 FIELD: ind_comp_ok
Prompt: ind_comp_ok
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 59 FIELD: req_num_conse
Prompt: req_num_conse:
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Database: abaste.dbo.solicitud_partida_requisicion.req_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 60 FIELD: prq_num_conse
Prompt: prq_num_conse:
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Database: abaste.dbo.solicitud_partida_requisicion.prq_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 63 FIELD: alm_clave_com
Prompt: alm_clave_com
In GrGup:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Database: abaste.dbo.solicitud_partida_requisicion.alm_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 67 FIELD: fecha_hoy
Prompt: fecha_hoy
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 33
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 74 FIELD: prq_cod_estat
Prompt: prq_cod_estat
In GrGup:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.partida_requisicion.prq_cod_estat
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 75 FIELD: num_sol
Prompt: num_sol
In GrGup:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Attributes:
Visible: no
Stop On: no
Highlight: yes

ID	Name	Routine Name	Class	Parameter
21	/			
92	Otra Solicitud	abctsol60 1	FPL	
93	Salida	abctsol60 1	FPL	
67	Act			
68	Cambio	abctsol69 1	FPL	
		abctsol63 1	FPL	
70	Baja	abctsol68 1	FPL	
		abctsol60 1	FPL	
71	Movtos.			
72	Seleccion Requisicion	abctsol64 1	FPL	
		abctsol66 1	FPL	
		abctsol65 1	FPL	
73	Partidas Selectivo	abctsol67 1	FPL	
		abctsol65 1	FPL	
83	Partidas Automatico			
84	Alta			
85	Actual Req.	abctsol191 1	FPL	
		abctsol191 1	FPL	
86	Todas Req.	abctsol191 1	FPL	
		abctsol191 1	FPL	
88	Baja			
89	Actual Req.	abctsol185 1	FPL	
		abctsol185 1	FPL	
90	Todas Req.	abctsol185 1	FPL	
		abctsol185 1	FPL	
77	Prov.			
78	Asigna	abctsol183 1	FPL	
79	Consulta	abctsol190 1	FPL	
80	Req. Ant.	abctsol184 1	FPL	
81	Req. Sig.	abctsol188 1	FPL	
96	Sol Asoc.	abctsol101 1	FPL	

ID	Object	When	Routine Name	Class	Parameter
	Form	enter	abctsol160 1	FPL	
Func	f04		abctsol184 1	FPL	
Func	f05		abctsol188 1	FPL	
Func	f06		abctsol101 1	FPL	
10	sco_num_conse	exit	abctsol161 1	FPL	
			abctsol166 1	FPL	
			abctsol165 1	FPL	
9	alm_clave	exit	abctsol162 1	FPL	
50	req_num	exit	abctsol192 1	FPL	
26	alm_nombre				
3	sco_fec_sol				
5	sco_fec_lim_cotiz	exit	abctsol113 1	FPL	
6	sco_comentarios	exit	abctsol163 1	FPL	
			abctsol171 1	FPL	
51	wok_cont	exit	abctsol172 1	FPL	
25	req_des_gener				
27	req_de				
35	req_no				
36	wk_num_sol				
37	wk_req_ini	exit	abctsol165 1	FPL	
38	wk_req_fin	exit	abctsol166 1	FPL	
			abctsol165 1	FPL	
39	total_req				
43	des_clave				
44	des_nombre				
45	counter				
46	wk_par_sol				
28	wk_req_act				
48	wk_par_req				
49	wrk_formname				
52	com_clave				
53	com_nivel				
54	wok_comp_usu				
55	com_clave_sup				
56	ind_comp_5k				
59	req_num_conse				
60	prq_num_conse				
63	alm_clave_com				
67	fecha_hoy				
74	prq_csd_estat				
75	num_sol	exit	abctsol161 1	FPL	
			abctsol166 1	FPL	
			abctsol165 1	FPL	

JOIN INFORMATION

req_num_conse = sco_num_conse
alm_clave_com = alm_clave

APT Report completed.

10/01/93 18:07
Page 1

abtcot01 1
FORM

Form Name: abtcot01
Form Version: 1
Title: Cotizaciones
Description:
Initial Field: cot_clave
Readonly: no
Show Menu: yes
Character Move: yes
Inherit Function Keys: yes
X: 0
Y: 0
Width: 77
Height: 17

Form abtcot01 1 isn't passed any parameters.

MENU
Title: 'Cotizaciones'
Text: /
Value: 21
Key: /
Text: Close
Value: 23
Text: Otra Cotizacion
Value: 62
Text: Salida
Value: 65
Text: Help
Value: 22
Text: Act.
Value: 67
Text: Alta
Value: 72
Text: Baja
Value: 73
Text: Cambio
Value: 74
Text: Cons.
Value: 68
Text: Sol.
Value: 69

MENU

Text: Prov.
Value: 66
Text: C Ant.
Value: 70
Text: C Sig.
Value: 71
Text: Sol Assoc
Value: 76

No groups.

GROUPS

ID: 3
Prompt: Fecha Cotizacion
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 28
Database: abaste.dbo.cotizacion.cot_fec_cotiz
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: yes

FIELD: cot_fec_cotiz

ID: 2
Prompt: Cva. Cotizacion
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 12
Database: abaste.dbo.cotizacion.cot_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

FIELD: cot_clave

ID: 15
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 37
Database: abaste.dbo.proveedor.prv_rzo_socia
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

FIELD: prv_rzo_socia

ID: 30
Prompt: No. Solicitud
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

FIELD: sco_num_conse

ID: 38
Prompt: Requisicion
FIELD: alm_clave
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 36
Prompt:
FIELD: wk_req_ini
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 37
Prompt: -
FIELD: wk_req_fin
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 5
Prompt: Fecha Vigencia
FIELD: fecha_vig
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 4
Prompt: Moneda
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.cotizacion.cot_tip_moned
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes
FIELD: mon_clave

ID: 13
Prompt: Cond. Pago
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 3
Database: abaste.dbo.cotizacion.cot_cnd_pago
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes
FIELD: prv_cnd_pago

ID: 12
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Database: abaste.dbo.cotizacion.cot_dia_condi_pago
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes
FIELD: prv_dia_condi_pago

ID: 8
Prompt: Cond.
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 3
Database: abaste.dbo.cotizacion.cot_cnd_entrg
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes
FIELD: prv_cnd_entre

ID: 7 FIELD: prv_lug_entre

Prompt: Lugar
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.cotizacion.cot_lug_entry
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 48 FIELD: des_lugar

Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 6 FIELD: prv_tpo_entre

Prompt: Tiempo
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Database: abaste.dbo.cotizacion.cot_tpo_entry
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 11 FIELD: prv_dto_gener

Prompt: General
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Database: abaste.dbo.cotizacion.cot_dto_pront_pago
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 69
Prompt: Por Partida
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 7
Attributes:
Visible: yes
Stop On: yes
Highlight: yes
FIELD: prv_descto_pp

ID: 9
Prompt: Por Pto. Pago
In Gpto:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 7
Database: abaste.dbo.cotizacion.cot_dto_gener
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes
FIELD: prv_dto_pront_pago

ID: 10
Prompt: % a
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Database: abaste.dbo.cotizacion.cot_dia_pront_pago
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes
FIELD: prv_dia_pront_pago

ID: 41
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: prv_rfc_repre

ID: 42 FIELD: pdo_num_conse
Prompt: %

In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 31 FIELD: cotiz_no
Prompt: Cotizacion No.

In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 10
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 32 FIELD: cotiz_de
Prompt: de

In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 10
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 44 FIELD: mon_nom_ingre
Prompt:

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 19
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 45

FIELD: des_cnd_pago

Prompt:

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 22
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 47

FIELD: des_cnd_entre

Prompt:

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 49

FIELD: field_49

Prompt: ----- Entrega -----

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 50

FIELD: field_50

Prompt: ----- Descuento -----

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 51
Prompt: % FIELD: field_51
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 52
Prompt: Dias FIELD: field_52
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 1
Prompt: Proveedor/RFC FIELD: prv_rfc
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Database: abaste.dbo.cotizacion.prv_rfc
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 54
Prompt: wrk formname FIELD: wrk_formname
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 53

FIELD: counter

Prompt:

In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 55

FIELD: prv_nacionali

Prompt: a Consultar.

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Database: abaste.dbo.proveedor.prv_nacionali
Use Default: no
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 56

FIELD: fech_entre

Prompt:

In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 59

FIELD: num_dia

Prompt: Num. Dia.

In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 60 FIELD: dias_mas

Prompt: Dias
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 61 FIELD: alm_nombre

Prompt: Centro
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 36
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 63 FIELD: wok_comp_usu

Prompt: wok_comp_usu
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 64 FIELD: ind_comp_ok

Prompt: ind_comp_ok
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 65

FIELD: com_clave

Prompt: com clave
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 66

FIELD: com_clave_sup

Prompt: com clave_sup
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 68

FIELD: cot_des_lug

Prompt: cot_des_lug
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 35
Attributes:
Visible: no
Stop On: no
Highlight: no

ID	Name	Routine Name	Class	Parameter
21	/			
23	Close			
62	Otra Cotizacion	abctcot08 1	FPL	
65	Salida	abctcot08 1	FPL	
22	Help	help	Supp	
67	Act.			
72	Alta	abctcot05 1	FPL	
73	Baja	abctcot16 1	FPL	
74	Cambio	abctcot06 1	FPL	
68	Cons.	abctcot02 1	FPL	
69	Sol.	abctcot18 1	FPL	
66	Prov.	abctcot26 1	FPL	
70	C Ant.	abctcot04 1	FPL	
71	C Sig.	abctcot07 1	FPL	
76	Sol Ascc	abctcot101 1	FPL	

ID	Object	When	Routine Name	Class	Parameter
	Form	enter	abctcot08 1	FPL	
Func f02			abctcot02 1	FPL	
Func f04			abctcot26 1	FPL	
Func f05			abctcot04 1	FPL	
Func f06			abctcot07 1	FPL	
Func f03			abctcot18 1	FPL	
3	cot fec cotiz	exit	abctcot24 1	FPL	
2	cot_clave	exit	abctcot09 1	FPL	
15	prv_rzo socia	exit	abctcot27 1	FPL	
30	sco_num_conse	exit	abctcot10 1	FPL	
38	alm_clave	exit	abctcot29 1	FPL	
36	wk_feq_ini	exit	abctcot30 1	FPL	
37	wk_req_fin				
5	fecha_vig	exit	abctcot11 1	FPL	
4	mon_clave	exit	abctcot14 1	FPL	
13	prv_cnd_pago	exit	abctcot15 1	FPL	
12	prv_dia_condi_pago				
8	prv_cnd_entre	exit	abctcot17 1	FPL	
7	prv_lug_entre	exit	abctcot19 1	FPL	
48	des_lugar				
6	prv_tpo_entre	exit	abctcot28 1	FPL	
11	prv_dto_gener				
69	prv_desccto_pp				
9	prv_dto_pront_pago				
10	prv_dia_pront_pago				
41	prv_rfc_repre				
42	pdo_num_conse				
31	cotiz_no				
32	cotiz_de				
44	mon_nom_ingre				
45	des_cnd_pago				
47	des_cnd_entre				
1	prv_rfc	exit	abctcot12 1	FPL	
54	wrk_formname				
53	counter				
55	prv_nacionali				
56	fecH_entre				
59	num_dia				
60	dias_mes				
61	alm_nombre				
63	wok_comp_usu				
64	ind_comp_ok				
65	ccm_clave				
66	ccm_clave_sup				
68	cot_des_lug				

APT Report completed.

10/01/93 18:13
Page 1

abpegen01 1
FORM

Form Name: abpegen01
Form Version: 1
Title: Seleccion para orden de compra
Description:
Initial Field: alm_clave
Readonly: no
Show Menu: yes
Character Move: no
Inherit Function Keys: no
X: 0
Y: 0
Width: 77
Height: 17
Form Parameters:

Parameter	Type	Direction
alm_clave	field	input
wrk_num_conse	field	input
wrk_ped_conse	field	input

MENU

MENU

Title: 'Seleccion para orden de compra'

Text: /
Value: 21
Key: /

Text: Cancelar
Value: 23
Key: c

Text: Ayuda
Value: 22
Key: a

Text: Actualiza
Value: 126
Key: a

Text: Consulta
Value: 91
Key: c

Text: Requisiciones
Value: 92
Key: r

Text: Cotizaciones
Value: 93
Key: c

Text: Pedidos
Value: 94
Key: p

Text: Movimientos
Value: 95
Key: m

Text: Posic.
Value: 83
Key: o

Text: Seleccion
Value: 108
Key: s

10/01/93 18:13

Page 2

abpegen01 1
FORM

Text: Requisicion
Value: 109
Key: r

Text: Cotizacion
Value: 110
Key: c

Text: Proveedor
Value: 111
Key: p

Text: Rango:Requis.
Value: 125
Key: a

Text: \/
Value: 106

Text: \
Value: 107

Text: Menu
Value: 82
Key: m

Text: Actualiza
Value: 89
Key: a

Text: requisicion
Value: 100

Text: sol:Cotizacion
Value: 101

Text: cotizaciones
Value: 119

Text: pedido:Modific
Value: 102

Text: movimto:Importac.
Value: 103

Text: movimto:Nacional
Value: 120

Text: Consulta
Value: 88
Key: c

Text: requisiciones
Value: 96

Text: cotizaciones
Value: 97

Text: pedidos
Value: 98

Text: movimientos
Value: 99

Text: Salida del Sit.
Value: 104
Key: s

Text: Limpia
Value: 127
Key: l

ID: 55 GROUP: wrk_nums

In Group:
Prompt:
Desc:
Display Rows: 10
Initial Rows: 10
Extent: 10
Maximum Rows: 10
Attributes:
Visible: no
Stop On: no

ID: 58 GROUP: wrk_rng_nums

In Group:
Prompt:
Desc:
Display Rows: 10
Initial Rows: 10
Extent: 10
Maximum Rows: 10
Attributes:
Visible: no
Stop On: no

ID: 62 GROUP: wrk_cot_nums

In Group:
Prompt:
Desc:
Display Rows: 10
Initial Rows: 10
Extent: 10
Maximum Rows: 10
Attributes:
Visible: yes
Stop On: yes

ID: 75 GROUP: req_cot

In Group:
Prompt:
Desc:
Display Rows: 8
Initial Rows: 8
Extent: 8
Maximum Rows: 100
Attributes:
Visible: yes
Stop On: no

ID: 9 FIELD: com_clave

Prompt: Comprador
In Group:
Desc: Clave del comprador para seleccionar sus requisiciones
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 6 FIELD: alm_clave

Prompt: Centro
In Group:
Desc: Clave de almacen al que pertenecen las requisiciones a sel.
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 53 FIELD: wrk_uni_conse

Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Attributes:
Visible: no
Stop On: no
Highlight: yes

ID: 3 FIELD: wrk_inf_conse

Prompt: Requis. De
In Group:
Desc: WRK. Num. Consec. Rango inf. para seleccion de requisiciones
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 4 FIELD: wrk_sup_conse

Prompt: A
In Group:
Desc: WRK. Num.consec. Rango sup. para seleccion de requisiciones
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 2 FIELD: wrk_cot_clave

Prompt: Cotizacion
In Group:
Desc: Clave de Cotizacion para seleccion de Cotizaciones
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 34
Prompt: FIELD: wrk_req_selec
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 3
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 26
Prompt: Proveedor FIELD: wrk_rzo_socia
In Group:
Desc: Razon Social del proveedor para seleccionar cotizaciones
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 40
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 48
Prompt: FIELD: wrk_prv_rfc
In Group:
Desc: RFC del proveedor para seleccion de cotizaciones
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 82
Prompt: \ Desc. FIELD: wok_des_part
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 7
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 70
Prompt: Prog. FIELD: wrk_tpo_pedid
In Group:
Desc: Indicador de si el pedido es programado o no.
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 50 FIELD: wrk_num_conse
Prompt: Pedido Num.
In Group:
Desc: Numero de pedido asignado o dado para incluir requisiciones
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 51 FIELD: wrk_ped_conse
Prompt:
In Group:
Desc: Numero de anexo del pedido. Asignado por sistema.
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 7 FIELD: alm_nombre
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 40
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 8 FIELD: titl
Prompt: Requisicion
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 10 FIELD: com_nombre
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 40
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 11
Prompt: wrk com_nivel
FIELD: wrk_com_nivel
In GrGup:
Desc: wrk. nivel de autoriz. del comprador
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 12
Prompt: com_nivel
FIELD: com_nivel
In GrGup:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 13
Prompt: wrk com_clave
FIELD: wrk_com_clave
In GrGup:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 39
Prompt: wk_prova
FIELD: wk_prova
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 52
Prompt: Selec.
FIELD: wrk_num_selec
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 3
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 54 FIELD: wrk_numeros
Prompt: wrk_numeros
In Group: wrk_nums
Desc: Grupo para almacenar num. requisicion dados en el campo
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 6
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 56 FIELD: wrk_inf_nums
Prompt: wrk_inf_nums
In Group: wrk_rng_nums
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 57 FIELD: wrk_sup_nums
Prompt: wrk_sup_nums
In Group: wrk_rng_nums
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 59 FIELD: wrk_inda
Prompt: wrk_inda
In Group:
Desc: Indice para grupo wrk_nums. Indica numeros cargados de
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 60 FIELD: wrk_indb
Prompt: wrk_indb
In Group:
Desc: Indice para grupo wrk_rng_nums. Indica rangos de requisicion
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 61
Prompt: wrk cot_numero
In Group: wrk cot_nums
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: wrk_cot_numero

ID: 63
Prompt: wrk indc
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: wrk_indc

ID: 64
Prompt: wrk ind_pedido
In Group:
Desc: indica si existe pedido
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: wrk_ind_pedido

ID: 65
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 30
Attributes:
Visible: no
Stop On: no
Highlight: yes

FIELD: wrk_mensaje

ID: 66
Prompt: /
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wr_diag

ID: 67
Prompt: wrk wrk
In Group:
Desc: Campo de trabajo usado en varios procedimientos para
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 6
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: wrk_wrk

ID: 68
Prompt:
In Group:
Desc: Campo usado como indice para carga de grupo req_cot usado
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 3
Attributes:
Visible: yes
Stop On: no
Highlight: no

FIELD: wrk_sus

ID: 69
Prompt: wrk posicion
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: wrk_posicion

ID: 71
Prompt: N/S
In Group:
Desc: Indicador de si el pedido es programado o no (S/N)
Appearance Definition:
Datatype: char
Field Type: trim
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: si_no

ID: 72
Prompt: ped_tpo_pedido
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

FIELD: ped_tpo_pedido

ID: 73
Prompt: wrk_sol_prove
In GrGup:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: wrk_sol_prove

ID: 74
Prompt: MUESTRA COTIZACIONES ? S/N
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: wrk_sel_sino

ID: 31
Prompt:
In Group: req_cot
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 3
Attributes:
Visible: yes
Stop On: yes
Highlight: no
FIELD: wrk_sel_requi

ID: 16
Prompt: Requis.
In Group: req_cot
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 5
Attributes:
Visible: yes
Stop On: no
Highlight: no
FIELD: req_num_conse

ID: 42
Prompt: St
In Group: req_cot
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: no
Highlight: no
FIELD: req_cve_statu

ID: 30
Prompt:

FIELD: wrk_sel_cotiz

In Group: req_cot
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 17

FIELD: cot_clave

Prompt: Cotizacion
In Group: req_cot
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 46

FIELD: fec_vigen

Prompt: Vigen.
In Group: req_cot
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 6
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 29

FIELD: wrk_fec_venci

Prompt: Ven
In Group: req_cot
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 20

FIELD: prv_rzo_socia

Prompt: Proveedor
In Group: req_cot
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 40
Database: abaste.dbo.proveedor.prv_rzo_socia
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 19 FIELD: prv_rfc
Prompt: prv_rfc;
In GrOp: req_cot
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Database: abaste.dbo.proveedor.prv_rfc
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 76 FIELD: wrk_formname
Prompt: wrk_formname
In GrOp:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 9
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 77 FIELD: wrk_ind_selec
Prompt: wrk_ind_selec
In GrOp:
Desc: wrk. determina si muestra cotizacion se solicitado desde
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 78 FIELD: wrk_ind_llama
Prompt: wrk_ind_llama
In GrOp:
Desc: wrk. indica si fue llamado por otra forma de pedido
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 79 FIELD: com_clave_sup
Prompt: com_clave_sup;
In GrOp:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 80

FIELD: rzo

Prompt: rzo

In Group:

Desc:

Appearance Definition:

Datatype: char

Field Type: entry

Max Len: 40

Attributes:

Visible: no

Stop On: no

Highlight: no

ID: 81

FIELD: rfc

Prompt: rfc

In Group:

Desc:

Appearance Definition:

Datatype: char

Field Type: entry

Max Len: 20

Attributes:

Visible: no

Stop On: no

Highlight: no

ID	Name	Routine Name	Class	Parameter
21	/			
23	Cancelar	abpegen93 1	FPL	
22	Ayuda	exit_routine	Ext	
126	Actualiza	help	Supp	
91	Consulta	abpegen15 1	FPL	
92	Requisiciones			
93	Cotizaciones	abreqin05 1	Form	
94	Pedidos			
95	Movimientos			
83	Posic.			
108	Seleccion	abpegen90 1	FPL	
109	Requisicion	abpegen90 1	FPL	
110	Cotizacion	abpegen90 1	FPL	
111	Proveedor	abpegen90 1	FPL	
125	Rango:Requis.	abpegen90 1	FPL	
106	X	abpegen06 1	FPL	
107		abpegen07 1	FPL	
82	Menu			
89	Actualiza			
100	requisicion	abpegen92 1	FPL	
		exit_routine	Ext	
101	sol:Cotizacion	abpegen92 1	FPL	
		exit_routine	Ext	
119	cotizaciones	abpegen92 1	FPL	
		exit_routine	Ext	
102	pedido:Modific	abpegen92 1	FPL	
		exit_routine	Ext	
103	movimto:Importac.	abpegen92 1	FPL	
		exit_routine	Ext	
120	movimto:Nacional	abpegen92 1	FPL	
		exit_routine	Ext	
88	Consulta			
96	requisiciones	abpegen91 1	FPL	
		exit_routine	Ext	
97	cotizaciones			
98	pedidos			
99	movimientos			
104	Salida del Sit.	abpegen49 1	FPL	
127	Limpia	abpegen09 1	FPL	

ID	Object	When	Routine Name	Class	Parameter
	Form	enter	abpegen00 1	FPL	
Func f04			abpegen06 1	FPL	
Func f05			abpegen07 1	FPL	
Func f03			abpegen09 1	FPL	
Func f01			abpegen15 1	FPL	
55	wrk_nums				
58	wrk_rng_nums				
62	wrk_cot_nums				
75	req_cot				
9	com_clave	exit	abpegen04 1	FPL	
6	alm_clave	exit	abpegen01 1	FPL	
53	wrk_uni_conse	exit	abpegen28 1	FPL	
3	wrk_inf_conse	exit	abpegen02 1	FPL	
4	wrk_sup_conse	exit	abpegen13 1	FPL	
2	wrk_cot_clave	exit	abpegen12 1	FPL	
34	wrk_req_selec	exit	abpegen05 1	FPL	
26	wrk_rzo_socia	exit	abpegen11 2	FPL	
48	wrk_prv_rfc	exit	abpegen11 1	FPL	
82	wok_des_part	exit	abpegen104 1	FPL	
70	wrk_tpo_pedid	exit	abpegen41 1	FPL	
50	wrk_num_conse				
51	wrk_ped_conse				
7	alm_nomBre	exit	abpegen29 1	FPL	
			abpegen14 1	FPL	
			abpegen10 1	FPL	
			abpegen08 1	FPL	
			abpegen43 1	FPL	
10	com_nombre				
11	wrk_com_nivel				
12	com_niv01				
13	wrk_com_clave				
39	wk_prove				
52	wrk_num_selec				
54	wrk_numeros				
56	wrk_inf_nums				
57	wrk_sup_nums				
59	wrk_inda				
60	wrk_inda				
61	wrk_cot_numero				
63	wrk_inda				
64	wrk_ind_pedido				
65	wrk_mensaje				
67	wrk_wrk				
68	wrk_sus				
69	wrk_posicion				
72	ped_tpo_pedid				
73	wrk_sel_prove				
74	wrk_sel_sino	exit	abpegen42 1	FPL	
31	wrk_sel_requi	enter	abpegen25 1	FPL	
16	req_num_conse	enter	abpegen05 1	FPL	
42	req_cve_estado				
30	wrk_sel_cotiz				

17 cot_clave
46 fec_vigen
29 wrk_fec_venci
20 prv_rzo_socia
19 prv_rfc
76 wrk_formname
77 wrk_ind_selec
78 wrk_ind_llama
79 com_clave_sup
80 rzo
81 rfc

APT Report completed.

10/01/93 18:19
Page 1

abmmnov01 1
FORM

Form Name: abmmnov01
Form Version: 1
Title: MOVIMIENTOS DE MATERIAL
Description:
Initial Field: mov_tipo
Readonly: no
Show Menu: yes
Character Move: no
Inherit Function Keys: no
X: 0
Y: 0
Width: 77
Height: 20
Form Parameters:

Parameter	Type	Direction
agt_clave	field	both
tra_clave	field	both
agt_nombre	field	input
por_trafico	field	input

MENU
Title: 'MOVIMIENTOS DE MATERIAL'
Text: /
Value: 21
Key: /
Text: Close
Value: 23
Text: Otro Trafico
Value: 69
Text: Otro Movimiento
Value: 70
Text: Otro Pedido
Value: 90
Text: Salida
Value: 71
Text: Movto
Value: 65
Key: m
Text: Alta
Value: 73
Key: a
Text: Baja
Value: 67
Key: b
Text: Cambio
Value: 68
Key: c
Text: Partida
Value: 72
Key: p
Text: Todas
Value: 83
Key: t

MENU

Text: Cancela
Value: 82
Key: c

Text: Baja
Value: 81
Key: b

Text: Consultas
Value: 74
Key: c

Text: Pedidos por Movimiento
Value: 85
Key: p

Text: Movimientos por Pedido
Value: 86
Key: m

Text: Situación por Partida
Value: 88
Key: s

Text: Bultos
Value: 61
Key: b

Text: Facturas
Value: 87
Key: f

ID: 2 FIELD: tra_clave
Prompt: TrSfico
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Database: abaste.dbo.movimiento_material.tra_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: yes

ID: 4 FIELD: mov_tipo
Prompt: Tipo
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.movimiento_material.mov_tipo
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 3 FIELD: mov_num_conse
Prompt: Movimiento
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Database: abaste.dbo.movimiento_material.mov_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 6 FIELD: mov_fecha
Prompt: Fecha
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 27
Database: abaste.dbo.movimiento_material.mov_fecha
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 10 FIELD: mov_tip_trans

Prompt: Transporte
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Database: abaste.dbo.movimiento_material.mov_tip_trans
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 9 FIELD: mov_gui_talon

Prompt: Guña
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Database: abaste.dbo.movimiento_material.mov_gui_talon
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 7 FIELD: mov_cnd_flete

Prompt: Cond. Flete
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Database: abaste.dbo.movimiento_material.mov_cnd_flete
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 8 FIELD: mov_remitente

Prompt: Remitente
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 40
Database: abaste.dbo.movimiento_material.mov_remitente
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 57
Prompt: FIELD: ped_status
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 50
Prompt: FIELD: alm_nombre
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 40
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 42
Prompt: FIELD: ppe_num_conse
In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 4
Database: abaste.dbo.partida_pedido.ppe_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 43
Prompt: FIELD: mov_cantidad
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Database: abaste.dbo.movimiento_material.mov_cantidad
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 21
Prompt: Alm
FIELD: alm_clave
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Database: abaste.dbo.partida_pedido.alm_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 22
Prompt:
FIELD: ped_clave
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 10
Database: abaste.dbo.partida_pedido.ped_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 23
Prompt:
FIELD: ped_num_conse
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Database: abaste.dbo.partida_pedido.ped_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: yes

ID: 58
Prompt:
FIELD: ped_fec_statu
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 27
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID: 24
Prompt: Part
FIELD: gp_ppc_num_conse
In Group: partmov
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 4
Database: abaste.dbo.partida_pedido.ppc_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: no

ID: 38
Prompt: Descripcion
FIELD: gp_des_nombre
In Group: partmov
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 255
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 27
Prompt: Cant.
FIELD: gp_ppc_cantidad
In Group: partmov
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Database: abaste.dbo.partida_pedido.ppc_cantidad
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 106
Prompt: Ev
FIELD: gp_mov_ant
In Group: partmov
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 91
Prompt: Ant.
FIELD: gp_eve_anter
In Group: partmov
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 36
Prompt: Acumu
FIELD: gp_can_acumu
In Group: partmov
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 5
Prompt: Cant.
FIELD: gp_mov_cantidad
In Group: partmov
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Database: abaste.dbo.movimiento_material.mov_cantidad
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 34
Prompt: ST
FIELD: gp_ppe_cve_statu
In Group: partmov
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.partida_pedido.ppe_cve_statu
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 74
Prompt: gp_ppe fec statu
FIELD: gp_ppe_fec_statu
In Group: partmov
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 27
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 31
Prompt: gp_des_clave: FIELD: gp_des_clave
In Group: partmov
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Database: abaste.dbo.partida_pedido.des_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 98
Prompt: gp_ppc_cve_progr: FIELD: gp_ppc_cve_progr
In Group: partmov
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 45
Prompt: FIELD: des_nombre
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 255
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 46
Prompt: FIELD: ppc_cantidad
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Database: abaste.dbo.partida_pedido.ppc_cantidad
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 47
Prompt: FIELD: wk_can_acumu
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 48
Prompt: FIELD: ppe_cva_statu

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.partida_pedido.ppe_cva_statu
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 59
Prompt: FIELD: des_clave

In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Database: abaste.dbo.partida_pedido.des_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 26
Prompt: prq_num_conse: FIELD: prq_num_conse

In Group:
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Database: abaste.dbo.partida_pedido.prq_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 61
Prompt: prv_rfc: FIELD: prv_rfc

In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 20
Database: abaste.dbo.factura_proveedor.prv_rfc
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 69
Prompt: wk indice: FIELD: wk_indice
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 71
Prompt: wk total: FIELD: wk_total
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 72
Prompt: sw_pos descr FIELD: sw_pos_descr
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 73
Prompt: FIELD: ppe_fec_statu
In Group:
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 27
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 76
Prompt: wk can_acumu_RF FIELD: wk_can_acumu_RF
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 77 FIELD: wk_can_acumu_RX
Prompt: wk can acumu_RX
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 78 FIELD: wk_can_acumu_EM
Prompt: wk can acumu_EM
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 79 FIELD: wk_can_acumu_RA
Prompt: wk can acumu_RA
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 81 FIELD: wk_mov_tipo
Prompt: wk mov_tipo
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 82 FIELD: prv_nacionali
Prompt: prv nacionali
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 62
Prompt: Fact
In Group: partfac
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 14
Database: abaste.dbo.factura_trafico.ftr_num_factu
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: no
FIELD: ftr_num_factu

ID: 67
Prompt: Fecha
In Group: partfac
Desc:
Appearance Definition:
Datatype: datetime
Field Type: entry
Max Len: 27
Database: abaste.dbo.factura_trafico.ftr_fec_factu
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no
FIELD: ftr_fec_factu

ID: 65
Prompt: Importe
In Group: partfac
Desc:
Appearance Definition:
Datatype: money
Field Type: entry
Max Len: 24
Database: abaste.dbo.factura_trafico.ftr_imp_factu
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no
FIELD: ftr_imp_factu

ID: 66
Prompt: Mon
In Group: partfac
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.factura_trafico.mon_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no
FIELD: mon_clave

ID: 88
Prompt: sw_can_acumu_EM
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
FIELD: sw_can_acumu_EM

ID: 89
Prompt: sw_xei_movto
In Group:
Desc:
Appearance Definition:
Datatype: bit
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: sw_xei_movto

ID: 90
Prompt: Proveedor
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 40
Attributes:
Visible: yes
Stop On: no
Highlight: no
FIELD: prv_rzo_socia

ID: 94
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: wk_ave_ater

ID: 96
Prompt: Pedido No.
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: sw_ped_no

ID: 97
Prompt: de
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no
FIELD: sw_ped_da

ID:100
Prompt: FIELD: ppe_cve_progr
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 1
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:101
Prompt: 1 Si viene de trafico FIELD: por_trafico
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:102
Prompt: wrk formname FIELD: wrk_formname
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:104
Prompt: Mov FIELD: mov_ant
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Attributes:
Visible: no
Stop On: no
Highlight: no

ID:108
Prompt: wk mov cantidad FIELD: wk_mov_cantidad
In Group:
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 25
Prompt: req_num_conse: FIELD: req_num_conse
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Database: abaste.dbo.partida_pedido.req_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 109
Prompt: sta_nivel FIELD: sta_nivel
In Group:
Desc:
Appearance Definition:
Datatype: tinyint
Field Type: entry
Max Len: 3
Attributes:
Visible: no
Stop On: no
Highlight: no

ID: 15
Prompt: Bul FIELD: gp_bul_num_conse
In Group: partbul
Desc:
Appearance Definition:
Datatype: smallint
Field Type: entry
Max Len: 6
Database: abaste.dbo.bulto.bul_num_conse
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: yes
Highlight: no

ID: 16
Prompt: Cant. FIELD: gp_bul_cantidad
In Group: partbul
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Database: abaste.dbo.bulto.bul_cantidad
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 17
Prompt: Tp
FIELD: gp_bul_tipo
In Group: partbul
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.bulto.bul_tipo
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 18
Prompt: Peso
FIELD: gp_bul_peso
In Group: partbul
Desc:
Appearance Definition:
Datatype: float
Field Type: entry
Max Len: 20
Database: abaste.dbo.bulto.bul_peso
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 19
Prompt: U.H.
FIELD: gp_ume_clave
In Group: partbul
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 4
Database: abaste.dbo.bulto.bul_uni_clave
Use Default: yes
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: no

ID: 111
Prompt: Compania
FIELD: cia_traf
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: yes
Stop On: no
Highlight: yes

ID:113
Prompt: Cia
In Group:
Desc:
Appearance Definition:
Datatype: int
Field Type: entry
Max Len: 11
Attributes:
Visible: no
Stop On: no
Highlight: yes

FIELD: cia

ID:115
Prompt:
In Group:
Desc:
Appearance Definition:
Datatype: char
Field Type: entry
Max Len: 2
Database: abaste.dbo.pedido.com_clave
Use Default: no
Run Rule: no
Attributes:
Visible: yes
Stop On: no
Highlight: yes

FIELD: com_clave

ID	Name	Routine Name	Class	Parameter
21	/			
23	Close			
69	Otro Trafico	abmmnov81 1	FPL	
70	Otro Movimiento	abmmnov88 1	FPL	
90	Otro Pedido	abmmnov65 1	FPL	
71	Salida	abut fvl 1	FPL	
		abmmnov67 1	FPL	
65	Movto			
73	Alta	abmmnov12 1	FPL	
67	Baja	abmmnov22 1	FPL	
68	Cambio	abmmnov32 1	FPL	
72	Partida			
83	Todas	abmmnov01 2	FPL	
82	Cancela	abmmnov02 2	FPL	
81	Baja	abmmnov10 2	FPL	
74	Consultas			
85	Pedidos por Movimiento	abmmnov38 1	FPL	
86	Movimientos por Pedido	abmmnov90 1	FPL	
88	Situacion por Partida	abmmnov34 1	FPL	
61	Bultos	abmmnov21 1	FPL	
87	Facturas	abmmnov07 1	FPL	
		abmmnov00 1	FPL	

ID	Object	When	Routine Name	Class	Parameter
	Form	enter	abmmmov00 1	FPL	
			abmmmov01 1	FPL	
Func f05			abmmmov07 1	FPL	
Func f04			abmmmov21 1	FPL	
Func f06			abmmmov73 1	FPL	
107	partmov				
87	partfac				
56	partbul				
1	agt_clave	exit	abmmmov46 1	FPL	
41	agt_nombre				
2	tra_clave	exit	abmmmov01 1	FPL	
4	mov_tipo	exit	abmmmov04 2	FPL	
3	mov_num_conse	exit	abmmmov07 2	FPL	
6	mov_fecha	exit	abmmmov100 1	FPL	
10	mov_tip_trans				
9	mov_gul_talon				
7	mov_end_flete	exit	abmmmov70 1	FPL	
8	mov_remitente	enter	abmmmov76 1	FPL	
			abmmmov77 1	FPL	
			abmmmov43 1	FPL	
			abmmmov35 1	FPL	
			abmmmov09 1	FPL	
21	alm_clave				
22	ped_clave				
23	ped_num_conse				
58	ped_fec_statu				
57	ped_staEus				
50	alm_nombre				
42	ppe_num_conse	exit	abmmmov42 1	FPL	
43	mov_canEidad	exit	abmmmov01 9	FPL	
24	gp_ppe_num_conse	enter	abmmmov50 1	FPL	
38	gp_des_nombre				
27	gp_ppe_cantidad				
106	gp_mov_ant				
91	gp_eve_antor				
36	gp_can_acumu				
5	gp_mov_cantidad				
34	gp_ppe_cve_statu				
74	gp_ppe_fec_statu				
31	gp_des_clave				
98	gp_ppe_cve_progr				
45	des_nombre				
46	ppe_cantidad				
47	wk_can_acumu				
48	ppe_cve_statu				
59	des_clave				
26	prq_num_conse	exit	abmmmov06 2	FPL	
61	prv_rfc	exit	abmmmov05 2	FPL	
			abmmmov08 1	FPL	
			abmmmov64 1	FPL	
			abmmmov64 2	FPL	
69	wk indice				
71	wk_total				
72	sw_pos_descr				
73	ppe_fec_statu				

76	wk_can_acumu_RF			
77	wk_can_acumu_RX			
78	wk_can_acumu_EM			
79	wk_can_acumu_RA			
81	wk_mov_tipo			
82	prv_nacionali			
62	ftr_num_factu			
67	ftr_fec_factu			
65	ftr_imp_factu			
66	mon_clave			
88	sw_can_acumu_EM			
89	sw_axi_movto			
90	prv_rsõ_socia			
94	wk_ave_ant			
96	sw_ped_no			
97	sw_ped_de			
100	pps_cvs_prog			
101	por_trafico			
102	wrk_formname			
104	mov_ant			
108	wk_mov_cantidad			
25	req_num_conse	exit	abmmmov02 1	FPL
			abmmmov03 2	FPL
109	sta_nivel			
15	qp_bul_num_conse	exit	abmmmov71 2	FPL
16	qp_bul_cantidad			
17	qp_bul_tipo			
18	qp_bul_peso			
19	qp_ume_clave			
111	cia_traf			
113	cia			
115	com_clave			

APT Report completed.

APENDICE E.

DOCUMENTACION RECOPIADA

NOTA: En este apéndice sólo se incluye una parte de la información referente al mismo, debido a lo extenso del sistema.



PRODUCTOS QUIMICOS, S.A. DE C.V.

ALMACEN MAT. Y REFRACTARIOS
 SUBSIDIARIA DE INDUSTRIAS PENOLAS, S. A. DE C. V.
 -OFICINAS GENERALES- CORRESPONDENCIA A:
 Rlo de la Plata No. 48 APDO. POSTAL No. 686
 Colonia Cuauhtémoc C.P. 06500 MEXICO 06000, D.F.
 México, D.F. Tel: 2-86-81-33 FAX: 286-21-77

CITENSE ESTOS
 NUMEROS EN TODA
 CORRESPONDENCIA

1a. MODIFICACION AL PEDIDO

ESTE PEDIDO SUBSTITUYE AL DE FECHA: 04/10/93

PROVEEDOR: PHE 380607001
 PETROLEOS MEXICANOS
 CORREGIDORA No. 534 PISO 7
 ESQ. CON FRANCISCO I. MADERO
 COATZACOALCOS, VER. C.P. 96400
 VERACRUZ MEXICO
 ATTN: ING. IGNACIO CRUZ L.
 TELEFONO: (921)2-66-17 FAX: (921)2-95-74
 COTZ:

F. PEDIDO	EMBARQUE	PEDIDO NUMERO
17/11/93	07/10/93 O ANTES	15GP 00024-00 HOJA: 1

CONVENIO DE SUMINISTRO No.:
 CUENTA CARGO: 0000-0000-0000-0000
 AUT. DESEMBOLSO: 78-0000- -000

FACTURAR A: PRODUCTOS QUIMICOS S.A. DE C.V.
 CARRETERA MEXICO-PACHUCA KM 11.5
 SANTA CLARA, ECATEPEC DE MORELOS
 EDO. DE MEXICO C.P. 25573
 ENVIAR FACTURAS AUTOGRAFAS Y DOCTOS. EMB:
 KM. 11.5 CARRET. MEXICO-PACHUCA
 SANTA CLARA ECATEPEC DE MORELOS
 ESTADO DE MEXICO, C.P. 25573
 AT'N: DEPTO. DE CUENTAS POR PAGAR

CONSIGNAR VIA: NOSOTROS RECOGEMOS
 PRODUCTOS QUIMICOS S.A. DE C.V.
 ALMACEN MAT. Y REFRACTARIOS
 KM 11.5 CARRETERA MEXICO-PACHUCA
 SANTA CLARA, ECATEPEC DE MORELOS
 EDO. DE MEXICO 25573, MEXICO

CONDICIONES DE PAGO:
 ANTICIPO TOTAL

DESC. POR P/PAGO
 A DIAS

CONDICIONES DE ENTREGA: LAB
 REFINERIA POZA RICA, VER.

AZUFRE Lin 38 S. Lin

PA RT	CANTI DAD	UNIDAD PARTE No.	DESCRIPCION	CODIGO	CNT RCB	PRECIOS EN: Moneda Nacional. UNITARIO	TOTAL
1	20.0000	TON M GP 24 1	AZUFRE CRUDO A GRANEL EN LAJAS, de acuerdo a la siguiente especificacion: PROPIEDADES QUIMICAS. Pureza como S (Base seca) 99.3% min Humedad 1.0% max Acidez libre como H2SO4.. 0.07% max Cenizas 0.4% max			175.50	3,510.00

AUTORIZACIONES

 MARIO SOSA JIMENEZ

VALOR EN: Moneda Nacional.	N\$3,861.00
VALOR TOTAL MN:	N\$3,861.00
COMPRADOR: ING. PEDRO ZUIGA	TEL:

TRANSPORTE	TALON	FECHA	RECIBIDO POR: FIRMA:	FECHA ELABORACION	30/09/93
FACT. REMS.	PESO T.	# BULTOS		FECHA RECEPCION COMPRA	01/10/93
				FECHA ESTIMADA	02/10/93
				FECHA REQUERIDA	13/10/93
RECIBO #	PARCIAL	FECHA R.	NOMBRE	MONTO EST. PRIORIDAD Normal	TIPO: A
	TOTAL			AUTORIZA: FOLIO: SOLIC. ANTONIO LOPEZ	
				DEPTO: EPO:	OT:

PROV/EXPED/CONSEC/EXPEDI/ALM/REC/C. X PGR/OR. DE PAGO/IMP/AG. AD/AG.EMB
 FECHA: Nov 19 1993, HORA: 19:40



PRODUCTOS QUIMICOS S.A. DE C.V.

ALMACEN MAT. Y REFRACTARIOS
SUBSIDIARIA DE INDUSTRIAS PENOLÉS, S. A. DE C. V.
-OFICINAS GENERALES-
Río de la Plata No. 48
Colonia Cuauhtémoc C.P. 06500
México, D.F. Tel: 2-86-81-33

CORRESPONDENCIA A:
APDO. POSTAL No. 686
MEXICO 06000, D.F.
FAX: 286-21-77

CITENSE ESTOS
NUMEROS EN TODA
CORRESPONDENCIA

1a. MODIFICACION AL PEDIDO

ESTE PEDIDO SUSTITUYE AL DE FECHA: 04/10/93

PROVEEDOR: PETROLEOS MEXICANOS	PME 380607001	F. PEDIDO 17/11/93	EMBARQUE 07/10/93 O ANTES	PEDIDO NUMERO 15GP 00024-00 HOJA: 2
CORREGIDORA No. 534 PISO 7				

PA RT	CANTI DAD	UNIDAD PARTE No.	DESCRIPCION	CODIGO	CNT RCB	PRECIOS EN: Moneda Nacional. UNITARIO	TOTAL
			PROPIEDADES FISICAS.				
			Densidad aparente 1.1 - 1.2 Punto de fusion 112.8 C				
			SUBTOTAL BRUTO				N\$3,510.00
			I.V.A.	10.00%			N\$351.00

NOTAS:

EL PRODUCTO AMPARADO EN EL PRESENTE PEDIDO SERA RETIRADO DE LA REFINE-
RIA DE POZA RICA, VER.
EL PRECIO SERA EL VIGENTE A LA FECHA DE EMBARQUE.

NOTA 1a. MODIFICACION.

Se emite la presente modificacion para ajustar la cantidad total de
acuerdo a lo recibido en la planta.

AUTORIZACIONES MARIO SOSA JIMENEZ	VALOR EN: Moneda Nacional.	N\$3,861.00
	VALOR TOTAL MN:	N\$3,861.00
	COMPRADOR: ING. PEDRO ZUÑIGA	TEL:



MET-MEX PEÑOLES, S.A. DE C.V.

SUBSIDIARIA DE INDUSTRIAS PEÑOLES, S. A. DE C. V.

-OFICINAS GENERALES-

Río de la Plata No. 48
Colonia Cuauhtémoc C.P. 06500
México, D.F. Tel: 2-86-81-33

CORRESPONDENCIA A:
RFD. POSTAL No. 686
MEXICO 06000, D.F.
FAX: 286-21-77

CITENSE ESTOS
NÚMEROS EN TODA
CORRESPONDENCIA

SOLICITUD DE COTIZACION

PROVEEDOR: PETROLEOS MEXICANOS	y/o	FECHA DE EMISION	LIMITE F COTIZAR.	SOLICITUD No.
		02/12/91	11/11/93	AFT 03958 00024 HOJA: 1
ATTN:	C.P.	ENTREGA REQUERID	ENTREGA OFRECIDA	FAVOR COTIZAR LAB FABRICA SUS MEJORES:
Telefono:	Fax:	15/10/91		-PRECIOS -CONDS. DE PAGO -TIEMPO ENTREGA

CONDICIONES DE PAGO:	DESC. POR P/PAGO A DIAS	CONDICIONES DE ENTREGA:
----------------------	----------------------------	-------------------------

MOTOREDUCTORES MARCA PHILADELPHIA MODELO MTE-50-PTO .

PA RT	CANTI DAD	UNIDAD PARTE No.	DESCRIPCION	CODIGO	CNT RCB	PRECIOS EN: UNITARIO	TOTAL
1	2.00	PIEZA APT 3958 1	MOTOREDUCTORES MARCA PHILADELPHIA M ODELO 3805-S , MODELO PTO PARA --MOTOR DE 7.5 H.P., MARCA US TIPOTCC V, 4 POLOS, 3 FASES, 60 Hz, 1750RPM PARA ENTRADA DEL REDUCTOR Y 100RPM D E SALIDA, RELACION DE REDUCCION 17.5:1 , CON FACTOR DE SERVICIO DE -2.4 , C APACIDAD MECANICA DEL REDU--CTOR DE 18.2 HP, INCLUYE EL MEDIO --COPLER PARA LA ADAPTACION DE FLECHAAGITADORA Y BASE SOPORTE PARA EL --MOTOR Y PARA FIJAR EL MOTOREDUCTORA PUENTE DE SO PORTE . .SE DEBERA APLICAR RECUBRIMIE NTO EPO-XICO DE ALTOS SOLIDOS A 2 CA PAS PARA OBTENER UN ESPESOR DE 8 MILE SIMAS --COMO PRIMARIO Y ACABADO , TI PO -- --AMERLOCK 400 MARCA AMERCOAT, PARA --PROTEGER EL EQUIPO DE CORROS ION POR--SALPICADURAS Y VAPORES DE AC IDO SUL--PURICO Y CAL .EL CARGO POR E STE RECUBRIMIENTO ES --DE : \$ 350,000 .00 POR CADA REDUCTOR. (SIN I.V.A.) .				

PARA ACLARACIONES COMUNICARSE CON:	NOMBRE Y FIRMA	COTIZACION: No: FECHA:
ING JOSE ANTONIO CORONA		VIGENCIA:

Resumen General - Agosto 1993

REQUISICIONES

Ofna/Comprador	Recibidas Reqs Part	Reqs Can.	Reqs Col.	Ant +	Jun 2	Jul 1	Ago 1	Reqs Venc	Total Reqs	Anti Mes	Prom Col.	
Mexico, D.F.	172	604	31	134	16	34	52	123	102	225	431	45
Equipos y Proye	150	496	24	83	12	34	50	106	96	202	359	52
04 J. ACUNA	50	103	4	19	0	4	7	37	11	48	63	35
05 V. OROZCO	25	81	8	18	2	15	11	14	28	42	89	56
06 G. HUERTA	29	151	9	8	1	5	15	22	21	43	71	69
07 J. A. CORONA	27	91	3	29	0	1	2	16	3	19	23	59
08 A. RODRIGU	19	70	0	9	9	9	15	17	33	50	113	41
Materias Primas	42	108	7	51	4	0	2	17	6	23	72	33
12 N. PEREZ	27	93	4	32	0	0	1	16	1	17	18	33
14 R. Alanís	7	7	0	11	0	0	0	1	0	1	1	38
15 P. ZUITICA	8	8	3	8	4	0	1	0	5	5	53	28

PEDIDOS

Ofna/Coop	Elaborada Peds Part	Modi fic.	Cent ados	Ant +	Jun 3	Jul 2	Ago 1	Peds Venc	Total Peds	Anti Mes	Prom Cump	
Mexico, D.F.	150	373	62	156	163	80	100	123	466	878	2132	23
Equip y Proy	80	236	41	100	121	65	78	83	347	560	1609	29
04 J. ACUNA	18	41	11	23	27	15	18	16	76	114	330	25
05 V. OROZCO	11	38	9	18	35	10	14	14	73	121	397	35
06 G. HUERTA	12	17	11	27	15	9	16	14	54	78	245	39
07 J. A. CORON	29	79	2	32	30	13	14	21	80	156	183	21
08 A. RODRIG	10	61	8	0	12	18	16	18	64	91	254	0
Materias P.	70	137	21	56	42	15	22	40	119	318	523	11
12 N. PEREZ	51	118	11	39	7	3	9	26	45	74	140	13
14 R. Alanís	11	11	4	15	20	7	12	8	47	126	235	1
15 P. ZUITICA	8	8	6	12	15	5	1	6	27	118	148	6



REPORTE DEL DEPTO CORP ABASTECIMIENTO

AREA: Materias Primas
 COMPRADOR 15 ING. PEDRO ZUIGA

RECIBIDAS REQNS PARTS C

REQUISICIONES: 30/09/93

EN EL MES 13 15
 ACUMULADA 156 172

-RESUMEN DE REQUISICIONES PENDIENTES Y SU ANTIGUEDAD-

ANTIG-MES MES	ANT	12 SEP	11 OCT	10 NOV	9 DIC	8 ENE	7 FEB	6 MAR	5 ABR	4 MAY	3 JUN	2 JUL	1 AGO	0 SEP	REON VENC	TOTA REQN	
ESPERA CD	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4	1	5
X SOL.COT	0	0	0	1	0	0	0	0	0	0	0	0	0	2	1	3	
TOTAL	0	0	0	1	0	0	0	0	0	0	0	1	0	6	2	8	

REQN-RECIBIDAS
 MES REQ PAR

- HISTORICO DE ANTIGUEDADES -

OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	REON VENC	TOTA REQN			
13	15	0	0	0	1	0	0	0	0	0	0	1	0	6	2	8

P E D I D O S

EN EL MES	PEDS PART	MONTO	ACUMULADOS	PEDS PARTS	MONTO
PEDS NALS:	8	8	PEDS NALS:	103	108
PEDS IMPN:	2	2	PEDS IMPN:	70	76
SUBTOTAL:	10	10	SUBTOTAL:	173	184
MODF NALS:	7	7	MODF NALS:	14	14
MODF IMPN:	0	0	MODF IMPN:	4	4
SUBTOTAL:	7	7	SUBTOTAL:	18	18
TOTAL:	17	17	TOTAL:	191	202

- RESUMEN DE PEDIDOS PENDIENTES Y SU ANTIGUEDAD -

ANTIG-MES MES	ANT	13 SEP	12 OCT	11 NOV	10 DIC	9 ENE	8 FEB	7 MAR	6 ABR	5 MAY	4 JUN	3 JUL	2 AGO	1 SEP	PEDS VENC	TOTA PEDS
EN FRONTE	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2
EN TRANCI	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	3
ESPERAN H	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	3
POR EMBAR	2	0	1	0	0	0	2	0	1	2	3	1	1	4	1	10
EN ACLARA	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	5
TOTAL	2	0	1	0	0	0	2	0	1	4	4	1	3	8	26	116

PEDS-COLOCADOS
 MES PED PAR

- HISTORICO DE ANTIGUEDADES -

OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	REON VENC	TOTA REQN				
13	10	2	0	1	0	0	0	2	0	1	4	4	1	3	8	26	116

REPORTE DEL DEPTO CORP ABASTECIMIENTO

UNIDAD: TI TIZAPA													RECIBIDAS	REQNS	PARTS	CANC	
ALMACEN: TIC TIZAPA CONST													EN CL MES	4	15	0	
REQUISICIONES: 31/10/93													ACUMULADA	79	438	12	
-RESUMEN DE REQUISICIONES PENDIENTES Y SU ANTIGUEDAD-																	
ANTIG-MES	ANT	12	11	10	9	8	7	6	5	4	3	2	1	0	REQN	TOTA	ANT
MES	OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	VENC	REON	REON	MES
ESPERA CO	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	5	1
TRAMITE F	0	0	0	0	0	0	0	0	0	0	0	1	1	0	2	4	1
X SOL COT	0	0	0	0	0	0	0	0	0	0	0	1	2	1	4	4	4
TOTAL	0	0	0	0	0	0	0	0	0	0	0	3	4	4	7	11	10

REQN-RECIBIDAS - HISTORICO DE ANTIGUEDADES -

MES REQ PAR	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	4	15	0	0	0	0	0	0	0	0	0	0	0	0	0	3	4	4	7	11	10
-------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----

P E D I D O S

EN EL MES	PEDS	PART	MONTO	ACUMULADOS	PEDS	PARTS	MONTO
PEDS NALS:	7	34		PEDS NALS:	45	150	
PEDS IMPN:	4	9		PEDS IMPN:	14	39	
SUBTOTAL:	11	43		SUBTOTAL:	59	219	
MODF NALS:	4	44		MODF NALS:	11	79	
MODF IMPN:	0	0		MODF IMPN:	1	1	
SUBTOTAL:	4	44		SUBTOTAL:	12	50	
TOTAL:	15	87		TOTAL:	71	299	

- RESUMEN DE PEDIDOS PENDIENTES Y SU ANTIGUEDAD -

ANTIG-MES	ANT	13	12	11	10	9	8	7	6	5	4	3	2	1	PEDS	TOTA	ANT
MES	OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	VENC	REON	REON	MES
EN TRANCI	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3	3	1
POR ENBAR:	0	0	0	0	0	0	0	0	0	2	1	1	4	10	20	33	37
EN ACLARA	0	0	0	1	0	1	1	1	0	0	0	0	1	2	7	11	30
TOTAL	0	0	0	1	0	1	1	1	0	2	1	1	6	16	30	47	90

PEDS-COLOCADOS - HISTORICO DE ANTIGUEDADES -

MES PED PAR	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	11	43	0	0	0	1	0	1	1	1	0	2	1	1	6	16	30	47	90
-------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----	----

SERVICIOS INDUSTRIALES PEÑONES, S.A. DE C.V.
 Dirección de Logística y Abastecimiento
 Subdirección de Abastecimiento

PEDIDOS PENDIENTES A LA FECHA: 27/09/93
 Clasificados por Área y Comprador

AREA: 02 Bienes de Capital

COMPRADOR: 09 ING. SAUL JIMENEZ ARAM

ALM	PEDIDO	RUN PART	PROVEEDOR	SI	FECHA STATUS	FECHA AUTORIZA	FECHA ENBARQUE	ULT. EXP.	PRX. EXP.	DIAS ATRASO	SEM TRANS	MESES TRANS
	RAS 27697 00	001	LURGI GMBH	XE	13/05/93	13/05/93	18/11/93			0	0	0
	QR 04690 00	011	ABB ENERGIE	XE	16/08/93	16/08/93	19/11/93			0	0	0
	MS 03231 00	009	SULZER ESCHER WYSS	XE	24/05/93	24/05/93	10/12/93			0	0	0
	AFT 06310 00	008	ABB IND.UMD AUTOMATION AKT	XE	06/07/93	06/07/93	17/12/93			0	0	0
	AFT 06310 01	003	ABB EQUIPOS Y SISTEMAS, S.	XE	06/07/93	06/07/93	17/12/93			0	0	0
	QR 04691 00	009	SULZER ESCHER WYSS	XE	03/09/93	03/09/93	30/12/93			0	0	0
	QR 03724 00	002	A B B E M E R G Y H	XE	27/07/93	27/07/93	21/02/94			0	0	0
	RAS 32074 00	002	ANTIEMGESELLSCHAFT KUMLE	XE	08/09/93	08/09/93	05/03/94			0	0	0
	MS 04119 00	007	SIEMENS, AG	XE	09/09/93	09/09/93	08/03/94			0	0	0

PEDIDOS SIN RETRASO 31
 PEDIDOS RETRASADOS 8
 TOTAL DE PEDIDOS 39
 DIAS DE RETRAS 251
 DIAS DE RETRAS 251
 PROMEDIO DIAS POR PEDIDO 31
 PROMEDIO DIAS POR PEDIDO 6

STATUS	ROUTOS	C/RET	S/RET	DIAS	DIAS/RET	DIAS/TOT
XE	39	8	31	251	31	6
EF/ET	2	2	0	171	86	86
RR	3	3	0	478	96	96
OTROS	3	1	2	106	106	33

PEDIDOS	PARTIDAS	MESES	PUNTOS
1	4	10	10
1	1	8	8
1	7	7	7
1	1	6	6
2	3	5	10
3	3	4	12
7	42	3	21
7	30	2	14
3	4	1	3
23	148	0	0
49	245	46	91

REPORTE MENSUAL DEL DEPTO. CORP. DE ABASTECIMIENTO

PEDIDOS CERRADOS

en el periodo comprendido del 13/09/93 al 18/09/93

Clasificados por Area y Comprador

AREA: 02 Bienes de Capital

COMPRADOR: 08 Alma Rosa Rodriguez Caruante

ALM	FEDIDO	MON PARTS	FECHA ENTREGA	PROVEEDOR	DESCRIPCION GENERAL	FECHA CIERRE	FECHA ENTREGA	DIAS RET	SIN RET	MES RET		
FRE	11823	1	4	19/07/93	MYX, S. A. DE C. V.	SOFTWARE	17/09/93	18/06/93	0	0	0	
FRE	12625	0	2	04/08/93	INFORMATICA TIRON, S.A. DE C	IMPRESORA HP	17/09/93	04/08/93	0	0	1	
LCC	51	1	1	30/07/93	COMVALTECH, INC.	ANALIZADORES DE PH	17/09/93	09/08/93	10	1	2	
RO	4533	0	1	14/09/93	HUMET, STA CRUZ Y ASOC., S.A. D	DISCO DURO DE 670 MB.	17/09/93	25/08/93	0	0	0	
RO	4620	1	1	16/07/93	JAR. ELECTRONICA APLICADA, SA	EQUIPO POWER LOGIC.	17/09/93	04/09/93	50	6	3	
MA	3044	6	1	16/07/93	IDEAS COMPUTACIONALES DE MEX	EQUIPO DE COMPUTO	17/09/93	05/08/93	20	2	2	
NS	3616	0	1	10/09/93	FORMY INTERNATIONAL INC.	REPARACION DE TABLILLA	17/09/93	13/08/93	0	0	0	
OM	52751	0	1	25/06/93	HERCYKIA, S.A.	ACONDICIONADOR MAGNETICO DE	17/09/93	26/06/93	1	0	1	
RM	12111	0	2	09/06/93	M. CANTU GLEZ V/O TECNOCONTROL	EQUIPO DE CODIFICACION	17/09/93	10/06/93	62	8	3	
SE	7	0	4	01/06/93	INFORMATICA TIRON, S.A. DE C	Computadoras y Equipo	17/09/93	17/08/93	77	10	3	
SE	8	0	2	01/06/93	COMPUTADORAS PORTATILES S.A.	EQUIPO DE COMPUTO	17/09/93	18/08/93	78	10	3	
SE	8	1	2	01/06/93	INFORMATICA TIRON, S.A. DE C	EQUIPO DE COMPUTO	17/09/93	18/08/93	78	10	3	
SE	42	0	1	14/09/93	INTIEGRA, S.A. DE C.V.	UNIDAD DE RESPALDO EN CINTA	17/09/93	13/09/93	0	0	1	
SE	44	0	1	14/09/93	INTIEGRA, S.A. DE C.V.	UNIDAD DE RESPALDO, NCA.	17/09/93	13/09/93	0	0	1	
SP	5454	0	3	06/09/93	INFORMATICA TIRON, S.A. DE C	VECTRA E IMPRESORA	17/09/93	17/09/93	11	0	1	
SP	5467	0	2	15/09/93	INFORMATICA TIRON, S.A. DE C	VECTRA HP	17/09/93	13/09/93	0	0	1	
TIC	51	0	1	30/07/93	COMVALTECH, INC.	ANALIZADORES DE PH	17/09/93	09/08/93	10	1	2	
TOR	2873	0	2	19/07/93	COMURC. Y DESARR. COMPUTAC. SA	Equipo de Comunicacion	17/09/93	03/09/93	46	5	3	
TOTAL PEDIDOS RETRASADOS:			11				PROMEDIO DIAS POR RETRASADOS:			40		
TOTAL SIN RETRASO:			7				PROMEDIO DIAS POR NOVIMIENTO:			25		
TOTAL DE NOVIMIENTOS:			18	TOTAL DE DIAS:			443					

FECHA DE PROCESO: 24/09/93

ASPECTO: DI_1

PEÑÓLES

HOJA: 7

SERVICIOS INDUSTRIALES PEÑÓLES, S.A. DE C.V.
 Dirección de Logística y Abastecimiento
 Subdirección de Abastecimiento

REPORTE MENSUAL DEL DEPTO. CORP. DE ABASTECIMIENTO
 PEDIDOS CERRADOS
 en el periodo comprendido del 13/09/93 al 18/09/93
 Clasificados por Área y Comprador

COMPRADOR	CUMPLIDOS	NO CUMPLIDOS	DIAS RETRASO	PROMEDIO DIAS NO CUMPLIDOS	PROMEDIO GRAL.
04	0	6	342	57	57
05	0	3	145	48	48
06	1	13	632	49	45
08	7	11	443	40	25
SUBTOTAL	8	33	1,562	47	38
12	0	2	74	37	37
15	1	4	173	43	35
SUBTOTAL	1	6	247	41	35
TOTALES	9	39	1,809	46	38

FECHA DE PROCESO: 23/09/93
 25-09-93 1

HOJA: 9

SERVICIOS INDUSTRIALES PEÑOLÉS, S.A. DE C.V.
 Dirección de Logística y Abastecimiento
 Subdirección de Abastecimiento

PEÑOLÉS

REPORTE MENSUAL DEL DEPTO. CORP. DE ABASTECIMIENTO
 REQUISICIONES RECIBIDAS
 en el período comprendido del 13/09/93 al 18/09/93
 Clasificadas por Almacén, Unidad, Compañía y División

DIVISION: 3 METALES

COMPAÑIA: 3 MET-MEX PEÑOLÉS, S.A. DE C.

ALM	FOLIO	NUM PARTS	FECHA RECIB	DESCRIPCION GENERAL	ST	FECHA STATUS	DIAS TRANS	SER TRANS	RESER TRANS
AFT	6769	1	13/09/93	MOTORREDUCTOR MARCA JIV	EC	17/09/93	8	0	0
AFT	6770	1	13/09/93	MOTORREDUCTOR MARCA FALK	EC	17/09/93	8	0	0
AFT	6833	4	14/09/93	UNIDAD ENFRIAMIENTO HCA, UMITHERMICS	SC	14/09/93	9	0	0
TOTAL DE MOVIMIENTOS:				3					
TOTAL DIAS:				25	PROMEDIO DIAS/MOVIMIENTO: 8				
RAS	34614	1	13/09/93	Fundición y Maquinados	EC	17/09/93	8	0	0
RAS	35017	1	13/09/93	RIEL PERFIL S-10	SC	13/09/93	8	0	0
TOTAL DE MOVIMIENTOS:				2					
TOTAL DIAS:				16	PROMEDIO DIAS/MOVIMIENTO: 8				
RTR	42497	1	14/09/93	PAILA DE 250 TONELADAS DE CAPACIDAD	SC	14/09/93	9	0	0
TOTAL DE MOVIMIENTOS:				1					
TOTAL DIAS:				9	PROMEDIO DIAS/MOVIMIENTO: 9				
TR	50241	1	14/09/93	BOLSAS FILTRO ESTILO P-595-2	SC	14/09/93	9	0	0
TR	50256	1	14/09/93	REFACC. PARA BOMBA RHEINGUTE	EC	17/09/93	9	0	0
TR	50257	1	14/09/93	REFACC. PARA BOMBA RHEINGUTE	EC	17/09/93	9	0	0
TR	50258	1	14/09/93	REFACC. PARA BOMBA RHEINGUTE	EC	17/09/93	9	0	0
TR	50264	1	14/09/93	AMPLIFICADOR TIPO ZOM5	SC	14/09/93	9	0	0
TR	50313	1	13/09/93	RAMALES DE CADENA	SC	13/09/93	8	0	0
TR	50339	1	13/09/93	CONTACTO TRAPEZOIDAL PARA SOLERA	EC	17/09/93	8	0	0
TR	50366	1	13/09/93	TELA EMVIRIOICH	SC	13/09/93	8	0	0
TR	50367	1	13/09/93	TELAS FILTRANTES TIPO R-79	SC	13/09/93	8	0	0
TR	50377	1	13/09/93	REFACC. PARA BOMBA WERBERI	EC	17/09/93	8	0	0
TR	99926	1	14/09/93	REFACC. PARA TURBINA DE VAPOR NKK	SC	14/09/93	9	0	0
TR	99927	1	14/09/93	REFACC. PARA TURBINA DE VAPOR NKK	SC	14/09/93	9	0	0
TOTAL DE MOVIMIENTOS:				12					
TOTAL DIAS:				103	PROMEDIO DIAS/MOVIMIENTO: 9				
TOTAL DE MOVIMIENTOS/UNIDAD:				18					
TOTAL DIAS:				151	PROMEDIO DIAS/MOVIMIENTO: 9				
TOTAL DE MOVIMIENTOS/DIVISION:				18					
TOTAL DIAS:				153	PROMEDIO DIAS/MOVIMIENTO: 9				

SERVICIOS INDUSTRIALES PEÑOLAS, S.A. DE C.V.
 Dirección de Logística y Abastecimiento
 Subdirección de Abastecimiento

REPORTE MENSUAL DEL DEPTO. CORP. DE ABASTECIMIENTO
 REQUISICIONES RECIBIDAS
 en el período comprendido del 13/09/93 al 18/09/93
 Clasificadas por Almacén, Unidad, Compañía y División

	REQUISICIONES RECIBIDAS	DIAS	PROMEDIO DIAS POR MOVTO
Divisi n: 1	9	72	8
Compañía: 2	9	72	8
Almacén: SC	2	16	8
Almacén: SP	7	54	8
Divisi n: 2	9	76	8
Compañía: 7	1	9	9
Compañía: 39	1	9	9
Compañía: 55	2	16	8
Compañía: 71	1	9	9
Compañía: 73	2	15	8
Compañía: 76	1	9	9
Compañía: 95	1	9	9
Divisi n: 3	18	153	9
Compañía: 3	18	153	9
Almacén: AFT	3	25	8
Almacén: RNS	2	16	8
Almacén: STR	1	9	9
Almacén: TR	12	103	9
Divisi n: 4	6	54	9
Compañía: 4	3	26	9
Compañía: 88	1	10	10
Compañía: 96	2	18	9
Divisi n: 5	4	40	10
Compañía: 33	1	10	10
Compañía: 92	3	30	10
TOTALES	46	395	9

PEÑOLES

SERVICIOS INDUSTRIALES PEÑOLES, S.A DE C.V.
 Dirección de Logística y Abastecimiento
 Subdirección de Abastecimiento

REPORTE MENSUAL DEL DEPTO. CORP. DE ABASTECIMIENTO
 REQUISICIONES CANCELADAS
 en el período comprendido del 01/09/93 al 30/09/93
 Clasificadas por Almacén, Unidad, Compañía y División

DIVISION: 2 MINAS

COMPAÑIA: 04 SERVICIOS DE EXPLORACION, SA

REQUISICION	NUM PARTS	FECHA RECIB	DESCRIPCION GENERICA	SI	FECHA CANCEL	DIAS TRANS	SEN TRANS	MESES TRANS
UNIDAD: LC PROV. LA CIENEGA								
08-LCC-00229	8	14/06/93	INDICADOR DE NIVEL TIPO ELECTRODO	RC	16/09/93	88	11	3
09-LCC-00273	1	14/06/93	CORNETA-ALARMA	RC	10/09/93	88	11	3
05-LCC-00280	22	13/06/93	EQUIPOS DE LABORATORIO P/PROYECTO LA CIENEGA	RC	09/09/93	86	11	3
05-LCC-00291	1	29/06/93	EQUIPO DE LABORATORIO PARA PROYECTO LA CIENEGA	RC	09/09/93	72	9	3
05-LCC-00293	5	29/06/93	EQUIPO DE LABORATORIO PARA PROYECTO LA CIENEGA	RC	09/09/93	72	9	3
05-LCC-00297	0	29/06/93	EQUIPO DE LABORATORIO PARA PROYECTO LA CIENEGA	RC	09/09/93	72	9	3
05-LCC-00287	1	21/07/93	CILINDROS NEUMATICOS	RC	10/09/93	51	6	2
TOTAL DE MOVIMIENTOS:		7						
TOTAL DIAS:		529						76
UNIDAD: TI TIZAPA								
06-TI-00635	4	19/07/93	CABLE TRIFASICO VARIOS CALIBRES	RC	10/09/93	53	6	2
06-TI-00636	2	19/07/93	CABLE DE COBRE	RC	10/09/93	53	6	2
06-TI-00637	1	19/07/93	CENTRO DE CONTROL DE MOTORES	RC	10/09/93	53	6	2
TOTAL DE MOVIMIENTOS:		3						
TOTAL DIAS:		159						53
UNIDAD: TIC TIZAPA								
08-TIC-00050	3	25/05/93	INSTUMENTACION PROYECTO TIZAPA	RC	23/09/93	121	16	4
05-TIC-00083	22	13/06/93	EQUIPOS DE LABORATORIO P/PROYECTO TIZAPA	RC	09/09/93	86	11	3
04-TIC-00057	2	22/06/93	TANQUES DE ALMACENAM. PARA DIESEL Y GAS.	RC	23/09/93	93	12	3
05-TIC-00284	7	29/06/93	EQUIPO DE LABORATORIO PARA PROYECTO TIZAPA	RC	09/09/93	72	9	3
05-TIC-00093	1	29/06/93	EQUIPO DE LABORATORIO PARA PROYECTO TIZAPA	RC	09/09/93	72	9	3
05-TIC-00095	5	29/06/93	EQUIPO DE LABORATORIO PARA PROYECTO TIZAPA	RC	09/09/93	72	9	3
TOTAL DE MOVIMIENTOS:		6						
TOTAL DIAS:		516						86
UNIDAD: TIR TER								
08-TIR-00007	1	20/05/93	DIGITALIZADOR PARA GRABAR LOGO TIPOS	RC	24/09/93	127	17	4
TOTAL DE MOVIMIENTOS:		1						
TOTAL DIAS:		127						127
UNIDAD: MA MAICA								
08-MA-00383	2	20/07/93	AMALIZADOR DE PARTICULAS MARCA AUTOMUMPU	RC	23/09/93	65	8	2
TOTAL DE MOVIMIENTOS:		1						
TOTAL DIAS:		65						65
UNIDAD: PET UNIDAD PEREGRINA								
07-FE-01078	1	20/07/93	REFACC. PARA SCOOP TRAN JARVIS CLARK	RC	22/09/93	64	8	2
TOTAL DE MOVIMIENTOS:		1						
TOTAL DIAS:		64						64
UNIDAD: SE SERV. EXPL.								
08-SE-00040	1	08/07/93	INVERSOR DE CORRIENTE ELECTRICA MARCA BEST	RC	17/09/93	71	9	2
TOTAL DE MOVIMIENTOS:		1						
TOTAL DIAS:		71						71

SERVICIOS INDUSTRIALES PEÑALES, S.A DE C.V.
 Dirección de Logística y Abastecimiento
 Subdirección de Abastecimiento

REPORTE MENSUAL DEL DEPTO. CORP. DE ABASTECIMIENTO
 REQUISICIONES CANCELADAS
 en el período comprendido del 01/09/93 al 30/09/93
 Clasificadas por Almacén, Unidad, Compañía y División

	REQUISICIONES CANCELADAS	DÍAS	PROMEDIO DÍAS POR RÓTULO
Divisi n: 1	3	166	55
Compañía: 2	3	166	55
Unidad: IT	1	35	35
Unidad: SP	2	131	66
Divisi n: 2	29	1,804	62
Compañía: 7	7	529	76
Compañía: 39	9	675	75
Almacén: IT	3	159	53
Almacén: TIC	6	516	86
Compañía: 53	1	127	127
Compañía: 71	1	65	65
Compañía: 76	1	64	64
Compañía: 84	1	71	71
Compañía: 95	9	273	30
Divisi n: 3	3	63	21
Compañía: 3	3	63	21
Almacén: AFT	2	42	21
Almacén: TR	1	21	21
Divisi n: 5	1	10	10
Compañía: 33	1	10	10
TOTALES	36	2,043	57



SERVICIOS INDUSTRIALES PEÑONES, S.A. DE C.V.
 Dirección Logística y Abastecimiento
 Sub-Dirección de Abastecimiento

REQUISICIONES Y PEDIDOS EN PROCESO DE CLASIFICACION
 POR DIU-CORP EN LA OFICINA: México, D. F.

Bienes de Capital

Sep 28 1993

Co	Len.	Cons.	An	Description	Proveedor	Fec.	Entrega	Status	Fec.	Status
09	AFT	4616	0	VACUUM DISTILLATION SYSTEM FOR ZINCSEPAR	ABB METALLURGY INC.	Dec	18 1992	NR	Jan	5 1993
		5740	0	TIPO B MATERIAS PARA REPOSICION TORRE	CUSTODIS ECODEVIE INC.	May	12 1993	NR	Apr	4 1993
		5865	0	HANO DE OBRA PARA LA REPARACION DELMA	MVR TORRES DE ENFRIAMIENTO, SA CU	Aug	17 1993	XE	Apr	7 1992
		6125	0	CORPRA DE SOPORTE PARA CARCA MEGATINA	IRD MECHANALYSIS, INC.	Jul	7 1993	XE	May	27 1992
		6146	0	VENTILADOR CON MOTOR P/TORRE DEEMFRANIE	CUSTODIS ECODEVIE INC.	Jul	12 1993	XE	May	17 1992
		6206	0	MADQUINA MOLDEADORA DE LINGOTES DE ZINC	NETPRO MACHINERY LTD.	Nov	19 1993	PA	Jun	26 1992
		6310	0	MORHOS ELECTRICOS DE INDUCCION	ABB IND.UMD AUTOCATION ANTIEMGESELLSCHAFT	Dec	17 1993	XE	Jul	6 1993
					ABB EQUIPOS Y SISTEMAS, S.A. DE C.V.	Dec	17 1993	EC	Jul	1 1993
		6412	0	MORHOS ELECTRICOS				EC	Aug	11 1993
		6541	0	Torres de Enframiento				EC	Sep	24 1992
		6567	0	ASISTENCIA TECNICA MORHO DE INDUCCION Mo	ABB METALLURGY INC.	Aug	15 1993	XE	Sep	24 1992
		6610	0	ASISTENCIA TECNICA	UMTEC SYSTEMS OV	Oct	25 1993	PA	Aug	14 1992
		6647	0	MADQUINA CORTADORA DE LAMINA	DOXA, S. A. DE C.V.	Sep	29 1993	XE	Sep	23 1993
		6671	0	Torres de Enframiento				EC	Aug	24 1993
		6833	0	UNIDAD ENFRIAMIENTO MCA, UNITHERMICS				SC	Sep	14 1993
		6887	0	CAMBIADOR DE CALOR				SC	Sep	21 1993
BR		2949	0	MALLA RAMURADA P/CENTRIFUGA PUSHER(GROUP	KRAUSS MAFFEI CORP	Aug	20 1993	XE	May	28 1992
FRE		10910	0	SISTEMA DE CONTROL DE POLVOS VIASUPRESIO	SUPERKON, S.A. DE C.V.	Feb	22 1993	NR	Mar	25 1993
		11128	0	Riel y accesorios	L.B. FOSTER COMPANY	Jul	23 1993	XE	May	20 1993
					L.B. FOSTER COMPANY	Jul	23 1993	XE	May	28 1993
								EC	Aug	19 1993
FY		103	0	REFACC. PARA CENTRIFUGA				EC	Sep	17 1993
		1407	0	REFACC. PARA CENTRIFUGA MARCA ESCHER WYS				EC	Sep	17 1993
LC		4091	0	SISTEMA DE CAMBIO DE VIAS	L.B. FOSTER COMPANY	Jun	24 1993	XE	May	20 1992
		4113	0	Riel y accesorios	L.B. FOSTER COMPANY	Jun	17 1993	IF	Sep	16 1993
		4157	0	SISTEMA DE EQUIPO PNEUMATICO P/ABRIR-CERR	TESTO PNEUMATIC, S.A.	Aug	12 1993	XE	Aug	17 1993
		4773	0	Riel y accesorios	L.B. FOSTER COMPANY	Oct	22 1993	XE	Sep	24 1992
LCC		48	0	COLECTORES DE POLVO (DUST COLLECTORS)	AVANTE INGENIEROS, S.A. DE C.V.	Aug	27 1993	XE	May	21 1993
		153	0	VALVULA ROTATORIA PARA PRECIPITADOR	HAZEMAC, S.A. DE C.V.	Aug	26 1993	XE	Jul	8 1993
MS		3231	0	REFACC. PARA TURBINA DE GAS SULZERIZO 7	SULZER ESCHER WYSS	Dec	10 1993	XE	May	24 1992
		3404	0	Riel y accesorios	L.B. FOSTER COMPANY	Jul	15 1993	XE	May	21 1992
		3786	0	REFACCIONES PARA TURBINA A GAS/SULZER IIP	SULZER ESCHER WYSS	Jul	19 1993	NR	Jul	19 1993
		4117	0	REFACC. PARA TURBINA SIEMENS	SIEMENS, AG	Mar	9 1994	XE	Sep	4 1992
		4222	0	SERVICIOS TECNICOS A TURBOGENERADOR SIEH				SC	Sep	20 1993
		10556	0	CALANDRIA CAMBIADOR DE CALOR				EC	Jun	30 1993
		10595	0	INTERCAMBIADOR DE CALOR	ALFA LAVAL SHARPLES, S.A. DE C.V.	Sep	20 1993	XE	Jul	27 1993
OM		53279	0	REFACC. PARA MORHO MAERZ	WILHELM KRIEHL MASCHINENBAU	May	12 1993	XE	Aug	23 1993
OR		2621	0	REFACCIONES PARA TURBINA DE GAS ACASULZ	SULZER ESCHER WYSS	Jul	29 1993	NR	Aug	12 1993
		3483	0	REFACC. P/TURBO ACEC TIPO 21 -PDDDOMEST	ABB EMERGIE	Oct	25 1993	XE	May	17 1992
		3648	0	REFACC. PARA TURBINA ACEC TIPO MO.21.	A B B E M E R G Y H V	Oct	26 1993	XE	Jul	21 1992
		3655	0	REFACC. PARA TURBINA SIEMENS	SIEMENS A.C.	Nov	8 1993	XE	Jun	11 1993
		3698	0	REFACCIONES PARA TURBINA ACEC TIPO MO. 2	A B B E M E R G Y H V	Oct	26 1993	XE	Jul	22 1992
		3724	0	REFACC. PARA TURBOGENERADOR ACEC TIPO MO.	A B B E M E R G Y H V	Feb	21 1994	XE	Jul	27 1993
		3820	0	REFACCIONES PARA TURBINA ACEC TIPO MO. 2	A B B E M E R G Y H V	Oct	26 1993	XE	Jul	21 1992
		3930	0	REFACC. PARA TURBINA SULZERREFACC. PARA	SULZER ESCHER WYSS	Nov	2 1993	XE	Jun	12 1992
		4062	0	REFACCIONES PARA TURBINA ACEC TIPO MO. 2	A B B E M E R G Y H V	Oct	26 1993	XE	Jul	21 1992
		4690	0	REFACC. PARA TURBINA ACEC	ABB EMERGIE	Nov	19 1993	XE	Aug	16 1992
		4691	0	REFACC. PARA TURBINA SULZERISAPARE PARTS	SULZER ESCHER WYSS	Dec	30 1993	XE	Sep	3 1992
		4692	0	REFACC. PARA TURBINA SIEMENS				EC	Jul	19 1992
RAS		27697	0	FABRICACION DE TOBERAS	LURCI GMBH	Nov	18 1993	XE	May	13 1992
		27705	0	SPARE PARTS FOR ELECTROSTATIC PRECIPITAT	LURCI GMBH	Mar	26 1993	IF	Nov	1993
		29281	0	PAILA DEZINCADORA DE 240 TONELADAS	JOSEPH ADAMSON (EUROPE) LTD.	Ray	7 1993	XE	Jan	29 1993
		31476	0	REFACCIONES PARA PRECIPITADORES ELECTROS	LURCI EMERGIE UMD UEGELT GMBH	Oct	27 1993	XE	Aug	20 1993
		32074	0	REFACC. PARA TURBINA DE VAPOR BOB AT 3	ANTIMIEGSELSCHAFT KUMLE KOPPEL/KUSCH	Mar	5 1994	XE	Sep	8 1993
		32231	0	SPARE PARTS FOR TOASTING ROTARY COOLERSI	DEUTSCHE BABCOCK ANDLIGH A.C.	Sep	23 1993	XE	May	27 1992
		34983	0					EC	Sep	17 1993
		35017	0	RIEL PERFIL 5-10				EC	Sep	22 1993
RN		12772	0	REFACC. PARA MORHO METZSCH MOD. 422	METZSCH GERATEBAU GMBH	Aug	25 1993	XE	Aug	12 1992
		13407	0	REFACC. PARA RECTIFICADORA BLOHM HFS-512	FERROSTAL, A.C.	Oct	14 1993	XE	Sep	27 1992
		13844	0	REFACC. PARA RECTIFICADORA BLOHM HFS-512				EC	Sep	27 1992
		13883	0	REFACC. PARA RECTIFICADORA BLOHM HANSEAT				EC	Sep	27 1992
RTR		41067	0	PAILAS DE 250 TONS. PARA FUNDIR PLONO RE	JOSEPH ADAMSON (EUROPE) LTD.	Apr	16 1993	RB	Sep	24 1993
		41927	0	PAILA DE 250 TONS. CAPACIDAD				EC	May	11 1992
		42228	0	REFACC. PARA MOLDEADORA SHEPPARD 200ISA	NETPRO MACHINERY LTD.	Oct	10 1993	XE	Aug	12 1993
		42497	0	PAILA DE 250 TONELADAS DE CAPACIDAD				EC	Sep	22 1993
SC		76004	0	SC - TR - CARRO DE PELITIZACION COMPLETO				EC	Jul	7 1992
		76005	0	SC - OR - REFACC. PARA MORHO MAERZ				EC	Aug	11 1993
		80802	0	LINIA AUTOMATICA DE ESTIBADO DE LINGOTES				EC	Sep	17 1993
		89003	0	SC - TR - COLECTOR DE POLVO				EC	Aug	16 1993
TER		6	0	MADQUINA DE PULIDO	DIAMANTEK, S.A. DE C.V.	Jun	7 1993	XE	Jun	10 1993
TR		99926	0	REFACC. PARA TURBINA DE VAPOR KKK				SC	Sep	14 1993
		99927	0	REFACC. PARA TURBINA DE VAPOR KKK				SC	Sep	14 1993

ALVARO PEGOLES
 Sr. ALBERTO ENRIQUEZ

REPORTE DE REQUISICIONES Y PEDIDOS EN PROCESO DE CLASIFICACION POR ALMACEN EN LA OFICINA REGIONAL: México, D. F.

CU ALM.	FOLIO	AM.F.RECEP.	DESCRIPCION	PROVEEDOR	F.ENTREG	SI	F.STAT
04	TR	92927	0 May 6 1993 REFACC. P/BOMBA WERMERT	WERMERT PUMPEB GRBH	Jun 25 1993	ET	Jun 25 1993
06	TR	98037	0 May 17 1993	CECELEC ACEC, S.A.	Sep 23 1993	MR	Sep 23 1993
06	TR	76076	0 May 17 1993 DIODO ACEC SR-360/12	CECELEC ACEC, S.A.	Jun 3 1993	MR	Jun 3 1993
06	TR	98214	0 May 25 1993 DIODO ACEC SR-380/12	CECELEC ACEC, S.A.	Oct 16 1993	XI	Jun 10 1993
05	TR	98320	0 May 25 1993 RAMALES DE CADENA DE 0.96 MTS.	Band TKV TRANSPORTMAGLAGE KONSTRUKTION UND W	Nov 4 1993	XE	Jun 7 1993
06	TR	98409	0 May 28 1993	MARECHAL	Aug 9 1993	XE	Jun 25 1993
06	TR	90767	0 Jun 25 1993 CONTACTOS PARA INTERRUPTOS ALTA TENSIÓN	SIERENS, AG	Oct 22 1993	XI	Jul 13 1993
08	TR	98691	0 Jul 6 1993 REFACCIONES PARA TERMOMETRO	LURCI GRBH	Dec 20 1993	PA	Sep 23 1993
04	TR	99035	0 Jul 6 1993 REFACC. PARA BASCULAS PFISTER	PFISTER GRBH	Sep 23 1993	XE	Aug 16 1993
06	TR	99048	0 Jul 6 1993 RELEVADOR MARCA FRABA	FABRIK ELEKTR APPARATE GRBH	Oct 1 1993	XE	Jul 30 1993
08	TR	99067	0 Aug 12 1993 MANOMETRO	LURCI GRBH	Dec 20 1993	PA	Jan 23 1993
04	TR	99117	0 Jul 13 1993 TUBO ACERO INOXIDABLE 316	TUBE AMERICA, INC.	Jul 26 1993	MR	Jun 6 1993
15	TR	99109	0 Jul 16 1993 CONTACTO TRAPEZOIDAL PARA SOLERA	THE ELECTRIC MATERIALS CO.	Aug 27 1993	EF	Sep 21 1993
04	TR	99268	0 Jul 28 1993 TUBO DE ACERO INOX. 316 L, INCLUYE RIQ	TUBE AMERICA, INC.	Aug 23 1993	EF	Sep 21 1993
08	TR	99312	0 Jul 23 1993 TERMOPOZO DE 3/4" X 18"	LURCI GRBH	Oct 13 1993	XE	Jul 30 1993
04	TR	99447	0 Aug 3 1993 REFACC. PARA BOMBA VERTICAL RHEINHUTTE.	RHEINHUTTE VORN LUDWIGSDECH + CO.	Sep 23 1993	MR	Sep 23 1993
03	TR	99631	0 Aug 12 1993 FUSIBLES SEMICONDUCTORES 1750 ANP.		EC	Sep 21 1993	
15	TR	99639	0 Aug 12 1993 PLACAS BINETAlicas	E. I. DUPONT DE NEMOURS & CO. INC	Sep 2 1993	EF	Sep 24 1993
01	TR	99704	Aug 17 1993 INTERRUPTOR		EC	Sep 24 1993	
05	TR	99733	0 Aug 17 1993 TELA ALBANY ESTILO N-3005	TEXTIL FILTER, S.A. DE C.V.	Sep 17 1993	XE	Aug 23 1993
07	TR	99772	Aug 17 1993 PARTES PARA		EC	Aug 19 1993	
03	TR	99775	0 Aug 20 1993 LOMAS FILTRANTES	TEXTIL FILTER, S.A. DE C.V.	Oct 3 1993	XE	Sep 1 1993
04	TR	99882	Aug 24 1993 REFACC. PARA BOMBA RHEINHUTTE		EC	Aug 26 1993	
07	TR	99901	Aug 26 1993 AMILLO DE ENFRIAMIENTO P/BOMBA DE VACIO		EC	Aug 30 1993	
07	TR	99902	Aug 26 1993 CARISA PARA BOMBA DE VACIO SIERENS ELRO		EC	Aug 30 1993	
09	TR	99926	Sep 14 1993 REFACC. PARA TURBINA DE VAPOR KKK		SC	Sep 12 1993	
09	TR	99927	Sep 14 1993 REFACC. PARA TURBINA DE VAPOR KKK		SC	Sep 14 1993	

PA POR AUTORIZAR	003
XE PENDIENTE DE ENTRICAR	048
ET EN TRANSITO	001
EF EN FRONTERA	003
MR ESPERANDO RECIBO EN ALMACEN	008
RB RECIBIDO	001
SC SOLICITANDO COTIZACION	007
EC ESPERANDO COTIZACION	028
TOTAL ALMACEN	099



SERVICIOS INDUSTRIALES PEÑOLES S.A. DE C.V.

DEPARTAMENTO DE SISTEMAS LOGISTICA Y ABASTECIMIENTOS

REPORTE DE FALLA Y/O NUEVOS REQUERIMIENTOS

DIRIGIDO A: ING. ARTURO ENRIQUEZ

SOLICITANTE ING. GUILLERMO HUERTA, EXT. 3056

LOCALIDAD: MEXICO, D. F.

FECHA: NOV. 11, 1993.

DEPARTAMENTO: ABASTECIMIENTO

FOLIO: ABA-205

REPORTE DE FALLA
DE PROGRAMA.

MODULO: _____

FUNCION: _____

PROGRAMA: _____

NUEVO REQUERIMIENTO

DESCRIPCION: CON FECHA NOV. 10 A LAS 21 HRS. DE LA REQUISICION No. 06FRE 12818 SE EFECTUO LA CANCELACION DE LAS PARTIDAS 6 Y 13, OBTENIENDOSE CERO CAMBIO EN EL ESTATUS DE DICHAS PARTIDAS, DADO LO ANTERIOR LES SOLICITAMOS SEA REVISADO ESTE PROGRAMA YA QUE ATRASA NUESTROS MOVIMIENTOS. ESTE MOVIMIENTO FUE REALIZADO POR EL ING. O. MARTINEZ.

GERENTE

ING. O. MARTINEZ

AUTORIZACION

SOLICITANTE

ING. G. HUERTA

NOMBRE

SISTEMAS

RECIBIO: (NOMBRE / FECHA) GABRIEL CASSIO N.
11/NOV/93

COMENTARIOS DE SISTEMAS: _____

VERIFICO

ELABORO

APROBACION USUARIO

Vo. Bo. / FECHA

NOMBRE / FECHA

NOMBRE / FECHA

FLA-01-1090

-320- 11/Nov/93

APENDICE F.

**MANUAL DE USUARIO
SISTEMA DE ABASTECIMIENTO
UNIX/SYBASE**



INTRODUCCION

El objetivo del presente documento es brindar al usuario los pasos a seguir para la realización del Proceso de Abastecimiento en el Nuevo Sistema Informático.

El Sistema comprende tres grandes áreas Compras, Tráfico, e Infraestructura, desde las cuales podremos acceder todas las funciones para llevar a cabo el mencionado Proceso Administrativo.

Al proporcionar mediante este compendio las Pantallas Originales del Sistema con explicaciones detalladas tanto de los campos plasmados en cada una de ellas, como de la información resultante, se pretende lograr constituir gráfica y conceptualmente un documento de consulta y referencia.

Esperando cumplir con el objetivo antes mencionado agradecemos su atención.

**GERENCIA DE INFORMATICA
DIRECCION LOGISTICA Y ABASTECIMIENTO**

**TECLA [F1]**

[Compras]

[Botón de Función del cual dependen cinco submódulos de trabajo del Nuevo Sistema de Abastecimiento]

[Act./Cons]

[F1]

[Botón que permite acceder siete subfunciones por medio de las cuales se completará nuestro Proceso de Abastecimiento [Requisición, Solicitud de Cotización, Cotización, Pedido Colocar, Pedido Modificar, Movimiento Importación y Movimiento Exportación] que se explicarán en el siguiente Capítulo]

[Consulta]

[Por medio de este botón visualizaremos otras seis posibilidades de trabajo dentro de nuestro Nuevo Sistema Busqueda de Requisición, Busqueda de Pedido, Cotizaciones, Movimientos, Estadístico Requisición, Estadístico Pedido; y de esta manera podremos acceder todos los datos existentes en Sistema a través de captura de ciertos parámetros que se utilizarán como criterios de búsqueda y que se requerirán por medio de Cuadros de diálogo]

[Catálogos]

[En esta opción se le proporciona al usuario la posibilidad de impresión de los Catálogos de Información, que hasta el momento son Catálogo de Domiclios y Catálogo de Relación Almacén Domicilio]

[Reporte Pedidos]

[Posibilidad para el usuario de impresión de los Reportes preestablecidos con respecto a las mismas necesidades del usuario, correspondientes al Area de Pedidos y son; Lista de Precios, Pedidos Autorizados Diarios, Pedidos a Expeditar a la fecha, Pedido por Colocar o Cancelar Ordenado por Area y Comprador, Pedido por Colocar o Cancelar Ordenado por División y Remisión de Envío y Modificaciones]



[Reportes Requis]

[Posibilidad de impresión de los reportes correspondientes a el Módulo de Requisiciones con formato establecido como Requisiciones Colocadas por Area y Comprador, Requisiciones Colocadas por División y Cía., Requisiciones Recibidas por línea y Material, Requisiciones Pendientes de Colocar por Area y Comprador, Requisiciones Pendientes de Colocar por División y Cía., Tabla Comparativa de Requisición; los cuales se ejemplificarán en el Tercer Módulo de este Manual]

TECLA [F2]

[Tráfico]

[Sección en desarrollo]

TECLA [F3]

[Infraestructura]

[Tecla de función que le dará posibilidad al usuario de insertar información necesaria permanentemente en los archivos del Sistema de Abastecimiento; desde luego esa inserción debe ser por áreas Proveedores, Requisitores, Compradores, Materiales y Otros a manera de conservar el orden de los archivos y sea igualmente de eficiente el Seguimiento del Proceso del Sistema; la explicación de esta sección del Manual se encuentra el Modulo Número Cuatro].

TECLA [F8]

[/]

[Tecla que ofrece la posibilidad de salir del Sistema en el Momento que el usuario lo desee]



INDICACIONES GENERALES

- 1.- A LO LARGO DE LAS EXPLICACIONES DE UTILIZACION DEL SISTEMA POR MEDIO DEL MANUAL DAREMOS INDICACIONES DE DIGITACION DE TECLAS LAS CUALES ESTARAN RESALTADAS ENTRE CORCHETES CUADRADOS.
- 2.- EL AREA DE MENU DE TODAS LAS PANTALLAS ESTA COMPUESTA POR LAS TECLAS DE FUNCION LAS CUALES EN OCASIONES NO SE PUEDEN ACCESAR DIGITANDO DIRECTAMENTE LA TECLA, EN ESTE CASO OPRIMA LAS TECLAS SIGUIENTES, YA SEA PARA PASAR DE EL AREA DE TRABAJO AL AREA DE MENU O VICEBERSA.

[CTRL] + [R]

- 3.- CUANDO SE INDICA QUE LOS CAMPOS SON MODIFICABLES QUIERE DECIR QUE SU CONTENIDO SE PUEDE ACTUALIZAR, PARA LLEVAR A CABO DICHA MODIFICACION SE DEBE LIMPIAR EL CAMPO E INVARIABLEMENTE SE REALIZARA ESTA FUNCION OPRIMIENDO LAS TECLAS SIGUIENTES:

[POSICION CAMPO A MODIFICAR]
[CTRL] + [E]

- 4.- PARA LOGRAR UNA NAVEGACION DENTRO DE LOS CUADROS DE OPCIONES QUE APARECEN AL ACCESAR UNA TECLA DE FUNCION DEL AREA DE MENU SE REALIZARA CON LAS TECLAS [FLECHA HACIA ARRIBA] O [FLECHA HACIA ABAJO]



MODULO

DE

COMPRAS

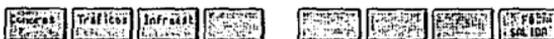


MANUAL SISTEMA DE ABASTECIMIENTO

SEGMENTO REQUISICIONES

PANTALLA PRINCIPAL LOGISTICA Y ABASTECIMIENTO

```
.....LOGISTICA Y ABASTECIMIENTO.....
71 Compras Trafico Infraestructura / 0
Act./Cons. -> Remision 1993 6:09:20PM
Consulta -> Requisicion
Catalogos -> Catalogos
Reportes de Pedidos -> Pedido:Colocar
Reportes de Requisiciones -> Pedido:Modificar
Novinto:Import.
Novinto:Nacional
.....
```



Para encontrarnos con el Segmento de Requisiciones en la Pantalla Principal del Sistema de Abastecimiento, se pide seleccionar las siguientes opciones.

.- Oprima [F1]

[Act./Cons.]

[Requisición]

PANTALLA DE REQUISICIONES

```
.....
1. Act. Cons.
2. Consulta
3. Catalogos
4. Reportes de Pedidos
5. Reportes de Requisiciones
6. Novinto:Import.
7. Novinto:Nacional
.....
LINEA 01 DESCRIPCION GENERAL
SUBLINEA 01 ESPECTROMETRO DE INFRARROJOS CENTRAL
.....
Acty 112 Compy 10 Pers 1001 H258 1124 1125 1126 1127 1128 1129 1130
```



DESCRIPCION DE LOS CAMPOS

En el momento que aparece la Pantalla de Requisiciones nos proporciona automáticamente los datos siguientes:

- a) El nombre del usuario que se encuentra trabajando en el Sistema.
Lo ubicamos al extremo superior derecho.
- b) La fecha del día actual
La ubicamos en el módulo de fechas extremo derecho.

Campos de Trabajo

1.- Centro	[clave de almacén]
2.- Requisición	[número de identificación de la requisición]
3.- Compañía	[Nombre de la Cía. responsable de la req.]
4.- Cuenta de Cargo	[Se cargan los montos para cada almacén]
5.- Autor. Desembolso	[Identificación para la adquisición]
6.- Monlo	[Suma a la que asciende la requisición]
7.- Depto.	[Departamento Solicitante]
8.- Solicitante	[Persona responsable del trámite]
9.- Folio	[Identificación consecutiva de la req.]
10- Comprador	[Clave y nombre responsable de la compra]
11- Fecha:	
Recb	[Fecha de recepción de la requisición]
Elab	[Fecha de elaboración de la requisición]
Reqd	[Fecha de requerimiento de la req.]
E/Coloc	[Fecha estimada de colocación]
12- Tipo	[Naturaleza de la requisición]
R	[Requisición de reparación]
A	[Requisición de adquisición]
S	[Requisición de sustitución]
13- Prioridad	[Tiempo de entrega estimada que se asigna]
A	[Extra urgente]
B	[Urgente]
C	[Normal]
14- Activo Fijo	
S	[Requisición de Activo Fijo]
N	[Requisición no es de Activo Fijo]
15- Partidas	[Proporciona el número de partidas] *
16- Colocadas	[No. partidas colocadas] *
17- Programadas	[No. partidas programadas] *
18- Notas	[No. de notas para esa requisición] &
19- Status	[Proporciona una clave de identificación] @
20- Fecha	[Ultima fecha de modificación de status] @
21- Línea	[Clave de material requerido]
22- Sublínea	[Subclave de material requerido]
23- Descripción Genérica	[Descripción de la línea de material]

* Datos actualizables desde la PANTALLA PARTIDAS [F3]

& Dato actualizable desde la PANTALLA NOTAS [F4]

@ Datos actualizables desde la PANTALLA DE PEDIDOS



DESCRIPCION DE AREA DE MENU

- [F1] [Actualiza]
[Alta] [Ingresa información al sistema]
[Cambio] [Permite modificar la información]
[Elim.] [Da de baja información de sistema]
[Transf] [Envía información a otros centros]
- [F2] [Consulta]
[Requisiciones] [Consulta de Requisiciones]
[Cotizaciones] [Consulta de cotizaciones]
[Pedidos] [Consulta de Pedidos]
[Movimientos] [Consulta de movimientos]
- [F3] [Ptd]
[Pantalla] [Pantalla para cargar Partidas]
[Requisiciones] [Consulta Partidas existentes]
- [F4] [Notas]
[Pantalla] [Pantalla de Notas]
[Notas] [Visualización Notas existentes]
- [F5] [Menu]
[Actualiza]
[Consulta]
[Reportes]
[Salida]
- [F6] [Limpia]
[Limpia todos los campos de la pantalla]
- [F7] [?]
- [F8] [Salida]
[Nos permite salir de pantalla]

EXPOSICION ESPECIFICA DE LOS COMANDOS



CAMPOS LLAVE:

- 1.- Centro: [clave almacén]
- 2.- Requisición: [Identificación de la requisición]

A partir de naturaleza de estos dos campos existen cuatro opciones a seguir:

TECLA ACTUALIZA [F1]

PRIMER OPCION

ALTA

- a) Los campos son la llave de la captura de otros datos [campos 3 al 23] requeridos para dar de Alta la requisición en Sistema, puesto que no ha sido registrada anteriormente.

Al completar la totalidad de los datos requeridos en la pantalla por parte del usuario, se deben seguir los pasos que a continuación se enlistan:

[F1]

[Alta]

[Enter]

SEGUNDA OPCION

- b) Cuando los campos pertenecen a una requisición ya existente a información de los campos 3 al 23 sera proporcionada automáticamente por el Sistema.

En este caso podemos considerar esta Pantalla para tres opciones de manejo de datos:

- 1) Pantalla de Consulta: al proporcionarnos toda la información cargada en Sistema acerca de una requisición específica.
- 2) Pantalla Modificable: nuestros datos por alguna razón pueden tener necesidad de ser y actualizados, para tal propósito en la Pantalla los campos 3 al 14 y 21 al 23 son modificables, dicha función la realizaremos como sigue:

[Posicionarse en el campo a modificar]
[Oprima [Ctrl] + [e] para limpiar el campo]
[Capture la información sustituta]
[Oprima [Enter] para aceptar]
[F1] + [Cambio] + [Enter]

Nota: Los campos 15 al 20 no son modificables en ningún caso, puesto que son determinados desde otros puntos del procedimiento.

TERCER OPCION

ELIM./CANCELA

En el momento de corroborar en sistema que los datos llave proporcionados pertenecen a una requisición existente, cabe la necesidad de Eliminar dicha requisición; esto lo lograremos siguiendo los pasos que a continuación se grafican.

[F1]
[Elim /Cancela] [Enter]



TRANSFERENCIA

Al momento de acceder la información y mantener la necesidad de atención de la requisición en otra localidad, tenemos que transmitir dicha requisición a archivos del almacén o localidad en donde se atenderá.

Realice los siguientes pasos:

[F1]
[Transferencia]
[Enter]

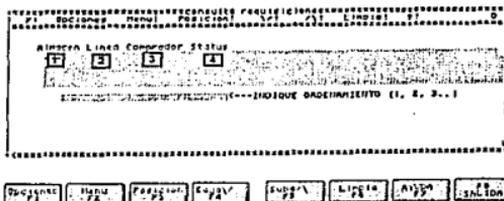
TECLA CONSULTA [F2]

Dentro de la Pantalla Principal de Requisiciones (Area de Menus) contemplamos la siguiente opción

[F2]
[Consulta]
[Requisición]
[Enter]

REQUISICIONES

PANTALLA DE CONSULTA DE REQUISICIONES



Proporciona una pantalla adicional donde tenemos la opción de consultar todas las requisiciones existentes con tan solo proporcionar los datos requeridos que son los siguientes:



DESCRIPCION DE DATOS

- | | |
|---------------|---|
| 1.- Almacen | [Clave almacén de consulta] |
| 2.- Línea | [Línea de material de la consulta] |
| 3.- Comprador | [Clave comprador responsable de la requisición] |
| 4.- Status | [Clave del status de las requisiciones] |

DESCRIPCION DEL AREA DE MENU:

- | | |
|-------------|---|
| [Opciones] | |
| [Resumen] | [Proporciona el concentrado de requisiciones] |
| [Selección] | [Captura de datos] + [F1] + [Enter] |
| [Menu] | |
| [Posicion] | [Sitúa el cursor en el primer campo]
[F3] + [Enter] |
| [V] | [Scroll abajo]
[Resumen de datos] + [F4] + [Enter] |
| [] | [Scroll arriba]
[Resumen de datos] + [F5] + [Enter] |
| [Limpia] | [Recorrido por c/u campos para dejarlos en blanco listos para la siguiente consulta] |
| [?] | [Ayuda proporciona información adicional de cada uno de los campos de la pantalla según en el que se encuentre posicionado el cursor] |
| [/] | [Salida nos da el paso de regreso a la Pantalla de Requisiciones] |

La siguiente opción del menú de Consulta es la Pantalla de Pedidos, donde tenemos la posibilidad de consultar los pedidos existentes que dependen de la requisición que estamos analizando

PANTALLA DE PEDIDOS

```

=====REQUISICIONES=====
| # | Actualiza | Consulta | B13 | Notas | B14 | B15 | B16 | B17 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 01 |          |          |          |          |          |          |          |          |
|-----|-----|-----|-----|-----|-----|-----|-----|
Requisición No. AF22: 6260 0 PR JUI 1 1993
Con Alm Pedido An ST Fco/ST Proveedor
85 A/T 6260 0 PR JUI 1 1993 CASA ROCAS, S.A. DE C.V.
| 7 | 3 | 4 | 5 | 6 | 7 | 8 |
=====
DUPLICAR 2-----BORRAR 7-----
  
```

Actualiza	Consulta	Pedidos	Notas	Menu	Limpiar	Ayuda	F6
-----------	----------	---------	-------	------	---------	-------	----

DESCRIPCION DE CAMPOS

- | | | |
|-----|-----------------|--|
| 1.- | Requisición No. | [Nos proporciona el no. de la requisición] |
| 2.- | Comprador | [Clave del Comprador] |
| 3.- | Almacen | [Clave del almacén] |
| 4.- | Pedido | [Clave del Pedido] |
| 5.- | Anexo | [Clave del anexo del Pedido] |
| 6.- | Status | [Clave de status del Pedido] |
| 7.- | Fecha de Status | [Fecha del último status del Pedido] |
| 8.- | Proveedor | [Información del proveedor designado] |

Nota: En el área de menú solo encontraremos la diagonal que representa el botón de salida de la Pantalla [/].



DESCRIPCION AREA DE MENU

[Actualiza]	
[Alta]	[Da de alta la partida detallada] [F1] + [Alta] + [Enter]
[Cambio]	[Actualiza cambios en los datos existentes] [F1] + [Cambio] + [Enter]
[Elim/C]	[Baja a los datos citados en área trabajo] [F1] + [Elim] + [Enter]
[Edit]	[Nos sitúa en la posición inmediata siguiente para ingresar datos] [F2] + [Enter]
[<->]	[Scroll horizontal en el área de despliegue] [F3] + [Enter]
[V]	[Scroll abajo] [F4] + [Enter]
[^]	[Scroll arriba] [F5] + [Enter]
[Mat./Prog]	[Material Programado]
[Cod.Unidad]	[Despliegue de Subpantalla Código Unidad] [F6] + [Cod.Unidad] + [Enter]
[Cod.Corpor]	[Despliega la clave de código corporativo para ese material] [F6] + [Cod.Corpor] + [Enter]
[Programa Plda]	[Despliegue de Pantalla Programa de Partidas relacionada a las partidas programadas para esa requisición]
[Actualiza]	[Modifica la información existente con los datos proporcionados recientemente]
[Elimina]	[Da de baja las partidas programadas]
[?]	[Ayuda para consultar información adicional a los campos en donde se encuentre posicionado el cursor] [Posición en campo] + [F7] + [Enter]
[/]	[Salida cierra la Pantalla actual cambiando a la Pantalla de Partidas de Requisiciones]



PANTALLA CODIGO MATERIAL

CODIGO DE MATERIAL	
Centro	[1] ALT
Código de Material	[2]
Unidad de medida	[3]
Activo fijo	[4]
Descripción material	[5]

Actualiza Consulta Material Corp Limpia

Actualiza Consulta Material Corp Limpia

La subpantalla de Código de Material nos permite consultar y modificar los códigos de material requerido.

DESCRIPCION DE CAMPOS

- 1.- Centro [Clave del Almacén]
- 2.- Código Material [Clave de material requerido]
- 3.- Unidad de Medida [Relación de identificación de unidad]
- 4.- Activo Fijo [Identifica si es activo fijo o no]
- 5.- Descripción Mat. [Descripción genérica del material]

DESCRIPCION DE AREA DE MENU

- | | |
|-----------------|--|
| [Actualiza] | [Permite ingresar el código no existente] |
| [Alta] | [F1] + [Alta] + [Enter] |
| [Cambio] | [Modificación de los códigos existentes] |
| | [F1] + [Cambio] + [Enter] |
| [Elim.] | [Da de baja las relaciones de códigos exist] |
| | [F1] + [Elim.] + [Enter] |
| [Consulta] | [Ver indicaciones en el menú de consultas en la Pantalla Principal de Requisiciones] |
| | [F2] + [Enter] |
| [Material Corp] | [Identifica si el material en cuestión esta considerado como material corporativo] |
| | [F3] + [Enter] |



[Limpia]

[Al presionar este botón todos los campos quedan en blanco listos para una nueva consulta].

[F6] + [Enter]

[Ayuda]

[Proporciona información adicional del campo donde el cursor este posicionado]

[F7] + [Enter]

PANTALLA DE CODIGO DE MATERIAL CORPORATIVO

LINEA	SUBLINEA	CODIGO	UNIDAD MEDIDA	DESCRIPCION

ACTUALIZA ALTA CAMBIO ELIM. AYUDA MENU

Pantalla que proporciona la información concerniente al Código Corporativo de el material en cuestión.

DESCRIPCION DE CAMPOS:

- | | | |
|-----|---------------|---|
| 1.- | Línea | [Línea de material de la requisición] |
| 2.- | Sublínea | [Sublínea de material de la requisición] |
| 3.- | Código | [Código Corporativo asociado al material] |
| 4.- | Unidad Medida | [Clave unidad de medida del material] |
| 5.- | Descripción | [Descripción de el material] |

DESCRIPCION DE AREA DE MENU

[Actualiza]

[Alta]

[Permite adicionar los Códigos Corporativos no existente en sistema]

[Captura Información] + [F1] + [Alta] + [Enter]

[Cambio]

[Para modificar los Códigos Corporativos existentes en sistema.]

[Modificación información] + [F1] + [Cambio] + [Enter]

[Elim.]

[Baja de los Códigos Corporativos que no son necesarios ya]



- [Información existente] + [F1] + [Elim.] + [Enter]
[Asoci/Codigos] [Proporciona los Códigos Asociados a el Código Corporativo en cuestión]
- [F2] + [Enter]
- [Limpia] [Al accesar esta opción todos los campos quedan en blanco para una siguiente consulta]
- [F6] + [Enter]
- [?] [Ayuda nos proporciona información adicional al campo donde este posicionado el cursor]
- [F7] + [Enter]

PANTALLA PARTIDAS PROGRAMADAS

La Pantalla de Partidas Programadas nos proporciona la facilidad de consultar o planear las partidas y sus programas de entrega, para tal fin seleccionaremos las siguientes teclas para poder accesarla.

- [F6]
[Programa Pida]
[Actualiza]
[Enter]

DESCRIPCION DE CAMPOS:

- | | | |
|-----|---------------|--|
| 1.- | Pta. | [No. de la partida que se esta consultando] |
| 2.- | Cantidad | [Cant. material requerido en dicha partida] |
| 3.- | Fecha inicial | [Fecha de inicio del programa de partidas] |
| 4.- | Num/Per | [No. de entregas programadas] |
| 5.- | Num/Per | [No. de períodos] |
| 6.- | Periód: | |
| | Meses | [Cada cuantos meses se realiza la entrega] |
| | Semanas | [Cada cuantas semanas se realiza la entrega] |



7.-	Ent	Días	[Cada cuantos días se realiza la entrega]
8.-	Fecha/Prog		[Indicación de las diferentes entregas]
9.-	Can/Periodo		[Fecha programada para cada entrega]
10.-	Suma		[Cantidad a surtir en cada entrega]
			[Suma de las cantidades de material]

DESCRIPCION DE AREA DE MENU

[?]		[Ayuda nos proporcionará información adicional a cada uno de los campos de la Pantalla dependiendo la posición del cursor]. [Posicionarse en un campo] + [F7] + [Enter]
[/]	[Acepta]	[Salida permite seleccionar dos opciones] [incluye el programa de partidas a sistema] [Programa de Partidas] + [F8] + [Acepta] + [Enter]
	[Cancela]	[Regreso a la Pantalla de Partidas] [Programa de Partidas] + [F8] + [Cancela] + [Enter]

TECLA NOTAS [F4]

La Pantalla nos proporciona un ambiente propicio para consultar, modificar e incluso dar de alta notas desplegables en la impresión de los pedidos pertenecientes a la requisición que hemos venido analizando.

Para visualizar la Pantalla de Notas seguiremos las indicaciones:

[F4]
[Enter]



PANTALLA DE NOTAS

```
.....
) 2| 1 Actualiza Catalogo Notas| 0
.....
Narración      Nota:      Fecha Jun 18 1993
.....
```

```
.....
) 2| 1 Actualiza      0
.....
Narración      Nota: AC      Fecha Jun 18 1993
.....
ASEGURAMIENTO DE LA CALIDAD
.....
El proveedor deberá incluir en cada embarque el "Certificado de Control
de Calidad" correspondiente, así como el "Certificado de Origen" cuando
proceda, y todos los reportes de pruebas que se hayan desarrollado
(químicos, físicos, etc.). Cuando existan, deberán entregarse también
Certificados de Análisis Físicos, Químicos y Gravitométricos, así como
los documentos necesarios que permitan garantizar la calidad de los
productos embarcados.
.....
```

DESCRIPCION DE CAMPOS:

- 1.- Narración [Es el área más amplia de la pantalla, puesto que en ella se despliega la descripción de la nota.]
- 2.- Nota [Clave de Catálogo de la nota acceso con las teclas [Ctrl] + [V] + [Enter]]
- 3.- Fecha [Fecha de alta de la nota]

DESCRIPCION DE MENU

- [Actualiza]
[Alta] [Cuando no existe la nota en sistema con esta opción podemos adicionarla]
[Captura Datos]+[F1]+[Alta]+[Enter].
- [Cambio] [Modificaciones a notas existentes]
[Modif. datos]+[F1]+[Cambio]+[Enter]
- [Elim.] [En el caso de no ser necesaria alguna de las notas del sistema con esta opción la damos de baja]
[Nota existente]+[F1]+[Elim]+[Enter]



[Catálogo Notas]

[Opción necesaria para acceder las notas que se encuentran residentes en sistema; en este caso es cuando podemos utilizar la facilidad del [Ctrl]+[V] mencionado anteriormente para consultar las claves de notas]
[F2] + [Enter]
Opcional desde campo notas [Ctrl + [V]

[?]

[Ayuda proporciona información adicional de cada uno de los campos de la Pantalla dependiendo de la posición del cursor]
[F7] + [Enter] desde cualquier campo

[/]

[Salida es la tecla que nos permite salir de la presente Pantalla y nos posiciona en la Pantalla de Requisiciones]
[F8]

TECLA MENU [F5]

Dentro de esta opción ubicamos la sección de navegación de la Pantalla de Requisiciones desde donde tenemos la opción de acceder diversas Pantallas de otros módulos del sistema a manera de Consulta, las cuales serán detalladas en sus Capítulos Correspondientes.

Nota: Menu sujeto a cambios.



DESCRIPCION DE CAMPOS

- 1.- Solicitud No. [No. de solicitud en cuestión, este campo puede capturarse el no. de una solicitud existente o presionar [Enter] para que el Sistema asigne el no. consecutivo siguiente a la solicitud que se dará de alta]
- 2.- Fecha de Solicitud [Fecha en que se realizó la solicitud de Cotización]
- 3.- Requisición [Captura de cve de Requisición para la cual deseamos conformar la Solicitud]
- 4.- Fec. Lim. Cot. [Fecha Limite de Cotización, es el campo donde se captura la fecha máxima de entrega de la respuesta de la Solicitud de Cotización por parte del Proveedor]
- 5.- Centro [Descripción o nombre del almacén que se esta haciendo cargo de la Requisición]
- 6.- Notas [En este campo el comprador podrá acentar ciertas observaciones relevantes con respecto a la Solicitud de Cotización]
- 7.- Partidas [Proporciona el Sistema en este campo el no. de partidas existentes para la requisición en cuestión]
- 8.- En Solicitud [Se debe capturar el no. de identificación de la partida o partidas que deseamos incluir en la Solicitud de Cotización; al presionar [Enter] en esta opción se asumirán la totalidad de las partidas existentes]
- 10- Descripción [Descripción del material o línea de material requerida]



DESCRIPCION DEL AREA DE MENU

[Act]	[Cambio]	[Permite realizar modificaciones a cierta solicitud ya existe] [F1] + [Cambio] + [Enter]
	[Baja]	[Cuando una solicitud ya no es necesaria es conveniente que se cancele de Sistema por razones de optimización y con tal fin se creó esta opción] [F1] + [Baja] + [Enter]
[Movtos]	[Selección Req]	[Subpantalla donde tenemos la opción de indicar a esta parte de la consulta que deseamos considerar más de una requisición en la Solicitud, y así indicamos el no. cve. de las requisiciones a incluir] [F2] + [Selección Req] + [Enter]
	[Partidas Sel]	[Subpantalla para poder realizar la selección detallada de las partidas a incluir en la Solicitud] [F2] + [Partidas Sel] + [Enter]
	[Partidas Aut]	[En el caso que no deseemos seleccionar las partidas de la o las requisiciones para la Solicitud en cuestión se puede utilizar esta opción la cual nos permite: Alta: Da de alta las partidas de las requisiciones incluidas en la Solicitud si aún no están consideradas [F2] [Partidas Automático] [Alta] [Actual Req./Todas Req] Baja: Da de baja las partidas de las requisiciones incluidas en la Solicitud. [F2] [Partidas Automático] [Baja] [Actual Req./Todas Req]



[Prov.]

[Asigna]

[Subpantalla donde se captura la Razón Social del Proveedor designado (basta con capturar la primer palabra de la razón social y el sistema automáticamente completará dicho campo además de la información requerida en cada uno de los campos de la Subpantalla)]

[Consulta]

[F3] + [Asigna] + [Enter]

[Subpantalla Proveedores por Solicitud que muestra los proveedores posibles para dicha solicitud]

[F3] + [Consulta] + [Enter]

[Req. Ant.]

[Permite visualizar cuando se estan considerando más de una requisición, las mismas en la pantalla en orden regresivo]

[F5] + [Enter]

[Req. Sig.]

[Es el mismo caso que el anterior, pero el orden de consulta será progresivo]

[F6] + [Enter]

[Sol Asoc.]

[Nos proporciona a través de una pantalla todas las Solicitudes de Cotizaciones derivadas de una requisición específica]

[F7] + [Enter]

[/]

[Este botón contiene las ultimas dos opciones de la pantalla.

a) Otra Solicitud: Limpia todos los campos de la pantalla dejandolos listos para una nueva consulta

b) Salida: Al oprimir esta opción saldremos de la Pantalla.

[F8] + [Enter]



PANTALLA DE SELECCION DE REQUISICION

```
*****Requisiciones*****
| 2 |
| Almacén APT: ICT-ENLX PENOLES |
| De la requisición 25000 a la 25009 Total 10 |
```

DESCRIPCION DE CAMPOS

- 1.- Almacén [Cvè. y descripción del almacén cargo]
- 2.- De la requisición [Campos donde se capturará el rango de claves de las requisiciones]
- 3.- Total [Se indica el no. de requisiciones a considerar según el rango capturado en el campo anterior]

DESCRIPCION AREA DE MENU

- [/]
- [En esta pantalla aparecen tres funciones las cuales emplearemos según las necesidades de cada usuario:
- Acepta:** Dará de alta el rango de requisiciones capturadas en la pantalla para que sean consideradas en la Solicitud.
- Cancela:** Es para salir de la Pantalla sin asumir los datos capturados en el caso de que los haya.
- Limpia:** Quedarán los campos en blanco listos para capturar otro rango de requisiciones.



PANTALLA DE SELECTIVO PARTIDAS

```

=====PARTIDAS DE COLIZACION=====
| Actualiza Scroll | 0
| Almacén: [ ] |
|-----|
| Part | Cod Material | Descripción | UTI | Cantidad |
|-----|-----|-----|-----|-----|
| [A] | [ ] | [ ] | [ ] | [ ] |
| [B] | [ ] | [ ] | [ ] | [ ] |
|-----|-----|-----|-----|
| [5] | [ ] | [ ] | [ ] | [ ] |
|-----|-----|-----|-----|

```

[Actualiza] [Scroll] [Almacén] [Partidas] [UTI] [Cantidad] [AYUDA] [SALIDA]

DESCRIPCION DE CAMPOS

- 1.- Almacén [Cve. del almacén]
- 2.- Requisición [Cve. de la Requisición]
- 3.- [Descripción del nombre del almacén]
- 4.- Area de despliegue [Es el área media de la pantalla donde se desplegarán las partidas para esa requisición, donde aparecen sombreadas las que ya se incluyeron dentro de la Solicitud de Cotización]
- 5.- Area de trabajo [Area donde se seleccionará el no. de partida que se desea incluir o excluir de la Solicitud]

DESCRIPCION DEL AREA DE MENU

[Actualiza]

[Incluye Par] [Con esta opción incluiremos las partidas seleccionadas en el área de trabajo siendo estas necesarias únicamente]

[Incluye tod] [Al optar por esta opción no es necesario seleccionar en el área de trabajo nada y el resultado es que la totalidad de las partidas estarán incluidas en la Solicitud de Cotización]



[Excluye Par] [Nos permite dejar fuera algunas partidas de la Solicitud]
[Excluye tod] [Con esta opción excluimos todas las partidas existentes para una Solicitud]

[Scroll]

[Tecla que posicionandose en el área de despliegue de partidas nos permite visualizar las partidas que por espacio se localizan en la pantalla]

[/]

[Nuevamente en este botón encontramos dos opciones:

Limpia: Para dejar en blanco los campos y así poder realizar otra consulta

Salida: Para dejar la Pantalla

PANTALLA DE SOLICITUD DE COTIZACION

PANTALLA DE SOLICITUD DE COTIZACION			
#	Selección	Formato	0
1	1	RAZON SOCIAL: DISEÑO Y DISEÑO: SAREC.U	2 TAB 830117
2	2	Atencion: SENGIO TORRES	
3	3	Calle: ESTRELLA: 40144: PISO: A: A: B	
4	4	Colonia: FUERTE	
5	5	Delegacion: CDMX	
6	6	País: MEX	3 Estado D.F.J. C. P. 04309 10
11	11	Telefono: 583-12-25	Fon FAX: 713-46732 12

F1 Salir	F2 Formato	F3	F4	F5	F6	F7	F8 SALIDA
----------	------------	----	----	----	----	----	-----------

Pantalla que accedemos desde el Menu de Proveedor seleccionando la opción de asigna, como ya se habia mencionado.



DESCRIPCION DE CAMPOS

1.- Razón Social	[Campo que contendrá el nombre del proveedor asignado]
2.- R.F.C.	[Registro Federal de Causantes del mismo proveedor]
3.- Representante	[Si el proveedor es extranjero este campo contendrá los datos del proveedor intermediario en México]
4.- Atención	[Nombre de la persona contacto con el proveedor]
5.- Calle	[Calle de ubicación del proveedor]
6.- Colonia	[Dato de dirección]
7.- Delegación	[Dato de dirección]
8.- País	[Especificación del país donde se ubica la dirección antes descrita]
9.- Estado	[Estado donde se encuentra ubicada la dirección antes descrita]
10- C.P.	[No. de Código Postal de la dirección]
11- Teléfono	[No. telefónico contacto del proveedor]
12- Fax	[No. de fax de contacto del proveedor]

DESCRIPCION DEL AREA DE MENU

[Selección]	[Subpantalla Selección de Proveedor donde podremos seleccionar el proveedor para nuestra Solicitud de Cotización] [F1] + [Enter]
[Formato]	[Al oprimir esta opción se enviará a impresión nuestra Solicitud de Cotización] [F2] + [Enter]
[/]	
[Limpia]	[Con esta función dejaremos en blanco todos los campos de la pantalla lista para realizar otra consulta] [F8] + [Limpia] + [Enter]
[Salida]	[Nos va proporcionar la posibilidad de salir de la presente pantalla para proseguir con nuestro proceso de la Solicitud de Cotización]



[Ejecuta]

DESCRIPCION DEL AREA DE MENU

[A] acceder esta opción el Sistema realiza la búsqueda de datos que cumplan con las condiciones que se proporcionaron con la captura en los diferentes campos de la forma; cabe aclarar que no es necesario que se proporcione información en todos los campos, basta con proporcionar la mayor cantidad de datos para que nuestra discriminación de datos sea la más exacta posible]

[F1] + [Enter]

[Acepta]

[A] Seleccionar la Razón Social que deseamos indicar en la Solicitud accedamos esta opción para que automáticamente se asigne en la Pantalla de Solicitud Cotización]

[F8] + [Acepta] + [Enter]

[Cancela]

[Nos permitirá salir de la Pantalla de Selección de Proveedor para regresar a la Pantalla de Solicitud de Cotización]

[F8] + [Cancela] + [Enter]

SUBPANTALLA DE PROVEEDORES POR SOLICITUD

*****SOLICITUD DE COTIZACION*****	
7 Act. Navios. Prom. Reg. Ant. I	Reg. Sig. I Sol. Asoc. I
Fecha de Solicitud Jun.30.1993 11:03:56PM	
Solicitud No. 7	Proveedores Por Solicitud 0
Fec. Lin. Co. 7	Proveedor AG-830117-1
Notes: COTIZACION 2 de 2	
Cotizacion 3 Ho. 1 de 2 proveedores.	
Partidos 0 en solicitud 13	
Descripcion Bateria Chincheros y Rotinas	





DESCRIPCION DE CAMPOS

- 1.- Proveedor [R.F.C. y nombre del Proveedor asociado a cierta Solicitud de Cotización]
- 2.- Cotización [Se indica el no. de la cotización en cuestión]
- 3.- No. [Indicativo del orden que ocupa el proveedor que se esta visualizando en la Pantalla]
- 4.- de [Cantidad total de proveedores asignados para la presente Solicitud de Cotización]

PANTALLA DE MENU SOLICITUDES ASOCIADAS

-----DEFAULT MENU-----

1	Requisición REQ 125408 No. Solicitud Proveedor	SOC.
2		licitud

F9 SALIDA

DESCRIPCION DE CAMPOS

- 1.- Requisición [Capture Cve. de identificación de la requisición en cuestión]
- 2.- Zona de despliegue [Zona donde se desplegarán los datos de número de solicitud y proveedor de las solicitudes asociadas para la requisición que capturamos]



DESCRIPCION DE AREA DE MENU

- [/]
- | | |
|-----------|---|
| [Acepta] | {Al indicar una de las solicitudes desplegadas con la consulta antes descrita y tomar esta opción, estamos asumiendo que es la solicitud que deseamos accesar para emplearla en la petición de Cotización.} |
| [Cancela] | {Nos va a permitir salir de la Pantalla cuando ya no necesitemos esta clase de consulta} |
| [Limpia] | {La función de esta opción es dejar en blanco todos los campos de la Pantalla listos para la siguiente consulta} |

De esta manera finalizamos el Módulo de Solicitud Cotización donde al dar termino a todo el Proceso Contenido en él, tendremos como resultado la Solicitud o Solicitudes de Cotización necesarias para dar un paso más dentro de llamado Proceso de Abastecimiento.



Cotización sin tomar en cuenta la Solicitud de Cotización.

DESCRIPCION DE CAMPOS

- | | |
|-------------------------|--|
| 1.- Cve. Cotización | [Cve. numérica de identificación de la Cotización la cual debe ser proporcionada por el Usuario] |
| 2.- Fecha de Cotización | [Es la fecha cuando se realizó el trámite de la Cotización] |
| 3.- Proveedor /RFC | [Nombre y Registro Federal de Causantes del Proveedor designado para la Cotización] |
| 4.- No. Solicitud | [Cve. numérica de la Solicitud de Cotización especificado cuando existe] |
| 5.- Requisición | [Cve. identificación de la Requisición] |
| 6.- Fecha Vigencia | [Fecha a la cual es valida la Cotización] |
| 7.- Centro | [Nombre del Almacén especificado en la requisición] |
| 8.- Moneda | [Cve. y nombre del tipo de Moneda pactada para la Cotización] |
| 9.- Cond. Pago | [Cve., descripción y días de condiciones de pago] |
| 10- Entrega Cond. | [Cve. condiciones de entrega; las claves se pueden acceder mediante:
[Ctrl] + [V]
[Posición Cve. deseada]
[Enter]
[Infor. en pantalla] |
| 11- Entrega Lugar | [Cve. condiciones de entrega: las claves se pueden acceder mediante:
[Ctrl] + [V]
[Posición Cve. deseada]
[Enter]
[Infor. en pantalla] |
| 12- Entrega Tiempo | [Captura del no. de días por transcurrir estimado] |
| 13- Entrega Fecha | [Fecha calculada base a días capturados por transcurrir] |
| 14- Dcto. General | [Porcentaje de descuento general] |
| 15- Dcto. por Partida | [Porcentaje de descuento para cada partida] |
| 16- Dcto. Pronto Pago | [Porcentaje descuento por pronto pago] |



		capturó]
	[Cancela]	[Nos permite salir de el cuadro de diálogo
	[Limpia]	[Deja los campos en blanco y listos para capturar otro criterio de búsqueda]
	[F3] + [Enter]	
[C. Ant]		[Nos permitirá visualizar en Pantalla las cotizaciones anteriores si es que existieran]
[C. Siguiete]		[Nos permitirá visualizar en Pantalla las cotizaciones anteriores si es que existieran]
[Sol. Asociada]		[Subpantalla detallada en el Segmento de Solicitud de Cotización la cual nos proporciona información concerniente a las Solicitudes de Cotización vinculada con la Cotización]
[/]		[Dos opciones para este bolón]
	[Otra cotización]	[Limpia todos los campos de la forma para que se pueda realizar otra consulta de cotización]
	[Salida]	[Nos permite salir de la Pantalla de Cotizaciones]



17- Dias Pronto Pago

[No. de días para considerarse de
Pronto Pago]

DESCRIPCION AREA DE MENU

[Act]	[Botón con tres opciones]
[Alta]	[Dará de alta la Cotización que no existe en Sistema previa captura de datos en cada uno de los campos de la Forma]
[Baja]	[Utilizado para borrar una Cotización de Sistema]
[Cambio]	[Cuando se requiere modificaciones en la información existente en Sistema correspondiente a una Cotización se realizarán los siguientes pasos: [Posición campo modificable] [Ctrl] + [V] [Campo en blanco] [Captura nva. información] [F1] + [Cambio] + [Enter]
[Consulta]	
[Requisiciones]	[Requisiciones Asociadas] [F2] + [Requisiciones] + [Enter]
[Cotizaciones]	[Cotizaciones asociadas] [F2] + [Cotizaciones] + [Enter]
[Pedidos]	[Pedidos asociados] [F2] + [Pedidos] + [Enter]
[Movimientos]	[Movimientos de material asociados] [F2] + [Movimientos] + [Enter]
[Solicitud]	[Cuadro de diálogo para especificar las requisiciones que se desea que sean consideradas dentro de la Cotización: .- Almacen [Cve. y nombre almacén] .- Requisición [No. de requisición criterio] .- A la [No. de requisición cierre de criterio] Area de Menu: [I] [Acepta] [Se asume la base criterio que se



La presente pantalla nos brinda los campos a capturar para colocar un pedido. Los campos han sido marcados con números progresivos para indicar el orden de captura de la información; a continuación se detalla cada uno de ellos.

DESCRIPCION DE CAMPOS

- 1.- Clave del Comprador [Dato indicado por sistema tomado clave acceso]
2.- Nombre de Comprador [Dato indicado por sistema tomado clave acceso]
3.- Centro [Teclee la clave del Centro a Requisar.]

Rango de requisiciones para un pedido normal o para un pedido maestro programado o no

- 4.- Requisición [Para un pedido normal]
[Capture el número de requisición [ret]
Desaparece no. de la requisición [ret]
Aparece leyenda "Muestra cotizaciones?"]

- 4.- Requisición inicial [Para un pedido maestro]
5.- Requisición final

[Capture el no. de requisición inicial [ret]
Capture el no. de requisición final [ret]
Desaparece no. de la requisición [ret]
Aparece leyenda "Muestra cotizaciones?"]

- 6.- Cotización [En este campo se captura el no. de cotización si ya la conocemos]

Nota: El cursor se posiciona en este campo sólo si no se capturó ningún dato en los campos 4 y 5.

- 7.- "Muestra Cotizaciones" S/N ["S" Despliega las cotizaciones existentes para esa requisición o rango de requisiciones. Campo 8. "N" No despliega ningún dato adicional]

En ambos casos después de ejecutar la instrucción el cursor pasará al Campo 9.

- 8.- Area desplegado [Muestra el no. requisiciones con sus cotizaciones]

- 9.- No. Referencia de Requisición [El contenido de este campo esta supereditado a la decisión del campo 7 como ya se mencionó]

[Cotizaciones]
[Capture el no. cotización a registrar [ret]
Al existir solo una cotización [ret]]



[No cotizaciones]

[Al no existir cotizaciones [ret]]

10.- Razón Social Proveedor

[Contiene la Razón Social del Proveedor]

[Opciones:

Capturar totalmente la Razón Social
Capturar por lo menos las cinco primeras
letras de la Razón Social del Proveedor
en mayúsculas.

El sistema nos proporcionará un cuadro
de diálogo que contendrá todos los re-
gistros cuyos datos coincidan con las
letras capturadas.

Para seleccionar la opción correcta se
requiere iluminarla navegando con las
flechas y posteriormente pulsando F8.]

Al obtener todos los datos que son requeridos por la primer pantalla solo
resta teclear F1, para pasar a la siguiente pantalla.



Al modificar los campos precio o de la partida:

- Se deja en blanco este campo
- Oprima F3 (Posición) para posicionar el cursor en la columna de precio
- Para modificar datos existentes pulse al mismo tiempo Ctrl+ E para borrarlos y capture la información deseada [ret]
- Para modificar alguno de los campos de la partida adicionales a los antes citados:
- Teclee el no. de partida a modificar dentro del Campo 9, y así editarlos.
- Para modificar datos existentes pulse al mismo tiempo Ctrl+ E para borrarlos y capture la información deseada [ret]

Al término de cualquiera de las situaciones anteriores debemos de determinar si es un Pedido Normal o un Pedido Programado.

* Pedido Normal

[F1]

[Actualizar]

[ret]

[F8]

[Acepta/Selec]

[ret]

"Pantalla Siguiete"

[F8]

[Cancela/Selec]

[ret]

"Salida"

* Pedido Programado

[F1]

[Programa]

[ret]

"Pantalla Programados"

[F8]

[Cancela/Selec]

[ret]

"Salida"

PANTALLA DE DATOS GENERALES

```

=====Situación de Pedidos=====
| 2 | Actuales | Consultas | Pidi | Retras | S.C. | ZORRACION | Demu | 0 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
1 | Usuario | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
2 | Region | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
3 | Motor | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
4 | Proveedor | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
5 | Fecha | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
6 | Forma de Pago | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
7 | Mon. Entradas | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
8 | Fecha de Embarque | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
9 | Desc. y P.P. | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
10 | Dias P.P. | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
11 | Descuento | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
12 | Valor Bruto | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
13 | Valor Descuento | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
14 | Valor Neto | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
15 | Imp. Pagado | 18 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 | 00000001 |
=====
| F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10 | F11 | F12 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| R | A | B | C | D | E | F | G | H | I | J | K |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| P | A | G | I | N | A | D | O |

```

Al aparecer la forma en la pantalla nos proporciona ciertos campos con información obtenida de la PANTALLA DE ASIGNACION DE PARTIDAS, los cuales son indicados.

De dichos datos algunos son modificables:

- 5 Fecha elaboración
- 10 Atención
- 11 Forma de pago
- 12 Dias forma de pago
- 13 Descuento por pronto pago
- 14 Dias pronto pago
- 21 Fecha de embarque
- 22 Moneda
- 24 Descuento
- 25 Iva



Para modificarlos se deben seguir los pasos que hemos venido realizando para cualquier modificación:

[Posición campo a modificar]
[Ctrl] + [E]
[Datos deseados]
[Enter]

Los campos que no contienen ninguna información al momento de aparecer la forma, son los que se requiere que se capturen para finalizar la colocación del pedido.

Para eficientar la captura existen cuadros de consulta que contienen datos relativos a cada uno de los campos marcados con los subíndices 11, 15,16,17,20,22.

Para acceder dicha información se requiere:

[Posicionarse en el campo] [flecha arriba-abajo] ó [tab]
[Enter]
[Ctrl] + [V]
[Cuadro de consulta]

[Visualización de todos los datos] [flechas [arriba] [abajo]]
[Dato requerido]
[Enter] [Se asume en pantalla]

Nota: Los campos 18 y 19 son dependientes del campo 3 puesto que corresponden al almacén del pedido en cuestión.

El campo 20 sólo aparecerá en el caso de que el pedido sea de importación, lo cual podemos verificar en la leyenda que aparece abajo de la especificación de Proveedor.



DESCRIPCION DE CAMPOS

1.- Usuario	[Cve. del responsable de modificación]
2.- Nombre	[Nombre responsable de modificación]
3.- Pedido	[Claves de centro, requisición, anexo]
4.- St	[Status y fecha de status]
5.- F. Elab.	[Fecha que se realizó dicho Pedido]
6.- Autoriza	[Indica si el Pedido esta autorizado]
7.- Partidas	[No. de Partidas pertenecientes]
8.- Notas	[No. de notas para este Pedido]
9.- Proveedor	[Nombre del Proveedor del Pedido]
10- Atn:	[El contacto del proveedor]
11- Forma de Pago	[Cve. identificación de pago a realizar]
12- Dias F. Pago	[Los días de plazo pactados para pago]
13- Desc. x PP	[Porcentaje de descto. pactado]
14- Dias P.P.	[No. días considerados como Pronto Pago]
15- Con. Entrega	[Cve. identif. condiciones entrega]
16- Lugar de Entrega	[Cve. y descripción lugar entrega]
17- Via de consig.	[Cve. via de consignación]
18- Dom. Consignar	[Cve. domicilio a consignar]
19- Dom. Fac	[Clave domicilio a facturar]
20- Agente Aduanal	[Cve. agente aduanal sólo en pedidos no nacionales]
21- F. Emb	[Fecha embarque]
22- Moneda	[Cve. identificación de moneda pactada]
23- Valor Bruto	[Costo Bruto del Pedido]
24- Descuento	[Porcentaje y valor de descuento]
25- Iva	[Porcentaje y valor de iva]
26- Valor Neto	[Valor después de desctos del Pedido]

DESCRIPCION AREA DE MENU

[ACTUALIZA]	
[Alta/Cambio]	[Dará de alta el pedido si no existe] [F1] + [Alta/Cambio] + [Enter]
[Cancela]	[Cancela las modificación realizadas] [F1] + [Cancela] + [Enter]
[Autoriza]	[Autoriza pedido con ultiima modif.] [F1] + [Autoriza] + [Enter]
[Cierre]	[Cierra el pedido en cuestión 'CE'] [F1] + [Cierre] + [Enter]



[CONSULTA]

[Requisiciones][Presenta Pantalla que permite consultar las requisiciones asociadas al pedido]
[F2] + [Requisiciones] + [Enter]
[Cotizaciones] [Proporciona cotizaciones existentes]
[F2] + [Cotizaciones] + [Enter]
[Movimientos] [Movimientos de Material asociados]
[F2] + [Movimientos] + [Enter]
[Pedidos] [Subpantalla para consulta de pedidos según criterios de captura]
[F2] + [Pedidos] + [Enter]
[Mont.x Prov] [Pantalla para captura de rango de fechas y así determinar Monto de compra requerido a cada proveedor]
[F2] + [Mont.xProv] + [Enter]

[PARTIDAS]

[Pantalla donde consultaremos las partidas de el pedido en cuestión, así como los programas de abastecimiento]
[F3] + [Enter]

[NOTAS]

[Pantalla que permite incluir, consultar, o dar de baja las notas necesarias para el pedido en cuestion]
[F4] + [Enter]

[D.C.]

[Dom.Consignar][Pantalla donde especificamos el domicilio a consignar correspondiente a el pedido]
[F5] + [Dom.Consig] + [Enter]
[Dom.Facturar] [Pantalla donde especificamos el domicilio a facturar correspondiente a el pedido]
[F5] + [Dom.Facturar] + [Enter]

[IMPRESION]

[Pedido] [Impresión del Formato del Pedido]
[F6] + [Pedido] + [Enter]
[Pedido/acuse] [Impresión del Pedido y su acuse de recibo]
[F6] + [Pedido/acuse] + [Enter]



[MENU]

- [Actualización]
[Requisición] [Permite entrar al Segmento de Requisiciones del Menu Principal del Sistema para posibles modificaciones]
[F7] + [Requisición] + [Enter]
- [Colización] [Con esta opción podemos acceder al Segmento Cotizaciones del Menu Principal del Sistema para su consulta o modificación]
[F7] + [Colización] + [Enter]
- [Pedidos/Col] [Pasará al Segmento de Pedidos Colocar del Sistema de Abastecimiento]
- [Movimientos] [Accesa el Segmento de Movimientos de Material del Menú Principal del Sistema]
- [/] [El botón representado por una diagonal nos proporciona dos opciones:
Limpia: Deja en blanco todos los campos de la forma listos para una nueva consulta.
[F8] + [Limpiar] + [Enter]
Salida: Permite salir de la Pantalla
[F8] + [Salir] + [Enter]

EXPLICACION ESPECIFICA DE LAS PANTALLAS DEPENDIENTES SEGMENTO PEDIDO MODIFICA

PANTALLA ESTATUS DE REQUISICIONES

1	Almacén	MGZ1
2	Comproador	1E
3	Status	SD
4	Fecha Inicial	930601
5	Fecha Final	930630

DESCRIPCION DE CAMPOS

1.- Almacén

[Cve. del Centro abastecedor]



- 2.- Comprador
- 3.- Status
- 5.- Fecha inicial
- 6.- Fecha final

- [Cve. del comprador responsable]
- [Cve. de status de requisición a consultar]
- [Fecha inicial de consulta]
- [Fecha final de consulta]

DESCRIPCION DE MENU

[REPORTE]

[Genera el reporte de acuerdo a los datos parámetro capturados en los campos anteriores]
[F1] + [Enter]

[/]

- [Limpia] [Deja todos los campos en blanco listos para una nueva consulta]
[F8] + [Limpia] + [Enter]
- [Salida] [Permite salir totalmente de la Pantalla de Consulta de Requisición]

REPORTE DE REQUISICIONES POR STATUS

COMP.	RIN	Requis	Fecha Recp	Descripcion	Generice	ST	Fecha
02	APT	6715	May 6 1993	ESPECTROMETRO JORN DOP	Equipo	SC	May 1
02	APT	6717	May 18 1993	IMP IMPACTO TOTAL TIPO DE SET		SC	May 1
02	APT	6718	Jun 4 1993	MÓDULO ELECTRICAS DE INYECCION		SC	Jun 1
02	APT	6719	Jun 8 1993	MATERIALES DE LABORATORIO		SC	Jun 1
02	APT	6720	Jun 8 1993	MÓDULO DE CONTROL		SC	Jun 1
02	APT	6721	Jun 8 1993	TRILINDO DE COMBUSTION		SC	Jun 1
02	APT	6722	Jun 9 1993	INSTRUMENTOS ELECTRONOMETRICOS		SC	Jun 1
02	APT	6723	Jun 9 1993	MÓDULO ELECTRICOS		SC	Jun 1
02	APT	6724	Jun 9 1993	MATERIALES REQUISICIONADOS		SC	Jun 1
02	APT	6725	Jun 21 1993	MOTOR SINCRONO		SC	Jun 1
TOTAL ROUTOS 18 TOTAL DE DIAS 268 PROXIMO DIAS ROUTO 28							

Nota: Para salir de la presente Pantalla de Consulta oprima:
[F8]
[Close]
[Enter]
[Pantalla de Status Requisición]

Al llegar a la Pantalla de Status Requisición tenemos dos opciones:

- a) Limpiar la Pantalla para consultar otro tipo de Requisiciones



[F8]

[Enter]

[Captura de datos]

[F1]

[Nvo.Reporte]

b) Salir de la Pantalla

[F8]

[Salida]

[Pantalla de Actualización de Pedidos]

PANTALLA DE CONSULTA DE PEDIDOS

Reporte	
<input type="checkbox"/> Almacén	ALM
<input checked="" type="checkbox"/> Comprador	COMPRADOR
<input type="checkbox"/> Status	STATUS
<input type="checkbox"/> Fecha Inicial	920001
<input type="checkbox"/> Fecha Final	920030

DESCRIPCION DE CAMPOS

- 1.- Almacén
- 2.- Comprador
- 3.- Status

[Cve. del Centro abastecedor]
[Cve. del comprador responsable]
[Cve. de status de requisición a consultar]

- 5.- Fecha inicial
- 6.- Fecha final

[Fecha inicial de consulta]
[Fecha final de consulta]

DESCRIPCION DE MENU

[REPORTE]

[Genéra el reporte de acuerdo a los datos parámetro capturados en los campos anteriores]
[F1] + [Enter]

[/]

[Limpia]

[Deja todos los campos en blanco listos para una nueva consulta]
[F8] + [Limpia] + [Enter]

[Salida]

[Permite salir totalmente de la Pantalla de Consulta de Requisición]



[F8] + [Salida]

REPORTE DE PEDIDOS POR STATUS

Cond.	Año	Conse	Año F.	Entrega	Descripcion Generica	ST	
07	077	2074	0	JUL 2 1992	MOTOFLECTORES MARCA	VF	
07	077	2164	0	OCT 1 1992	MOTOFLECTOR MARCA PHILADELPHIA	VF	
07	077	2130	0	JUL 2 1992	MOTOFLECTOR DE 8 H.P. MARCA	VF	
07	077	2102	0	FEV 5 1993	INDICADOR MARCA PHILADELPHIA	VF	
06	077	2209	0	AGO 7 1992	CLASIFICADOR DE AUTOMATA	VF	
06	077	2209	0	AGO 7 1992	BANCO DE RECTIFICACION Y PANTES	VF	
06	077	2145	0	AGO 7 1992	DEMANDO DE INGENIERIA PARA	VF	
06	077	2145	0	AGO 7 1992	ALTADEOR PARA PAISES DE	VF	
06	077	2145	0	AGO 7 1992	RDG. ELECTRONICO DE ALASKA	VF	
06	077	2101	0	AGO 24 1993	RDG. DE UN POLARONIMO	VF	
TOT. MONTOS:					35 TOTAL DE DIAS	2649 PROMEDIO DIAS MONTOS.	75

[F8] [F7] [F6] [F5] [F4] [F3] [F2] [F1] [Nvo.Reporte]

Nota: Para salir de la presente Pantalla de Consulta oprima:

[F8]

[Salida]

[Enter]

[Pantalla de Status Requisición]

Al llegar a la Pantalla de Status Requisición tenemos nuevamente

dos opciones:

a) Limpiar la Pantalla para consultar otro tipo de Requisiciones

[F8]

[Enter]

[Captura de datos]

[F1]

[Nvo.Reporte]

b) Salir de la Pantalla

[F8]

[Salida]

[Pantalla de Actualización de Pedidos]

PANTALLA DE MONTOS POR PROVEEDOR

Reporte	
1	Fecha Inicial: JUN 1
2	Fecha Final: JUN 30
3	Ordenado por proveedor
4	Ordenado por monto



DESCRIPCION DE CAMPOS

- | | |
|----------------------------|--|
| 1.- Fecha inicial | [Fecha de inicio del rango a consultar] |
| 2.- Fecha final | [Fecha de termino del rango a consultar] |
| 3.- Ordenado por proveedor | [Campo donde señalaremos con "x" si deseamos el orden de nuestro reporte por proveedor] |
| 4.- Ordenado por monto | [Campo donde señalaremos con "x" si deseamos el orden de nuestro reporte por monto total del pedido] |

DESCRIPCION DE MENU

[REPORTE]

[Genéra el reporte de acuerdo a los datos parámetro capturados en los campos anteriores]
[F1] + [Enter]

[/]

[Limpia]

[Deja todos los campos en blanco listos para una nueva consulta]
[F8] + [Limpia] + [Enter]

[Salida]

[Permite salir totalmente de la Pantalla de Consulta de Requisición]



13- Descripción	[Descripción material zona despliegue]
14- Part.	[Partida zona de trabajo modificación]
15- Cant.	[Cantidad zona de trabajo modif.]
16- Precio Uni.	[Precio Unitario zona de trabajo modif]
17- Descuento	[Descuento zona de trabajo modif.]
18- Requisición	[Identificación requisición zona de trabajo]
19- Partida Requisición	[No. de partida de requisición zona trabajo]
20- Status	[Status del pedido zona trabajo modif]
21- Fecha actual	[Fecha del día actual zona de trabajo]
22- Fecha programada	[Fecha de entrega de material programada zona de trabajo]
23- Descripción	[Descripción del material zona de trabajo]

DESCRIPCION AREA DE MENU

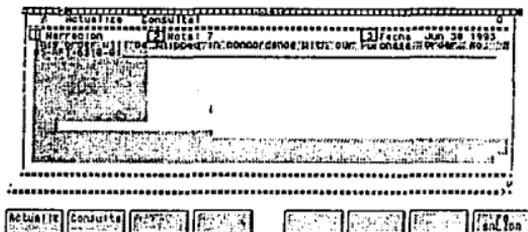
[ACTUALIZA]	
[Actualiza]	[Autoriza los posibles cambios realizados en la pantalla]
	[F1] + [Actualiza] + [Enter]
[Programa]	[Cambio a la Pantalla de Programas de Embarques para realizar los planes]
	[F1] + [Programa] + [Enter]
[CONSULTA]	
[Selec.Req.]	
	[F2] + [Selec.Req] + [Enter]
[Compras Ant.Prov]	
	[F2] + [Compras Ant. Prov] + [Enter]
[Compras Ant. Mtri]	
	[F2] + [Compras Ant. Mtri] + [Enter]
[Compras Ant Corp]	
	[F2] + [Compras Ant. Corp] + [Enter]
[Lista de Precios]	
	[F2] + [Listas de Precios] + [Enter]
[POSICION]	
[Siguiente]	[Desde la zona de despliegue muestra las partidas que por falta de espacio no son visibles en la primer pantalla]
	[F3] + [Siguiente] + [Enter]
[Anteriores]	[Una vez que estamos visualizando la segunda pantalla de despliegue podremos regresar a la primera con esta opción]
	[F3] + [Anteriores] + [Enter]



[Primeras]	[En algunas ocasiones existiran partidas suficientes para varias pantallas y donde vamos a desear ver despues de visualizar algunas de ellas la primera , se hará con este botón.] [F3] + [Primeras] + [Enter]
[Ultimas]	[Al igual que la opción anterior vamos a desear en alguna ocasión ver la ultima pantalla sin pasar por las anteriores de esta manera nuestro botón de Ultimas será util] [F3] + [Ultimas] + [Enter]
[V]	[Botón que permite una visualización dentro de el área de despliegue hacia abajo una por una de las partidas] [F4] + [Enter]
[^]	[Botón que permite la visualización de las partidas hacia arriba de una por una en la zona de despliegue] [F4] + [enter]
[V]	
[Close]	[Nos permite salir de la Pantalla o limpiar los campos de la misma según como nosotros escojamos]. [F8] + [Close] + [Enter]
[Limpia]	[Limpia cada uno de los campos de la forma dejandola lista para otra consulta de partidas]
[Salida]	[Nos permitirá dejar la Pantalla, regresando a la Pantalla de Actualización de Pedidos]

TECLA [F3]

PANTALLA DE NOTAS



DESCRIPCION DE CAMPOS

- 1.- Narración [Es el área más amplia de la pantalla, puesto que en ella se despliega la descripción de la nota.]
- 2.- Nota [Clave de Catálogo de la nota acceso con las teclas [Ctrl] + [V] + [Enter]]
- 3.- Fecha [Fecha de alta de la nota]

DESCRIPCION DE AREA DE MENU

- [Actualiza]
[Alta] [Cuando no existe la nota en sistema con esta opción podemos adicionarla]
[Captura Datos]+[F1]+[Alta]+[Enter].
- [Cambio] [Modificaciones a notas existentes]
[Modif. datos]+[F1]+[Cambio]+[Enter]
- [Elim.] [En el caso de no ser necesaria alguna de las notas del sistema con esta opción la damos de baja]
[Nota existente]+[F1]+[Elim]+[Enter]
- [Catálogo Notas] [Opción necesaria para acceder las notas que se encuentran residentes en sistema; en este caso es cuando podemos utilizar la facilidad del [Ctrl]+[V] mencionado anteriormente para consultar las claves de notas]
[F2] + [Enter]



[?]

Opcional desde campo notas [Ctrl] + [V]
[Ayuda proporciona información adicional de cada uno de los campos de la Pantalla dependiendo de la posición del cursor]

[/]

[F7] + [Enter] desde cualquier campo
[Salida es la tecla que nos permite salir de la presente Pantalla y nos posiciona en la Pantalla de Requisiciones]
[F8]

TECLA [F4]

PANTALLA DOMICILIO A FACTURAR

```
.....UON10110.....
 1 2 ..... 0 1
1 Pedido RF 6122 8
2 Tipo (1.-Factura 2.-Consigna) 1.....
3 Clave: 1
4 Domicilio
5 Telefono 6 Fax
.....
```

En esta pantalla acentamos el domicilio a donde deseamos que se Facture el pedido en cuestión.

DESCRIPCION DE CAMPOS

1.- Pedido

[Cve. de centro, requisición y anexo que aparecen automáticamente en la Pantalla al accesarla, cabe aclarar que dichos datos son los mismos que el usuario capturo y sobre los que se han hecho todas las consultas y modificaciones posibles dentro de este Módulo]

2.- Tipo

[Este campo identifica y define la naturaleza del domicilio mediante una clave; la clave de Domicilio Facturar es "1"]

3.- Clave

[Campo que identifica la relación del domicilio con el almacén correspondiente]



4.- Domicilio

[Campo que contiene la dirección completa del Domicilio a Facturar]

5.- Teléfono

[No. telefónico del domicilio]

6.- Fax

[No. de fax del domicilio]

DESCRIPCION DE AREA DE MENU

TECLA [F8]

[/]

[Al igual que lo hemos venido haciendo la tecla F8 nos permite salir de la Pantalla]

[F8] + [Close] + [Enter]



MANUAL SISTEMA DE ABASTECIMIENTO SEGMENTO MOVIMIENTOS IMPORTACION PANTALLA PRINCIPAL LOGISTICA Y ABASTECIMIENTO

```
.....LOGISTICA Y ABASTECIMIENTO.....
| 1. EL MÓDULO - Tráfico | Infraestructura | 99 | 11912501 | |
| 1 | Act./Cons. | - | 1 | Requisición |
| 2 | Conflicto | - | 1 | Policiación |
| 3 | Casos | - | 1 | Inspección |
| 4 | Reportes de Pedidos | - | 1 | Pedidologer |
| 5 | Reportes de Requisiciones | - | 1 | Pedido Importar |
| 6 | | | | Pedido Import |
| 7 | | | | Pedido Import |
| 8 | | | | Pedido Import |
| 9 | | | | Pedido Import |
| 0 | | | | Pedido Import |
.....
```



Como en todos los segmentos anteriores daremos comienzo desde la Pantalla de Logística y Abastecimiento, desde donde seguiremos los siguientes pasos para acceder al Segmento que explicaremos en esta parte del Manual.

[F1]

[Act./Cons.]

[Enter]

[Movimto:Import]

[Enter]

PANTALLA DE TRAFICOS

```
.....Tráficos.....
| 1 Agente Advenel | CIGR. ALEJANDRO | CHAVIS. DATA | ..... | |
| 2 No. de Tráfico | 0922 | Tipo Operación | IMPORTACION DEFINITIVA |
| 3 Compuce | ..... |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 0 | | | | |
.....
```

Description	Status	Fecha	Description	Ultima Realidad	Fecha
1) Frontera	Completada	Jan 2 1993	Frontera	Completada	Jan 2 1993
2) Arribo	Requiere	Jan 13 1993	Arribo	Requiere	Jan 13 1993
3) Expediente	Requiere	Jan 13 1993	Expediente	Requiere	Jan 13 1993
4) Expediente	Requiere	Jan 13 1993	Expediente	Requiere	Jan 13 1993
5) Expediente	Requiere	Jan 13 1993	Expediente	Requiere	Jan 13 1993
6) Expediente	Requiere	Jan 13 1993	Expediente	Requiere	Jan 13 1993
7) Expediente	Requiere	Jan 13 1993	Expediente	Requiere	Jan 13 1993
8) Expediente	Requiere	Jan 13 1993	Expediente	Requiere	Jan 13 1993
9) Expediente	Requiere	Jan 13 1993	Expediente	Requiere	Jan 13 1993
0) Expediente	Requiere	Jan 13 1993	Expediente	Requiere	Jan 13 1993

```
.....
```



DESCRIPCION DE CAMPOS

- 1.- Agente Aduanal
[Captura de la cve. y nombre de agente aduanal para acceder la Lista de Códigos de Identificación del Sistema de digitará:
[Ctrl] + [V];
En el momento de que aparezca el Cuadro de Diálogo se pide realizar las siguientes funciones para que se asuma el dato en la Pantalla Principal.
[Posicionarse en cve. deseada]+[Enter]]
- 2.- No. de Tráfico
[Dato que el usuario debe proporcionar al Sistema puesto que es dependiente total del flujo del Proceso y su consecutivo según control físico del material]
- 3.- Tipo de Operación
[Captura de dato según procedimiento de captura directa o teclas de consulta; para asumir el dato deseado.
[Ctrl]+[V]
{Posición dato deseado}
[Enter]
- 4.- Compañía
[Dato que aparece en Pantalla automáticamente dependiente de la cve. de identificación del tráfico]
- 5.- Status
[Se muestra un área de trabajo donde se requiere actualizar el status del tráfico según los status de modalidades de cada una de sus partidas. Mayor información sobre status ver Anexo 1]
- 6.- Ultima Modalidad
[Area de trabajo donde se actualizará el status de la última modalidad de el tráfico]



DESCRIPCION AREA DE MENU

- [Act] [Actualiza proporciona la posibilidad de activar las funciones dentro de el Segmento de Movimiento Importación]
[F1]
- [Alta] [Ingresará la información capturada en los campos de la Pantalla de Tráfico en el Sistema, cuando no existan]
[F1] + [Alta] + [Enter]
- [Baja] [Cuando no sean necesarios los datos de cierto tráfico existente en Sistema es conveniente depurarlos y con esta opción desaparecerán automáticamente]
[F1] + [Baja] + [Enter]
- [Cambio] [Cuando realizamos modificaciones a nuestra información de Sistema dependiente a un tráfico en cuestión debemos accesar esta función después de realizar dicha modificación para concretar la actualización de información].
[F1] + [Cambio] + [Enter]
- [Consulta] [Botón que aparece activo en el momento de accesar la Pantalla y nos permitirá visualizar los datos de cada uno de los tráficos existentes en Sistema organizados según el diseño de la Pantalla; este botón es útil cuando no sabemos o recordamos algún dato del tráfico que requerimos]
[F2] + [Enter]
- [Mod] [El botón de Modalidad presenta un Cuadro de Diálogo donde se programan las fechas de modalidad ya sea por medio de captura o haciendo uso de las teclas de función disponibles para este Cuadro:
[F1] Implantará la fecha del día corriente
[F6] Dejará en blanco toda el área de asignación de fechas de cada uno de los status

[F8] Salida de el Cuadro de diálogo]

[Mov]	[Tecla que contiene 4 de las formas de trabajo de esta parte del Sistema]
[Movtos Material] [Facturas]	[Subpantalla de donde encontraremos enlistadas todas las facturas que se han emitido para el Proveedor asignado del Tráfico; se dará una explicación más amplia en la sección de Subpantallas]
[Bultos]	[Subpantalla donde se proporcionará las observaciones pertinentes sobre el tipo de material y sus especificaciones; ver sección de Subpantallas]
[Gastos de Mercado]	[Ver nota anexa]
[T. Ant]	[Nos permitirá la visualización de los tráficos si existen que esten ubicados en sistema antes que el que estamos consultando]
[T. Sig.]	[Visualización de los tráficos si existen después del tráfico que estamos consultando]
[Fecha Hoy]	[Nos imprime la fecha del día corriente en el momento de realización del trámite del Tráfico]
[/]	[Como lo hemos venido haciendo desde esta tecla tenemos dos opciones: [Otro tráfico] Opción que deja todos los campos de la Forma en blanco listos para consultar otro tráfico. [Salida] Para salir de la Pantalla automáticamente



SECCION DE SUBPANTALLAS DE LA FORMA DE TRAFICO

SUBPANTALLA DE FACTURAS

```
.....FACTURAS del Pedido.....
# Facturas
# Numero 142.6688 Mon Dtl Fecha Jan 2 1992
# Alm. Pedido
# Mon. Y. Com.
# .....
10 # Pedido 142.6688 Dtl Jan 2 1992
# .....
FACTURA IMPORTE MON FECHA ALM PEDIDO ANEXO SI COMPRADOR ZONA
```

DESCRIPCION DE CAMPOS

- | | |
|---------------------|--|
| 1.- Número | [No. de identificación de la factura expedida] |
| 2.- Importe | [Cantidad a la que ascendió la factura] |
| 3.- Mon | [Cve. identificación de la moneda en la que fué emitida la factura] |
| 4.- Fecha | [La fecha de emisión de la factura] |
| 5.- Alm. | [Cve. identificación del almacén que expidió la factura] |
| 6.- Pedido | [No. de identificación del Pedido a cargo del cual se expidió la factura] |
| 7.- Anexo | [No. de anexo del pedido anterior] |
| 8.- SI | [Cve. de status del Pedido en cuestión] |
| 9.- Comprador | [No. de identificación del comprador responsable del Pedido] |
| 10- Zona de trabajo | [Compuesta por ocho campos en blanco mediante los cuales podremos modificar la información de la forma; se digita el no. de factura que deseamos modificar y automáticamente aparecerán en los siete restantes campos la información correspondiente lista para ser modificada y que posteriormente se dará de alta en sistema; opción de Cambio de la tecla [F1]] |



DESCRIPCION AREA DE MENU

- [Actualiza]
[Alta] [Botón con tres opciones del Sistema]
[Cuando la factura no existe en Sistema podremos capturar la información requerida y con esta opción la ingresaremos permanentemente]
[F1] + [Alta] + [Enter]
- [Baja] [Al considerar innecesaria la información de alguna factura podemos asignarla en la zona de trabajo y con esta opción la suprimiremos de Sistema].
[F1] + [Baja] + [Enter]
- [Cambio] [Como ya se mencionó en la sección anterior existe la posibilidad de cambiar la información referente a las facturas y con esta opción cerramos el ciclo de modificación de la información al digitar en la zona de trabajo el no. de factura apareciendo los datos concernientes a dicha factura posteriormente seguir los siguientes pasos:
[Posicionarse campo modificable]
[Ctrl] + [E] [campo limpio]
[Captura nueva información]
[F1] + [Cambio] + [Enter]
- (f) [En esta subpantalla también desde este botón accedemos dos opciones:
[Limpia] Campos en blanco de la zona de trabajo listos para consultar otra factura]
[Salir] Salida de la Subpantalla regresando a la Pantalla de Tráfico.]

SUBPANTALLA DE BULTOS

Bultos			
Actualiza			
No.	Cantidad	Observaciones	U.S.
1	1,000.0000	rate de una esquina	000.0000 X0
.....			
1 1 1:000.0000 rate de una esquina 000.0000 X0			

[F1] [Limpia] [Salir]



DESCRIPCION DE CAMPOS

1.- No.	[Número de identificación de la partida en cuestión del tráfico]
2.- Cantidad	[Cantidad de material surtido en esa partida]
3.- Observaciones	[Descripción de el material abastecido]
4.- Tp	[Tipo de operación realizada para el abastecimiento]
5.- Peso	[Se refiere a el volumen del material]
6.- U.M.	[Unidad de Medida en la que se calculó el peso de el material]
7.- Zona de trabajo	[Consta de 6 campos en blanco los cuales corresponden cada uno de ellos a los campos anteriores; donde mediante la captura de el no. de partida en el primer campo en blanco automáticamente en los cinco campos restantes aparecerá los datos concernientes a la partida seleccionada y de esta manera podremos realizar cualquier consulta o modificación de información].

DESCRIPCION DEL AREA DE MENU

[Actualiza]	[Tres opciones pueden ser accedidas desde este botón]
[Alta]	[Si la partida abastecida aún no ha sido cargada en Sistema, proporcionando los datos requeridos en la Zona de trabajo y después digitando las teclas siguientes se incluire la información permanentemente; [F1] + [Alta] + [Ente]
[Baja]	[Cuando la Información de una partida abastecida ya no es necesaria es conveniente depurarla y lo realizaremos mediante los siguientes pasos: [Digitar no. partida zona trabajo] [F1]



[Baja]

[Enter]

[Cambio]

[Para llevar a cabo algún cambio en la información de bullos realizaremos las siguientes funciones:

[Posición campo modificable]

[Ctrl] + [E] [Campo limpio]

[Captura nueva información]

[F1] + [Cambio] + [Enter]

[/]

[Botón desde donde realizaremos las siguientes funciones:

[Limpia] Después de teclear esta opción los campos quedarán en blanco listos para la siguiente consulta.

[Salida] Salida de la Pantalla de Bullos regresando a la Pantalla de Tráficos.]

SUBPANTALLA DE MOVIMIENTOS DE MATERIAL

```

*****MOVIMIENTOS DE MATERIAL*****
1  X  ANALISIS DE PARTIDAS CONSULTAS BULLOS FACILIDAD
2  0
3  Tipo Movimiento 11 Fecha del 1 1993 12
4  SP. Abonos 12
5  Transporte 13 Dura 10 Cond. File 14
6  Semblante 15 Comprob 10
7  Fact fecha Inporte non bul Cont. 16
8  11 12
9
10 RIM
11 Profesor
12 Part description Cont. IV Ant. Acum Cont. ST
13
14
15
16

```



DESCRIPCION DE CAMPOS

- 1.- Agl. Aduanal [Nombre y cve. del agente aduanal que aparecerá en el momento de desplegarse la Pantalla]
- 2.- Tráfico [No. identificación del Tráfico que también aparecerá en el momento de desplegarse la Pantalla]
- 3.- Tipo [Captura del tipo de movimiento, compendio de tipos de movimiento que se accesa y asume en la pantalla mediante los siguientes pasos:
[Ctrl] + [V]
[Selección de cve. deseada] [Enter]
- En este momento se carga en todos los campos que estaban en blanco de la Pantalla la información correspondiente si es que existe en Sistema]
- 4.- Movimiento [El movimiento es un número de identificación del movimiento de material el cual será proporcionado por el Sistema cuando se capture el tipo de Movimiento si es que existe especificaciones correspondientes o se capturará según especificaciones del Tráfico para darlo de alta]
- 5.- Fecha de Movimiento [Representa la fecha de realización del Movimiento]
- 6.- Transporte [Especificación de el medio de Transporte del Movimiento de Material, cuando no existe la información se debe capturar para darse de alta]
- 7.- Guía [No. de identificación de la guía de transporte]
- 8.- Cond. Flete [Cve. de condición de flete; Cuadro de diálogo para consultar las diferentes claves para este campo y así poder asumir una de ellas mediante los siguientes pasos:
[Ctrl] + [V]
[Selección de clave] [Enter]
- En este campo tambien se puede capturar la cve. directamente si ya lo



- conocemos]
- 9.- Remitente [Nombre del Proveedor; capturable o asumible por sistema a partir de el tipo de operación]
- 10.- Compañía [Cve. de la Cía solicitante]
- 11.- Zona despliegue Factura [En esta área se desplegarán la información correspondiente a las facturas pertenecientes a el movimiento en cuestión, tales como No. de factura, Fecha de expedición de la misma, Importe y moneda en que se pacto]
- 12.- Zona despliegue Bulto [Despliegue de información correspondiente a los bultos abastecidos para este movimiento; los campos serán Bulto, Cantidad de Material surtida, Tipo de Movimiento, Peso y Unidad de Medida]
- 13.- Especificación Pedido [Datos pertenecientes al Pedido tales como; Almacén, Pedido, Anexo, Status del Pedido, Ultima Fecha de Status del Pedido, No. de identificación del Comprador y Nombre del Almacén]
- 14.- Proveedor [Nombre del Proveedor asignado]
- 15.- Especificación Partidas [Información concerniente a las partidas del pedido y su abastecimiento; los campos son: Partida, Descripción, Cantidad, Envío Anterior, Acumulado, Cantidad y Status. Los tres primeros campos son la requisición original del pedido, en el campo [Envío Anterior] se alojará información perteneciente al abastecimiento en un movimiento anterior, [Acumulado] tendrá la cifra total de material abastecido, [Cantidad] y [Status] que pertenece al ultimo status del pedido.
- 16.- Zona de Trabajo [Donde existen dos campos en blanco para poder modificar la información de la Especificación de Partidas, especificado más adelante]



SUBPANTALLA DE EMBARQUES PROGRAMADOS

DESCRIPCION DE CAMPOS

- 1.- Pedido [Especificación del Pedido con los datos de almacén, requisición y anexo]
- 2.- Partida [Contiene la cantidad de partidas habilitadas en el momento de abrir el cuadro]
- 3.- Zona despliegue [Despliegue de programa de entregas, donde se especifica:
.- Partida en cuestión
.- Cantidad a surtir en cada evento
.- Cantidad Surtida en Evento Anterior
.- Acumulado de material abastecido
.- Cantidad
.- Status de cada una de las partidas]
- 4.- Zona de Trabajo [Zona donde digitando el no. de partida se podrá proseguir con la captura de programa de abastecimiento del Tráfico para realizar la función se digitará:
[Partida en primer campo Zona Trabajo]
[Cuadro Embarques Programados]
[Captura Partida 1er. campo]
[Cantidad Material 2o.campo]
[/] [accepta el programa]

DESCRIPCION DEL AREA DE MENU

- [Baja] [Botón para depurar el Sistema de información de Programa de Partidas que ya no es útil o confiable y se realizará de la siguiente manera:
[Especif. Partida 1er Zona de trabajo]
[F1]
[Enter]
- [Cancela] [Con este botón cancelaremos todo el programa de partidas]
[F2]
[Enter]
- [/] [Salida de el Cuadro de diálogo regresando a la Subpantalla de Movimientos de Material]
[F8]
[Enter]



DESCRIPCION AREA DE MENU SUBPANTALLA DE MOVIMIENTOS DE MATERIAL

[Mvto]	[Botón que proporciona tres diferentes funciones de la Pantalla]
[Alta]	[Cuando un movimiento de material no existe en Sistema con esta opción después de la captura lo ingresaremos permanentemente] [Captura de información] [F1] + [Alta] + [Enter]
[Baja]	[Al considerarse no necesario un Movimiento de Material existente en Sistema se borrará de Sistema mediante esta opción] [Movimiento en Pantalla] [F1] + [Baja] + [Enter]
[Cambio]	[Al presentarse la necesidad de modificar la información existente en Sistema correspondiente a un Movimiento de Material la podremos modificar mediante esta opción] [Movimiento en Pantalla] [Posición en campo modificable] [Ctrl] + [V] [Campo limpio] [Captura nva. información] [F1] + [Cambio] + [Enter]
[Partida]	[Tres Posibilidades correspondientes al área de Partidas de la Pantalla]
[Todas]	[Nos permite la visualización y acumulación de todas las partidas del movimiento en cuestión] [F2] + [Toda] + [Enter]
[Cancela]	[Al seleccionar una partida en la Zona de Trabajo de la Forma y decidimos que no deseamos realizar ninguna función, accedamos esta función para cancelar la selección de partida] [F2] + [Cancela] + [Enter]
[Baja]	[Nos permite dar de baja la partida seleccionada en la Zona de Trabajo] [F2] + [Baja] + [Enter]
[Consultas]	[Realizaremos tres tipos de consulta desde este botón]



- [Pedidos x Mov] [Cuadro de diálogo que proporciona información de consulta donde encontramos los siguientes campos:
- Cve. Almacén
 - No. identificación del Pedido
 - No. identificación del Anexo
 - Status del Pedido asociado al Mov.
 - No. cve. de Comprador
- y en el área de Menu encontramos:
- [/] [Proporciona dos opciones]
- [Acepta] [Para asumir la información en la Pantalla de Movimientos de Material después de haber seleccionado uno de los Pedidos].
- [Cancela] [Opción para salir del Cuadro de diálogo]
- Ver anexo de cuadros de diálogo
- [F3] + [Pedidos x Mov] + [Enter]
- [Mov. x Pedido] [Cuadro de diálogo para identificar los movimientos asociados al Pedido, donde encontramos los siguientes campos:
- Cve. Agente Aduanal
 - No. del Tráfico en cuestión
 - Tipo de movimiento
 - Consecutivo numérico del Pedido
- En el área de Menú encontramos:
- [/] [Botón con dos opciones]
- [Acepta] [Para asignar la información correspondiente a un Movimiento seleccionado en la Pantalla de Movimientos de Material]
- [Cancela] [Nos permite salir del Cuadro de Diálogo]
- [F3] + [Movimientos x Pedido] + [Enter]
- [Bultos] [Subpantalla donde encontramos la información correspondiente a la cantidad de material abastecido explicada a detalle en la sección del Área de Menú de la Forma de Tráfico]
- [F4] + [Enter]



[Facturas]	[Subpantalla que contiene la información relativa a las facturas correspondientes a ya expedidas para ese Proveedor; esta Subpantalla fué explicada a detalle en el área de Menú de la Forma de Tráfico] [F5] + [Enter]
[/]	[Botón que nos permite elegir entre cuatro opciones]
[Otro Tráfico]	[Limpia los campos donde se despliega la información correspondiente al tráfico y los deja listos para cargar ya sea por sistema o captura una nva. especificación de tráfico] [F8] + [Otro Tráfico] + [Enter]
[Otro Movimiento]	[Limpia los campos donde se despliega la información correspondiente al movimiento de material y los deja listos para cargar ya sea por sistema o captura una nva. especificación de movimiento] [F8] + [Otro Movimiento] + [Enter]
[Otro Pedido]	[Limpia los campos donde se despliega la información correspondiente al pedido y los deja listos para cargar ya sea por sistema o captura una nva. especificación de movimiento] . [F8] + [Otro Pedido] + [Enter]
[Salida]	[Nos permite dejar la Subpantalla de Movimientos de Material regresando a la Pantalla de Tráfico] [F8] + [Salida] + [Enter]

Nota: La opción de Gastos de Mercado contenida en el botón de Movimientos en el área de Menu de la Pantalla de Tráfico no se documentará por estar en desarrollo en el momento de elaboración de el Manual.

Para el Segmento de Movimiento de Material Nacional se utiliza la misma Pantalla de Movimientos de Material tomando en cuenta como requisito el consultar la tabla de Tipos de Movimiento y capturando cualquiera de los dos tipos de Movimiento en campo de de Tipo para que a partir de el mismo se complete la información en la Pantalla; ver anexo 1.



MODULO

DE

INFRAESTRUCTURA



MANUAL SISTEMA DE ABASTECIMIENTO MODULO DE INFRAESTRUCTURA

PANTALLA PRINCIPAL LOGISTICA Y ABASTECIMIENTO

```
.....LOGISTICA Y ABASTECIMIENTO.....
F1 Compras  F2 Traficos  F3 Infraestructura: 0
SELECCIONE UNA OPCION
Proveedores
Equipos
Compras
Materiales
Otras
.....
JUL 1 1995 2:46:28AM
```



El Módulo de Infraestructura es el que nos permitirá Consultar y Modificar los Datos existentes en Sistema de las áreas que se consideran básicas para el funcionamiento del mismo y a partir de las cuales girará nuestro Proceso de Abastecimiento.

PANTALLA DE PROVEEDORES

```
.....PROVEEDORES.....
F2 Actualizo  Domicilio: Linea: 03
R.F.C. del Proveedor
Razon Social
Nacionalidad (M,J,L)
Vie de notificacion
Tiempo de entrega
Condiciones de entrega
Cue. condicion pago
Dias pronto pago
Representa a (R.F.C.)
Modalidad de pago
Dias de pago
Lugar de entrega
Descuento general
Dias condicion de pago
Descuento pronto pago
```



Con la Pantalla de Proveedores el usuario será libre del manejo de la Información existente en Sistema por medio de Consultas o de la inclusión de los datos de un nuevo Proveedor cuando así lo exija sus necesidades previa la captura de los datos en los campos correspondientes de la Pantalla y accedando la Zona de Menu [F1] [Actualiza]; cabe aclarar que los botones [F2] [Domicilio] y [F3] [Linea] contienen información sobre el Domicilio del Proveedor en cuestión y de las líneas de materiales que suministra.

PANTALLA DE REQUISITORES

```

.....LOGISTICA Y ABASTECIMIENTO.....
/1 COMPRAS TRAFICO INGRESOS ..... 214612841
SELECCIONE UMR OPCIO REQUISITORES -> DIVISION
1 CONGRESOS -> COMPANIA
MATERIALES -> ALMAC
OTROS -> CENTRO
  
```

Esta opción se compone de cuatro subpantallas donde se Consulta, Ingresa o Modifica la información perteneciente a los posibles entes que de alguna manera son objeto de requisiciones de material.

DIVISION

```

.....DIVISION.....
/1 COMPRAS /1 ACTUALIZA DONDE/12 CIES/RJ001 LIMITE/ 03
SELECCION DIVISION NOMBRE DE DIVISION
11 CORPORATIVO
  
```

En la presente Pantalla podremos Consultar las Divisiones existentes del Grupo y para visualizar sus claves digitar:

[Ctrl] + [V] + [Posición en División deseada] + [Enter]

```

.....Llaves Listas.....
/1 COMPRAS /1 ACTUALIZA REQUIR CANCEL/ 03
SELECCION DIVISION NOMB DIV_nombre
1 CORPORATIVO
CORPORATIVO
INDUSTRIAS
INDUSTRIAS
INDUSTRIAS
OPERAIONES INGENIERIA Y CONSTRUCCIONES
SAL HOLDINGS
  
```


En este caso se sigue el flujo de dependencia de Compañía - Unidad dentro de la forma pidiendo que se elija una Compañía para que con [Ctrl] + [V] el sistema arroje información al propósito de las Unidades dependientes de ella; y los posibles Centros o Almacenes que existen,

En el área de menú se aloja nuestro acostumbrado botón de [Actualiza] para realizar los cambios o inclusiones necesarias de información, también nos encontramos con el botón [Centros] para que de esta manera al digitarlo nos da una lista de los Centros.

CENTROS

```

.....LOGISTICA Y ABASTECIMIENTO.....
#1 Compras Traficos Infraestructura
#2 Reclutamiento Personal
#3 Centros
#4 Almacenes
#5
#6
#7
#8
#9
#10
#11
#12
#13
#14
#15
#16
#17
#18
#19
#20
#21
#22
#23
#24
#25
#26
#27
#28
#29
#30
#31
#32
#33
#34
#35
#36
#37
#38
#39
#40
#41
#42
#43
#44
#45
#46
#47
#48
#49
#50
#51
#52
#53
#54
#55
#56
#57
#58
#59
#60
#61
#62
#63
#64
#65
#66
#67
#68
#69
#70
#71
#72
#73
#74
#75
#76
#77
#78
#79
#80
#81
#82
#83
#84
#85
#86
#87
#88
#89
#90
#91
#92
#93
#94
#95
#96
#97
#98
#99
#100
#101
#102
#103
#104
#105
#106
#107
#108
#109
#110
#111
#112
#113
#114
#115
#116
#117
#118
#119
#120
#121
#122
#123
#124
#125
#126
#127
#128
#129
#130
#131
#132
#133
#134
#135
#136
#137
#138
#139
#140
#141
#142
#143
#144
#145
#146
#147
#148
#149
#150
#151
#152
#153
#154
#155
#156
#157
#158
#159
#160
#161
#162
#163
#164
#165
#166
#167
#168
#169
#170
#171
#172
#173
#174
#175
#176
#177
#178
#179
#180
#181
#182
#183
#184
#185
#186
#187
#188
#189
#190
#191
#192
#193
#194
#195
#196
#197
#198
#199
#200
#201
#202
#203
#204
#205
#206
#207
#208
#209
#210
#211
#212
#213
#214
#215
#216
#217
#218
#219
#220
#221
#222
#223
#224
#225
#226
#227
#228
#229
#230
#231
#232
#233
#234
#235
#236
#237
#238
#239
#240
#241
#242
#243
#244
#245
#246
#247
#248
#249
#250
#251
#252
#253
#254
#255
#256
#257
#258
#259
#260
#261
#262
#263
#264
#265
#266
#267
#268
#269
#270
#271
#272
#273
#274
#275
#276
#277
#278
#279
#280
#281
#282
#283
#284
#285
#286
#287
#288
#289
#290
#291
#292
#293
#294
#295
#296
#297
#298
#299
#300
#301
#302
#303
#304
#305
#306
#307
#308
#309
#310
#311
#312
#313
#314
#315
#316
#317
#318
#319
#320
#321
#322
#323
#324
#325
#326
#327
#328
#329
#330
#331
#332
#333
#334
#335
#336
#337
#338
#339
#340
#341
#342
#343
#344
#345
#346
#347
#348
#349
#350
#351
#352
#353
#354
#355
#356
#357
#358
#359
#360
#361
#362
#363
#364
#365
#366
#367
#368
#369
#370
#371
#372
#373
#374
#375
#376
#377
#378
#379
#380
#381
#382
#383
#384
#385
#386
#387
#388
#389
#390
#391
#392
#393
#394
#395
#396
#397
#398
#399
#400
#401
#402
#403
#404
#405
#406
#407
#408
#409
#410
#411
#412
#413
#414
#415
#416
#417
#418
#419
#420
#421
#422
#423
#424
#425
#426
#427
#428
#429
#430
#431
#432
#433
#434
#435
#436
#437
#438
#439
#440
#441
#442
#443
#444
#445
#446
#447
#448
#449
#450
#451
#452
#453
#454
#455
#456
#457
#458
#459
#460
#461
#462
#463
#464
#465
#466
#467
#468
#469
#470
#471
#472
#473
#474
#475
#476
#477
#478
#479
#480
#481
#482
#483
#484
#485
#486
#487
#488
#489
#490
#491
#492
#493
#494
#495
#496
#497
#498
#499
#500
#501
#502
#503
#504
#505
#506
#507
#508
#509
#510
#511
#512
#513
#514
#515
#516
#517
#518
#519
#520
#521
#522
#523
#524
#525
#526
#527
#528
#529
#530
#531
#532
#533
#534
#535
#536
#537
#538
#539
#540
#541
#542
#543
#544
#545
#546
#547
#548
#549
#550
#551
#552
#553
#554
#555
#556
#557
#558
#559
#560
#561
#562
#563
#564
#565
#566
#567
#568
#569
#570
#571
#572
#573
#574
#575
#576
#577
#578
#579
#580
#581
#582
#583
#584
#585
#586
#587
#588
#589
#590
#591
#592
#593
#594
#595
#596
#597
#598
#599
#600
#601
#602
#603
#604
#605
#606
#607
#608
#609
#610
#611
#612
#613
#614
#615
#616
#617
#618
#619
#620
#621
#622
#623
#624
#625
#626
#627
#628
#629
#630
#631
#632
#633
#634
#635
#636
#637
#638
#639
#640
#641
#642
#643
#644
#645
#646
#647
#648
#649
#650
#651
#652
#653
#654
#655
#656
#657
#658
#659
#660
#661
#662
#663
#664
#665
#666
#667
#668
#669
#670
#671
#672
#673
#674
#675
#676
#677
#678
#679
#680
#681
#682
#683
#684
#685
#686
#687
#688
#689
#690
#691
#692
#693
#694
#695
#696
#697
#698
#699
#700
#701
#702
#703
#704
#705
#706
#707
#708
#709
#710
#711
#712
#713
#714
#715
#716
#717
#718
#719
#720
#721
#722
#723
#724
#725
#726
#727
#728
#729
#730
#731
#732
#733
#734
#735
#736
#737
#738
#739
#740
#741
#742
#743
#744
#745
#746
#747
#748
#749
#750
#751
#752
#753
#754
#755
#756
#757
#758
#759
#760
#761
#762
#763
#764
#765
#766
#767
#768
#769
#770
#771
#772
#773
#774
#775
#776
#777
#778
#779
#780
#781
#782
#783
#784
#785
#786
#787
#788
#789
#790
#791
#792
#793
#794
#795
#796
#797
#798
#799
#800
#801
#802
#803
#804
#805
#806
#807
#808
#809
#810
#811
#812
#813
#814
#815
#816
#817
#818
#819
#820
#821
#822
#823
#824
#825
#826
#827
#828
#829
#830
#831
#832
#833
#834
#835
#836
#837
#838
#839
#840
#841
#842
#843
#844
#845
#846
#847
#848
#849
#850
#851
#852
#853
#854
#855
#856
#857
#858
#859
#860
#861
#862
#863
#864
#865
#866
#867
#868
#869
#870
#871
#872
#873
#874
#875
#876
#877
#878
#879
#880
#881
#882
#883
#884
#885
#886
#887
#888
#889
#890
#891
#892
#893
#894
#895
#896
#897
#898
#899
#900
#901
#902
#903
#904
#905
#906
#907
#908
#909
#910
#911
#912
#913
#914
#915
#916
#917
#918
#919
#920
#921
#922
#923
#924
#925
#926
#927
#928
#929
#930
#931
#932
#933
#934
#935
#936
#937
#938
#939
#940
#941
#942
#943
#944
#945
#946
#947
#948
#949
#950
#951
#952
#953
#954
#955
#956
#957
#958
#959
#960
#961
#962
#963
#964
#965
#966
#967
#968
#969
#970
#971
#972
#973
#974
#975
#976
#977
#978
#979
#980
#981
#982
#983
#984
#985
#986
#987
#988
#989
#990
#991
#992
#993
#994
#995
#996
#997
#998
#999
#1000
#1001
#1002
#1003
#1004
#1005
#1006
#1007
#1008
#1009
#1010
#1011
#1012
#1013
#1014
#1015
#1016
#1017
#1018
#1019
#1020
#1021
#1022
#1023
#1024
#1025
#1026
#1027
#1028
#1029
#1030
#1031
#1032
#1033
#1034
#1035
#1036
#1037
#1038
#1039
#1040
#1041
#1042
#1043
#1044
#1045
#1046
#1047
#1048
#1049
#1050
#1051
#1052
#1053
#1054
#1055
#1056
#1057
#1058
#1059
#1060
#1061
#1062
#1063
#1064
#1065
#1066
#1067
#1068
#1069
#1070
#1071
#1072
#1073
#1074
#1075
#1076
#1077
#1078
#1079
#1080
#1081
#1082
#1083
#1084
#1085
#1086
#1087
#1088
#1089
#1090
#1091
#1092
#1093
#1094
#1095
#1096
#1097
#1098
#1099
#1100
#1101
#1102
#1103
#1104
#1105
#1106
#1107
#1108
#1109
#1110
#1111
#1112
#1113
#1114
#1115
#1116
#1117
#1118
#1119
#1120
#1121
#1122
#1123
#1124
#1125
#1126
#1127
#1128
#1129
#1130
#1131
#1132
#1133
#1134
#1135
#1136
#1137
#1138
#1139
#1140
#1141
#1142
#1143
#1144
#1145
#1146
#1147
#1148
#1149
#1150
#1151
#1152
#1153
#1154
#1155
#1156
#1157
#1158
#1159
#1160
#1161
#1162
#1163
#1164
#1165
#1166
#1167
#1168
#1169
#1170
#1171
#1172
#1173
#1174
#1175
#1176
#1177
#1178
#1179
#1180
#1181
#1182
#1183
#1184
#1185
#1186
#1187
#1188
#1189
#1190
#1191
#1192
#1193
#1194
#1195
#1196
#1197
#1198
#1199
#1200
#1201
#1202
#1203
#1204
#1205
#1206
#1207
#1208
#1209
#1210
#1211
#1212
#1213
#1214
#1215
#1216
#1217
#1218
#1219
#1220
#1221
#1222
#1223
#1224
#1225
#1226
#1227
#1228
#1229
#1230
#1231
#1232
#1233
#1234
#1235
#1236
#1237
#1238
#1239
#1240
#1241
#1242
#1243
#1244
#1245
#1246
#1247
#1248
#1249
#1250
#1251
#1252
#1253
#1254
#1255
#1256
#1257
#1258
#1259
#1260
#1261
#1262
#1263
#1264
#1265
#1266
#1267
#1268
#1269
#1270
#1271
#1272
#1273
#1274
#1275
#1276
#1277
#1278
#1279
#1280
#1281
#1282
#1283
#1284
#1285
#1286
#1287
#1288
#1289
#1290
#1291
#1292
#1293
#1294
#1295
#1296
#1297
#1298
#1299
#1300
#1301
#1302
#1303
#1304
#1305
#1306
#1307
#1308
#1309
#1310
#1311
#1312
#1313
#1314
#1315
#1316
#1317
#1318
#1319
#1320
#1321
#1322
#1323
#1324
#1325
#1326
#1327
#1328
#1329
#1330
#1331
#1332
#1333
#1334
#1335
#1336
#1337
#1338
#1339
#1340
#1341
#1342
#1343
#1344
#1345
#1346
#1347
#1348
#1349
#1350
#1351
#1352
#1353
#1354
#1355
#1356
#1357
#1358
#1359
#1360
#1361
#1362
#1363
#1364
#1365
#1366
#1367
#1368
#1369
#1370
#1371
#1372
#1373
#1374
#1375
#1376
#1377
#1378
#1379
#1380
#1381
#1382
#1383
#1384
#1385
#1386
#1387
#1388
#1389
#1390
#1391
#1392
#1393
#1394
#1395
#1396
#1397
#1398
#1399
#1400
#1401
#1402
#1403
#1404
#1405
#1406
#1407
#1408
#1409
#1410
#1411
#1412
#1413
#1414
#1415
#1416
#1417
#1418
#1419
#1420
#1421
#1422
#1423
#1424
#1425
#1426
#1427
#1428
#1429
#1430
#1431
#1432
#1433
#1434
#1435
#1436
#1437
#1438
#1439
#1440
#1441
#1442
#1443
#1444
#1445
#1446
#1447
#1448
#1449
#1450
#1451
#1452
#1453
#1454
#1455
#1456
#1457
#1458
#1459
#1460
#1461
#1462
#1463
#1464
#1465
#1466
#1467
#1468
#1469
#1470
#1471
#1472
#1473
#1474
#1475
#1476
#1477
#1478
#1479
#1480
#1481
#1482
#1483
#1484
#1485
#1486
#1487
#1488
#1489
#1490
#1491
#1492
#1493
#1494
#1495
#1496
#1497
#1498
#1499
#1500
#1501
#1502
#1503
#1504
#1505
#1506
#1507
#1508
#1509
#1510
#1511
#1512
#1513
#1514
#1515
#1516
#1517
#1518
#1519
#1520
#1521
#1522
#1523
#1524
#1525
#1526
#1527
#1528
#1529
#1530
#1531
#1532
#1533
#1534
#1535
#1536
#1537
#1538
#1539
#1540
#1541
#1542
#1543
#1544
#1545
#1546
#1547
#1548
#1549
#1550
#1551
#1552
#1553
#1554
#1555
#1556
#1557
#1558
#1559
#1560
#1561
#1562
#1563
#1564
#1565
#1566
#1567
#1568
#1569
#1570
#1571
#1572
#1573
#1574
#1575
#1576
#1577
#1578
#1579
#1580
#1581
#1582
#1583
#1584
#1585
#1586
#1587
#1588
#1589
#1590
#1591
#1592
#1593
#1594
#1595
#1596
#1597
#1598
#1599
#1600
#1601
#1602
#1603
#1604
#1605
#1606
#1607
#1608
#1609
#1610
#1611
#1612
#1613
#1614
#1615
#1616
#1617
#1618
#1619
#1620
#1621
#1622
#1623
#1624
#1625
#1626
#1627
#1628
#1629
#1630
#1631
#1632
#1633
#1634
#1635
#1636
#1637
#1638
#1639
#1640
#1641
#1642
#1643
#1644
#1645
#1646
#1647
#1648
#1649
#1650
#1651
#1652
#1653
#1654
#1655
#1656
#1657
#1658
#1659
#1660
#1661
#1662
#1663
#1664
#1665
#1666
#1667
#1668
#1669
#1670
#1671
#1672
#1673
#1674
#1675
#1676
#1677
#1678
#1679
#1680
#1681
#1682
#1683
#1684
#1685
#1686
#1687
#1688
#1689
#1690
#1691
#1692
#1693
#1694
#1695
#1696
#1697
#1698
#1699
#1700
#1701
#1702
#1703
#1704
#1705
#1706
#1707
#1708
#1709
#1710
#1711
#1712
#1713
#1714
#1715
#1716
#1717
#1718
#1719
#1720
#1721
#1722
#1723
#1724
#1725
#1726
#1727
#1728
#1729
#1730
#1731
#1732
#1733
#1734
#1735
#1736
#1737
#1738
#1739
#1740
#1741
#1742
#1743
#1744
#1745
#1746
#1747
#1748
#1749
#1750
#1751
#1752
#1753
#1754
#1755
#1756
#1757
#1758
#1759
#1760
#1761
#1762
#1763
#1764
#1765
#1766
#1767
#1768
#1769
#1770
#1771
#1772
#1773
#1774
#1775
#1776
#1777
#1778
#1779
#1780
#1781
#1782
#1783
#1784
#1785
#1786
#1787
#1788
#1789
#1790
#1791
#1792
#1793
#1794
#1795
#1796
#1797
#1798
#1799
#1800
#1801
#1802
#1803
#1804
#1805
#1806
#1807
#1808
#1809
#1810
#1811
#1812
#1813
#1814
#1815
#1816
#1817
#1818
#1819
#1820
#1821
#1822
#1823
#1824
#1825
#1826
#1827
#1828
#1829
#1830
#1831
#1832
#1833
#1834
#1835
#1836
#1837
#1838
#1839
#1840
#1841
#1842
#1843
#1844
#1845
#1846
#1847
#1848
#1849
#1850
#1851
#1852
#1853
#1854
#1855
#1856
#1857
#1858
#1859
#1860
#1861
#1862
#1863
#1864
#1865
#1866
#1867
#1868
#1869
#1870
#1871
#1872
#1873
#1874
#1875
#1876
#1877
#1878
#1879
#1880
#1881
#1882
#1883
#1884
#1885
#1886
#1887
#1888
#1889
#1890
#1891
#1892
#1893
#1894
#1895
#1896
#1897
#1898
#1899
#1900
#1901
#1902
#1903
#1904
#1905
#1906
#1907
#1908
#1909
#1910
#1911
#1912
#1913
#1914
#1915
#1916
#1917
#1918
#1919
#1920
#1921
#1922
#1923
#1924
#1925
#1926
#1927
#1928
#1929
#1930
#1931
#1932
#1933
#1934
#1935
#1936
#1937
#1938
#1939
#1940
#1941
#1942
#1943
#1944
#1945
#1946
#1947
#1948
#1949
#1950
#1951
#1952
#1953
#1954
#1955
#1956
#1957
#1958
#1959
#1960
#1961
#1962
#1963
#1964
#1965
#1966
#1967
#1968
#1969
#1970
#1971
#1972
#1973
#1974
#1975
#1976
#1977
#1978
#1979
#1980
#1981
#1982
#1983
#1984
#1985
#1986
#1987
#1988
#1989
#1990
#1991
#1992
#1993
#1994
#1995
#1996
#1997
#1998
#1999
#2000
#2001
#2002
#2003
#2004
#2005
#2006
#2007
#2008
#2009
#2010
#2011
#2012
#2013
#2014
#2015
#2016
#2017
#2018
#2019
#2020
#2021
#2022
#2023
#2024
#2025
#2026
#2027
#2028
#2029
#2030
#2031
#2032
#2033
#2034
#2035
#2036
#2037
#2038
#2039
#2040
#2041
#2042
#2043
#2044
#2045
#2046
#2047
#2048
#2049
#2050
#2051
#2052
#2053
#2054
#2055
#2056
#2057
#2058
#2059
#2060
#2061
#2062
#2063
#2064
#2065
#2066
#2067
#2068
#2069
#2070
#2071
#2072
#2073
#2074
#2075
#2076
#2077
#2078
#2079
#2080
#2081
#2082
#2083
#2084
#2085
#2086
#2087
#2088
#2089
#2090
#2091
#2092
#2093
#2094
#2095
#2096
#2097
#2098
#2099
#2100
#2101
#2102
#2103
#2104
#2105
#2106
#2107
#2108
#2109
#2110
#2111
#2112
#2113
#2114
#2115
#2116
#2117
#2118
#2119
#2120
#2121
#2122
#2123
#2124
#2125
#2126
#2127
#2128
#2129
#2130
#2131
#2132
#2133
#2134
#2135
#2136
#2137
#2138
#2139
#2140
#2141
#2142
#2143
#2144
#2145
#2146
#2147
#2148
#2149
#2150
#2151
#2152
#2153
#2154
#2155
#2156
#2157
#2158
#2159
#2160
#2161
#2162
#2163
#2164
#2165
#2166
#2167
#2168
#2169
#2170
#2171
#2172
#2173
#2174
#2175
#2176
#2177
#2178
#2179
#2180
#2181
#2182
#2183
#2184
#2185
#2186
#2187
#2188
#2189
#219
```


Pantalla que contiene todos los datos de cada comprador dependiendo de la cina analizada.

En el área de menu se encuentran los botones de [Actualización] para modificar la información existente, [Lineas] líneas de material que existen a requisitar, [Usuarios] los posibles usuarios a quienes se les debe dar atención

PANTALLA DE MATERIALES

```

.....LOGISTICA Y ADQUISICION.....
#1 COMPRAS Trafico Infraestructura.....0
#
SELECCIONE UNA OPCION
Requisitores ->
Compradores ->
Materiales ->
Otros ->
.....
Cod. Almacen
Cod. Corporal
Unidad de
Linea/Sublinea
.....
  
```

Esta Pantalla nos proporciona cuatro subpantallas de consulta de información a propósito de Materiales.

COD. ALMACEN

```

.....CODIGO DE MATERIAL.....
#1 Actualiza consulta material Corp1 Lin1o1 El.....0
.....
Centro APT MET-MEX PENOLAS
Codigo de Material
Unidad de medida
Activo fijo
Descripcion Material
.....
  
```

Subpantalla con el objetivo de identificar el material por su código con respecto a cada almacén o centro, requiriendose datos como unidad de medida y descripción del material además de identificación si es Activo Fijo o no.

Area de Menu contiene la opción de [Actualizar] la información, [Consultas de datos dependientes a este segmento], [Material Corporativo] identificación del material



COND. MATERIAL

```
.....CODIGO
#1 ASOCIACION ASOC/COD#01 UnidMed 01 0
.....
Linea 71 Equipo de Laboratorio
Sublinea
Codigo
Unidad Medida
Descripcion
.....
```

Actualiza [F1] RespCod [F2] [F3] [F4] [F5] [F6] [F7] [F8] [F9] [F10] [F11] [F12] [F13] [F14] [F15] [F16] [F17] [F18] [F19] [F20]

Pantalla con especificaciones del tipo de material a abastecer proporcionando la línea sublínea, peso y descripción el código es una clave que se obtendrá mediante la tecla de Menu Asoc/Cod.

Area de menú, [Actualización] para modificar la información existente, [Asoc./Cod] para conocer la cve de los Códigos]

UNIDAD/MEDIDA

```
.....LOGISTICA Y ABASTECIMIENTO
#1 Compras Tráfico Infraestructura
.....
SELECCIONE UNA OPCION (F1, F2...)
.....Unidades de medida.....
#1 ASOCIACION LIND#01
.....
Clave de la Unidad (Esp) |
Unid.Operar(es) (Esp) |
Clave de la Unidad (Ing) |
Unid.Operar(es) (Ing) |
Descripcion |
.....
```

Actualiza [F1] [F2] [F3] [F4] [F5] [F6] [F7] [F8] [F9] [F10] [F11] [F12] [F13] [F14] [F15] [F16] [F17] [F18] [F19] [F20]

Pantalla que contiene los datos relativos a unidades de medida y claves de unidad en idioma español e inglés para emplearse en los procesos del Sistema además de una descripción.

Area e menú solo existe la posibilidad de modificar nuestra información mediante el botón de [Actualiza].



LINES/SUBLINEA

```
*****ABASTECIMINTO*****
#1 Compras Trafico Abastecimiento
*****
SEL: 1 2 3 4 5 6 7 8 9 0
*****
Linea      -
Sublinea
Descripcion
```



Subpantalla donde se consultaran o modificarán las líneas o sublíneas de material, al igual que su descripción.

Area de menu contiene tres teclas de función activa, [Actualización] para cualquier modificación, [Sublínea] se accesa cuando no se tiene el conocimiento de las sublíneas para la línea designada, [Comprador] lista de compradores autorizados para este tipo de material.

PANTALLA DE OTROS CONCEPTOS

```
*****ABASTECIMINTO*****
#1 Compras Trafico Abastecimiento
*****
SELECCIONE UNA OPCION
*****
Promotoras -> JUL 1 1993 6:53:14AM
Compradores ->
Categorías ->
Otras ->
*****
Tipo de Cambio
Categoría de Compras
Categoría de Mater
Categoría de Tablas
*****
```



Pantalla para consultar y modificar cinco conceptos adicionales del Sistema de Abastecimiento.



CATALOGO DE NOTAS

```
.....NOTES.....
1 21 Notulas 0
  Narracion      Nota      Fecha
.....
[OPCIONES] [Volver] [Ingresar] [.....] [.....] [.....] [.....] [SALIR]
```

Subpantalla para visualizar los datos del Catálogo de Notas donde se nos permite modificar la información existente o adicionar la que se juzgue necesaria.

DOMICILIO

```
.....LOGISTICA Y REABASTECIMIENTO.....
1 21 Condor Trafico Ingresos 2
  SELECCIONE UNA OPCION (F1, F2, ...)
  21 Notulas Lindis 0
  Cve Domicilio: 0000 W
  Domicilio: .....
  Telefono: .....
  Fax: .....
.....
[OPCIONES] [Volver] [Ingresar] [.....] [.....] [PA] [SALIR]
```

Subpantalla para consultar los diferentes domicilios del Catálogo dependiendo de su clave; en esta pantalla también se pueden realizar modificaciones y altas de domicilios.



CATALOGO DE TABLAS

Control de tablas	
Actualiza	Ciencianos
Numero	Nombre de la tabla
	Condiciones de pago
	Condiciones de entrega
	Medida
	Uso de conservación
	Lugar de entrega
	Costos Reales de Mercado
	Valores
	Condiciones de Flete en Recepción
	Tipos de operación (Buy/Sell)
	Uso de documentos de Material
	Uso de notificaciones



Subpantalla donde encontraremos una relación entre el no. cvc. de identificación de las diferentes tablas y el nombre que describen su función, recalcando que estos campos pueden ser modificados como en cualquier otra Pantalla con la tecla de función [Actualiza].

Con la explicación de este Módulo damos por finalizado el Manual recordando que como se menciona en la parte introductoria del Manual, está sujeto a cambios y mejoras que en su momento después de documentadas de esta misma manera se les hará llegar las Actualizaciones.

ANEXO 1

TIPOS DE MOVIMIENTOS DE MATERIAL

MOVIMIENTOS NACIONALES

CLAVE	DESCRIPCION
"EM"	EMBARQUE DE MATERIAL
"RA"	RECIBO EN ALMACEN

MOVIMIENTOS INTERNACIONALES

CLAVE	DESCRIPCION
"RF"	RECIBO EN FRONTERA
"RX"	REEXPEDIDO