

27
2ef



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

**SISTEMA INTEGRAL DE CONTABILIDAD DE UNA CASA
POSTPRODUCTORA DE CINE Y VIDEO**

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A N
MARIO CARRASCO CARRASCO
EDGAR ALEJANDRO GARCIA ADAME
BLANCA LUCIA GIL CASTELLANOS

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

MEXICO, D. F.

DICIEMBRE DE 1998

257765



**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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 esta 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.

A ti Papá, estés donde estés, se que lo esperabas.

A ti Mamá por tu presencia.

A mis hermanos por su constante apoyo....

..... Y a ti, por siempre estar.

Mario.

A mis padres, a mis hermanos y a sus hijos.

Gracias a Dios, a Gus y a los Beatles.

Blanca.

A mis padres y mis hermanos, gracias por su apoyo.

Gracias a Dios.

Edgar.

CONTENIDO

	Objetivo	V
	Introducción	VII
Capítulo 1	Conceptos	
1.1	Metodología de Diseño y Desarrollo de Sistemas.....	1
1.1.1	Ciclo de Vida de los Sistemas.....	1
1.1.2	Conceptos y técnicas de Modelado.....	9
1.1.3	Metodología de Yourdon.....	21
1.1.4	Metodología Ssadm.....	24
1.1.5	Metodología Métrica Versión 2.....	27
1.1.6	Metodología Merise.....	31
1.1.7	Metodología de Case*Method.....	33
1.2	Bases de Datos.....	39
1.2.1	Conceptos.....	39
1.2.2	Modelo Jerárquico o de Árbol.....	42
1.2.3	Modelo de Red.....	44
1.2.4	Modelo Relacional.....	46
1.3	Redes.....	62
1.3.1	Definición.....	62
1.3.2	Tipos de Redes.....	64
1.3.3	Modelos de Red.....	67
1.3.4	Topologías de Redes Locales.....	73
1.3.5	Medios de Comunicación.....	78
1.3.6	Elementos de Interconexión.....	85
1.3.7	Protocolos.....	87
1.3.8	Tendencias de las Redes.....	91

1.4	Arquitectura Cliente Servidor.....	92
1.4.1	Definición.....	92
1.4.2	Sistemas Tradicionales y Sistemas Cliente Servidor.....	93
1.4.3	Arquitectura Tres Capas.....	95
1.4.4	Características de la Arquitectura Cliente Servidor.....	99
1.4.5	Tipos de Servidores.....	102
1.4.6	Middleware (Infraestructura Cliente Servidor).....	104
1.4.7	Monitor de Transacciones.....	107
1.4.8	Características de Algunos Monitores de Transacciones.....	110
1.5	Herramientas de Desarrollo.....	118
1.5.1	Herramientas de Apoyo para el Desarrollo de Sistemas	118
1.5.2	CASE.....	126
1.5.3	Upper CASE, Lower CASE	131
1.5.4	Análisis Y Selección del Software para Generar el Front-End.....	133
1.6	Contabilidad.....	146
1.6.1	Conceptos Básicos Contables.....	146
1.6.2	Áreas de Automatización en el Proceso Contable.....	165
Capítulo 2	Planteamiento del Problema y Propuesta de Solución	
2.1	Casa Postproductora de Cine y Vídeo.....	169
2.1.2	Descripción del Proceso Actual.....	174
2.2.	Problemática Actual.....	181
2.2.1	Áreas de Interés.....	181
2.2.2	Consecuencias del Problema.....	184

2.3	Requerimientos del Usuario.....	189
2.3.1	Requerimientos Generales	189
2.3.2	Requerimientos Particulares.....	191
2.3.3	Recopilación y Clasificación de Información.....	194
2.3.4	Plan de Trabajo.....	196
2.4	Estrategias de Solución.....	206
2.5	Opciones de Solución.....	213
2.6	Diagrama de Funcionalidad.....	219
2.6.1	Arquitectura Lógica Propuesta.....	219
2.6.2	Arquitectura Física Propuesta.....	221
2.6.3	Requerimientos Mínimos del Sistema.....	223
Capítulo 3	Diseño e Implementación del Sistema	
3.1	Especificaciones.....	227
3.2	Diseño y Construcción de la Base de Datos.....	238
3.2.1	Diagrama de Flujo de Datos (DFD).....	239
3.2.2	Diagrama Funcional.....	262
3.2.3	Diagrama Entidad Relación.....	277
3.2.4	Normalización.....	297
3.2.5	Matriz Entidad vs. Funciones.....	308
3.2.6	Diccionario de Datos.....	314
3.3	Desarrollo de Pantallas.....	364
3.3.1	Diseño y Construcción de Pantallas.....	364
3.3.2	Desarrollo de Reportes del Sistema.....	391
3.3.3	Depuración, Respaldo y Recuperación de Información.....	437

Capítulo 4	Pruebas del Sistema	
4.1	El proceso de Pruebas.....	441
4.1.1	Pruebas Unitarias.....	441
4.1.2	Pruebas de Integración.....	447
4.1.3	Pruebas de Aceptación.....	457
4.2	Plan de Pruebas del Sistema.....	460
4.2.1	Objetivos de las Pruebas.....	460
4.2.2	Alcances.....	461
4.2.3	Tabla de Roles y Responsabilidades.....	463
4.2.4	Lista de Programas.....	464
4.2.5	Lista de Pantallas.....	465
4.2.6	Lista de Reportes.....	467
4.2.7	Criterios de Terminación, Rechazo y Reanudación de Pruebas.....	469
4.2.8	Mecanismo de Seguimiento de Errores.....	469
4.2.9	Calendario de Actividades de Pruebas.....	470
4.2.10	Scripts de Pruebas.....	475
4.2.11	Reporte de Errores.....	478
4.2.12	Documento de Aceptación del Usuario.....	483
	Conclusiones.....	485
	Bibliografía.....	487
	Apéndice A.....	491
	Apéndice B.....	507

OBJETIVO

Elaborar un sistema que sustituya los procedimientos manuales contables de una Casa Postproductora de Cine y Video.

Aplicar una metodología de desarrollo de sistemas para resolver la problemática planteada.

INTRODUCCIÓN

Una de las necesidades que encontramos con mayor frecuencia en el trabajo diario de cualquier empresa o negocio, hoy en día, es sin duda, el control de la información que se origina de cada una de las transacciones comerciales que en dicho negocio se llevan a cabo.

El control fiel, actualizado y organizado de la información inherente a estos procesos comerciales, podemos englobarlo en lo que propiamente conocemos como Contabilidad.

Sin duda, contar con un sistema automático que nos permita ingresar, consultar, actualizar y reportar la información que nace de la parte administrativa de una empresa, resulta de primordial interés, sobretodo en los niveles de gerencia, pues es con dicha información que se obtienen los elementos necesarios para tomar decisiones, realizar planeaciones, especular, y concebir planes de contingencia y de opciones alternativas, para cada uno de los esquemas administrativos y financieros que la empresa se haya fijado como objetivos o metas comerciales.

Dicha información debe ser compartida por varios equipos de trabajo con funciones diferentes pero complementarias a fin de que juntos, logren los objetivos de la empresa. Compartir la información implica contar con los niveles de seguridad tales que verifiquen que sólo cierto personal podrá tener acceso a un conjunto de los datos de éstos, permitir operaciones adecuadas únicamente a la función que dicho personal debe realizar.

Este trabajo tiene como funcionalidad entregar un sistema que se encargue de llevar dicho control de la información contable, en una forma actualizada, clara, segura, compartiendo información, trabajando en tiempo real y entregando reportes y gráficas que sirvan a los intereses comerciales de una empresa en específico.

Este trabajo mostrará la situación actual en cuanto al manejo práctico, seguro y confiable de la información contable, los recursos físicos con los que se cuentan, y las necesidades principales que, a la entrega de este sistema, deben ser satisfechas.

Para esto, utilizamos una metodología de desarrollo de sistemas escogido entre varias, y mostramos las razones, ventajas y desventajas de nuestra elección. Siguiendo dicha metodología, y basándonos en el ciclo de vida en general de un sistema, presentamos nuestro *Análisis, Diseño, Construcción, Pruebas y Transición*, así como la solución final, esquemas, diagramas, entradas, salidas y su código.

Finalmente, hacemos hincapié en las conclusiones a las que este trabajo nos permitió llegar, los objetivos cumplidos, los conocimientos aplicados y la experiencia adquirida.

CAPITULO 1
CONCEPTOS

1. CONCEPTOS

1.1. METODOLOGÍA DE DISEÑO Y DESARROLLO DE SISTEMAS.

1.1.1. CICLO DE VIDA DE LOS SISTEMAS.

En términos generales, los sistemas de cómputo son desarrollados para resolver problemas relacionados al procesamiento de información, y en estos términos, pueden llegar a ser muy complejos. El objetivo del desarrollo de sistemas es limitar esta complejidad, al mismo tiempo entregar una solución real al problema expuesto. En este sentido, el problema representa el *requerimiento*, el cual debe ser *analizado* entregando una solución que será interpretada en el *diseño* junto con otras consideraciones relevantes.

Después, un equipo capacitado, interpretará el diseño, conceptual y detallado, realizando la *implementación* del sistema. Otro o tal vez el mismo, realizará la *evaluación* de esta implementación, ambas actividades para asegurar la funcionalidad del diseño y por ende, la solución del problema. Una vez hecho esto, continúa la *instalación*, la cual probablemente necesitará *soporte* y ocasionalmente *mantenimiento*, que junto a la *capacitación* de los usuarios, permite encontrar bugs (*errores de programación*) y nuevos requerimientos.

Estas etapas, junto con otras, se conocen como el **ciclo de vida de un sistema**.

El concepto de ciclo de vida de un sistema es fundamental para los métodos de la ingeniería de software. El propósito es conocer el proceso de la ingeniería de software y así poder monitorear y controlarlo. El modelo clásico del ciclo de vida es el "Modelo de Cascada". Sus fases se definen por sus actividades, entradas y salidas.

La ingeniería de software es una disciplina de la ciencia de la computación que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelva problemas. "Software" es el término usado para nombrar a los programas de las computadoras y toda la documentación asociada con ellos, como los manuales de usuario.

El ciclo está caracterizado en diferentes *etapas* o *fases*. Hay más de una forma de estructurar el ciclo de vida del software en fases. Una buena definición de una fase está compuesta por *actividades*, un conjunto de *entregables* que resultan de las actividades de estas fases en particular, y un conjunto aplicable de *medidas de control de calidad* para la fase. Además, un conjunto de información que forman la entrada (escrita o verbal) es necesaria para realizar las actividades de la fase. Las salidas (entregables) producidas por la fase pueden ser la entrada de las fases siguientes.

Las etapas más comúnmente usadas en el desarrollo del ciclo de vida del software son: **definición del problema**, el análisis de requerimientos, especificaciones, diseño, codificación, pruebas, operación y mantenimiento. Existen otras fases en el ciclo, todas en la categoría del modelo de cascada, es decir, el desarrollo del software secuencial desde la fase más alta o primera bajando hasta la más detallada. En la fig. 1.1.1.A. se muestra el ciclo que nace de las propuestas de Zelk, Gibs y Royc.

El modelo inicial del ciclo de vida del software se propuso en la década de los 70 y desde entonces el modelo ha tenido muchos perfeccionamientos y variaciones. Todos ellos pueden incluirse en un "macro" modelo del ciclo de vida. La fig. 1.1.1.B nos muestra el Macro modelo del ciclo de vida de los sistemas.

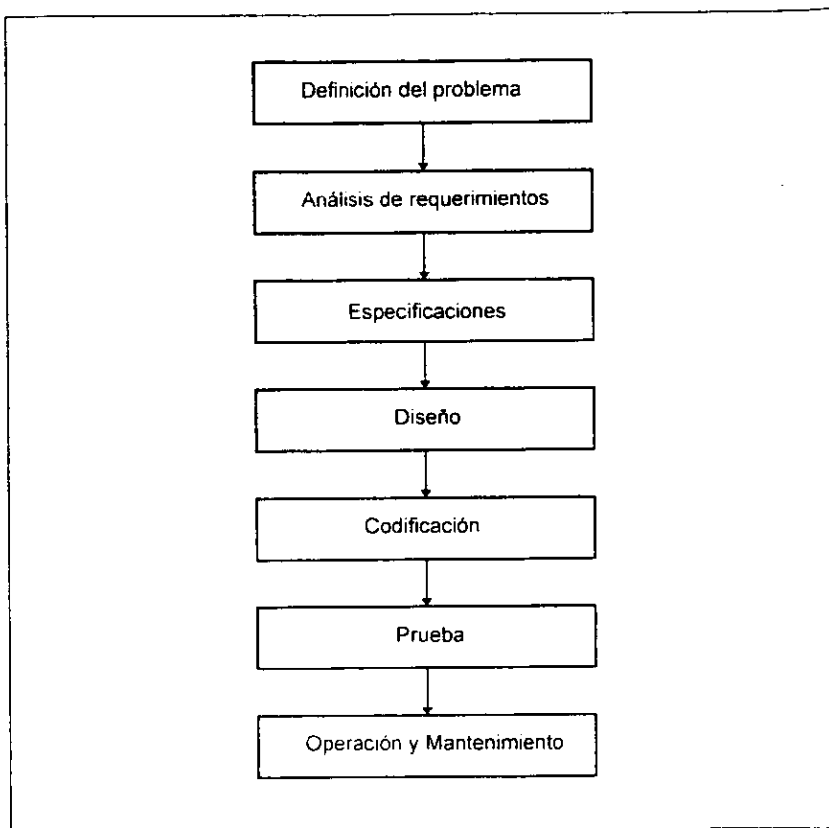


Fig. 1.1.1.A. Ciclo de vida del software.

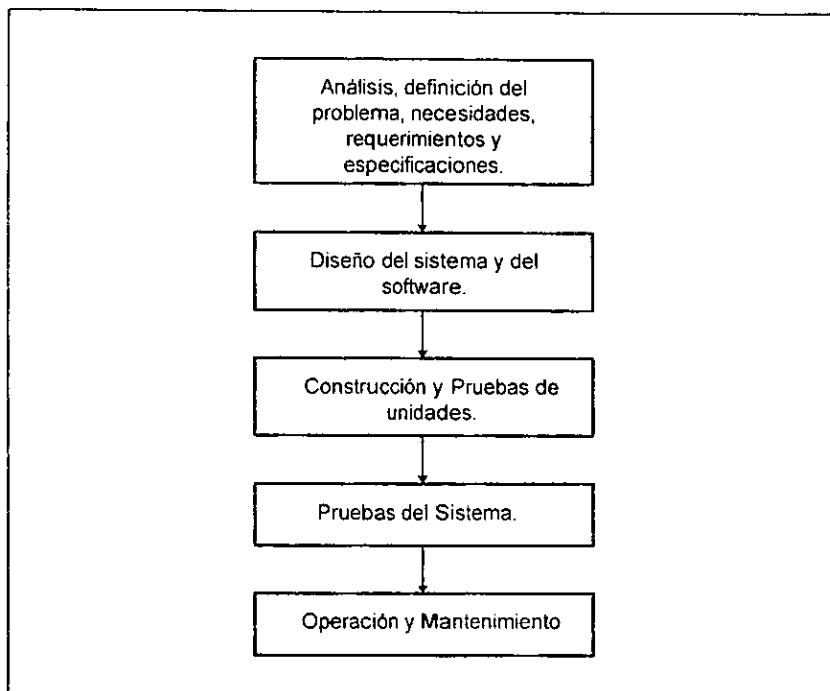


Fig. 1.1.1.B. Macro modelo del ciclo de vida.

1. Análisis, definición del problema, necesidades, requerimientos y especificaciones.

Consultando con los usuarios, se obtienen los servicios, restricciones y objetivos del sistema. Deben definirse de una manera comprensible, tanto para los usuarios como para el personal de sistemas. El problema se debe definir en los términos del usuario lo más preciso posible, la precisión es importante porque no podemos desarrollar una solución de un problema si no entendemos cuál es el problema, y sólo el usuario sabe lo que es o no es el problema. Los objetivos que se persiguen al analizar los requerimientos son: lograr un claro entendimiento entre el usuario y el desarrollador de lo que la solución debe realizar; llegar a acuerdos para definir los criterios para aceptar la solución; y llegar a un plan de trabajo detallando los

entregables de cada etapa. Al analizar las especificaciones se busca describir cómo será la solución: las entradas que el sistema procesa, cuáles funciones se realizarán para cada entrada, cuáles serán sus correspondientes salidas, y como se corresponden a las especificaciones antes señaladas.

2. **Diseño del sistema y del software.**

El diseño del sistema se divide en el diseño del software y del hardware. El diseño del software es el proceso de representar las funciones del sistema a fin de poder transformarlo con facilidad en uno o más programas de computación. El diseño del hardware es el proceso de representar la plataforma física en donde estará ejecutándose el sistema. Mientras que las especificaciones se concentran en el *qué* debe hacer el sistema, el diseño describe el *como* realizar la solución. El diseño estructura al sistema en módulos unidos, lógicos y funcionales; también selecciona las estructuras de datos y algoritmos para la implementación de los datos de entrada y salida y de las funciones del sistema. Los entregables de esta etapa son todos los diagramas y documentos del diseño, el diseño también debe ser verificado para asegurar la calidad con que se realizó.

3. **Construcción y pruebas de unidades.**

En esta etapa, el diseño se realiza como un conjunto de unidades de programación escritas en algún lenguaje ejecutable. Las pruebas implican la comprobación de que cada unidad cumpla con su especificación. El objetivo de la construcción es trasladar la solución en código. Idealmente, el código se escribe después de que el diseño terminó satisfactoriamente. La fase de construcción termina cuando todo el código está escrito y documentado, compilado, sin errores, siguiendo reglas y estándares de programación. Cada unidad de programación debe ser probada para asegurar que efectivamente realiza la función adecuada y sin errores.

4. **Pruebas del sistema.**

Es donde las unidades de programación se integran y se prueban como un sistema completo para asegurar que se cubren las necesidades iniciales. En este punto es

importante la presencia activa del usuario final. Tan pronto como el módulo ha sido escrito, debe ser probado esto es la *prueba de unidades de programación*. Después, los módulos se prueban en grupos para ver si interactúan adecuadamente, esto es la *prueba de integridad*. Necesitamos probar el sistema desarrollado en su plataforma de producción, es decir donde va a ejecutarse, esto es la *prueba del sistema*. Se recomienda probar el sistema en diferentes configuraciones y sistemas operativos, instalar el sistema y realizar la *prueba de instalación*. No todos los usuarios deben estar en las pruebas, es mejor iniciar con una prueba en un ambiente controlado con algunos usuarios y el equipo desarrollador, a esto se le conoce como la *prueba alfa*; una *prueba beta* libera el software a algunos usuarios "amigos" para encontrar problemas en ambientes reales. Una vez pasadas estas pruebas, puede, el sistema, entrar en operación.

5. **Operación y mantenimiento.**

Aquí se instala el sistema y se pone en uso práctico, suele ser la fase más larga del ciclo de vida. El mantenimiento implica corregir errores que no se descubrieron en las primeras etapas del ciclo de vida, mejorar la aplicación y aumentar los servicios del sistema a medida que se perciben nuevas necesidades. Después de que el sistema fue instalado exitosamente, puede ser puesto en operación. El mantenimiento detecta bugs procediendo al *mantenimiento correctivo*, adapta el software al nuevo ambiente y realiza las modificaciones del usuario, es decir el *mantenimiento de adaptación*, y mejora al sistema *mantenimiento perfectivo*. El mantenimiento asegura que las nuevas versiones del software no causen confusión y prevé nuevas adecuaciones o nuevas funcionalidades para el sistema. Esta fase es la más larga y termina cuando, por obsolescencia, se deshecha el sistema después de haber sido totalmente utilizado.

Existen diferentes metodologías que pueden ser utilizadas para el análisis y diseño de sistemas, algunas de ellas contemplan una parte del ciclo de vida del sistema, otras agrupan varias de ellas en una sola; pero todas coinciden en realizar, finalmente, un

producto desarrollado sistemáticamente alcanzando los niveles de calidad requeridos y necesarios. Así, lograr sistemas de mayor calidad mediante un diseño sistemático es el objetivo primordial de las metodologías de ingeniería de sistemas.

Algunos criterios que nos dan una idea de lo que se entiende por lograr la calidad de los sistemas son los siguientes:

- **Que el sistema funcione correctamente.** Esto es, que el sistema cumpla con los requerimientos que durante el ciclo de vida de este, fueron especificados por el usuario. Para esto, las metodologías deben proponer revisiones constantes con el usuario en cada fase de desarrollo, claras y completas, y no permitir pasar a la siguiente fase si no existe un mutuo acuerdo de funcionalidad. Así evitamos edificar sobre bases equívocas, que seguramente en un futuro afectarán la funcionalidad del sistema, trayendo consigo un retraso considerable que se traduce en pérdida de tiempo, dinero y credibilidad. Un sistema que no hace o hace parcialmente lo que se supone debe hacer, refleja que el diseñador tiene una escasa comprensión de las especificaciones requeridas del sistema.
- **Que el sistema este libre de errores.** En este punto en la revisión de la calidad, es fundamental contar con un buen nivel en el conocimiento y experiencia al aplicar la metodología de desarrollo del sistema, exige nuestras capacidades de análisis y conceptualización, de diseño y resolución de problemas y, en el caso de la construcción, del uso completo y profundo de la herramienta con la que se haya decidido realizar la codificación. El uso del idioma, especialmente escrito, tiene una gran importancia al ser el medio complementador de los diagramas que cada fases origine, pues en forma escrita se establecen las especificaciones exactas y se comunican entre el analista, el diseñador y los constructores o programadores que intervengan. Cumpliendo con estos puntos, el número de errores funcionales y de construcción se verán en su mejor caso anulados, en caso de los pocos existentes, deberán ser detectados en la fase de pruebas, evaluación y uso, corrigiéndose en la

fase de mantenimientos, en el entendimiento que serán entonces errores de forma y no de fondo.

- **Que el sistema funcione de la manera más eficiente.** Si el sistema funciona correctamente y no tiene errores, no es totalmente suficiente para ser de calidad alta, sólo si su análisis entregó la solución más eficiente, que se reflejará en el desempeño del sistema en puntos básicos como el tiempo de respuesta, los recursos utilizados en equipo, espacio en disco, uso adecuado de la red, personal involucrado en la producción, seguridad, medios de respaldo y depuración, uso de tecnologías actuales con posibilidades de crecimiento, ventaja general del costo-beneficio; entre otras. Todo esto da como resultado que el sistema no se vuelva obsoleto en poco tiempo, ni que la solución encontrada presente más problemas ulteriores a pesar de que cumpla la funcionalidad y no tenga errores.

Para llevar a cabo un sistema siguiendo los pasos del ciclo de vida de éste, existen varias metodologías que la ingeniería de software nos provee. De estas, que algunas de ellas serán expuestas en el siguiente punto, se escogió el Case*Method como metodología de desarrollo aplicado al problema que este trabajo presenta. Para Case*Method, el ciclo de vida de un sistema queda resumido y representado en la fig. 1.1.1.C.

- **Definición del problema:** la comprensión de las *funciones* del problema, su ambiente y elucidación de los requerimientos de su solución.
- **Diseño:** el cómo implementar una solución.
- **Implementación:** la construcción de la solución.
- **Evaluación:** las pruebas de unidad, integración, instalación y del sistema.
- **Instalación:** la integración de la solución en el ambiente funcional.
- **Uso:** la operación en el ambiente funcional, como se solicitó en un principio.
- **Mantenimiento o soporte:** la corrección y adecuación a las necesidades actuales (que pudieron haber cambiado durante el ciclo de la solución).

Las etapas de diseño, implementación y evaluación deben ser revisadas y algunas veces ejecutadas concurrentemente. De hecho, estas tres etapas se conocen colectivamente como el *proceso de desarrollo*.

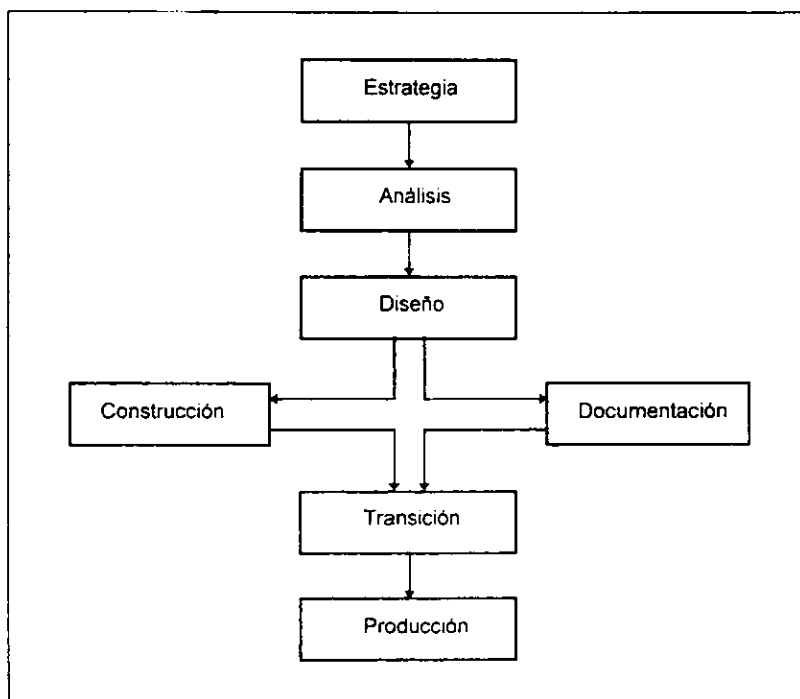


Fig. 1.1.1.C. Ciclo de vida de un sistema. Metodología Case*Method.

1.1.2. CONCEPTOS Y TÉCNICAS DE MODELADO.

Un **método** es la secuencia de pasos que ayuda a construir, a partir del original de la realidad, uno o varios modelos, con el objetivo de lograr un modelo material final o sistema que concrete la nueva propuesta de la realidad deseada en relación con el original. La representación abstracta del original es un **modelo**.

Un método es también un sistema de información, con un conjunto de *unidades* relacionadas entre sí y como una estructura de procedimientos que sintetizan un flujo

de *procesos* generador de resultados. Estas unidades describen las actividades, tareas, estados, funciones y productos, sus relaciones describen la estructura, los procedimientos, y los resultados de cada método.

La *estructura* del método se acopla al ciclo de desarrollo del sistema de información, organizándose en pasos (step-by-step). El ciclo de desarrollo tiene tres grandes grupos de fases: prejecutor (planificación, especificación, análisis), ejecutor (diseño, codificación, pruebas) y postejecutor (documentación, formación de usuarios, mantenimiento, rectificaciones); unos métodos no incluyen todas las fases en su estructura, y otros incluyen fases paralelas al desarrollo como documentación, calidad y configuración.

Las fases se subdividen en subfases (con distintos nombres, como módulos, etapas, actividades, etc.), hasta llegar a las *tareas*, actividades elementales realizadas en condiciones *atómicas* (un objetivo, un tipo de actor, un puesto). Cada tarea se apoya en un *procedimiento* que prescribe su forma de ejecución y es vehículo de comunicación entre los modeladores y los usuarios del sistema final. El procedimiento obtiene un *resultado*. El conjunto de resultados finales o *productos* (informes, diagramas, descripciones, código ejecutable), constituyen el sistema deseado.

Se han establecido un número de *técnicas* que son comunes a muchos métodos y a menudo se profundizan al margen de éstos. Estas técnicas son procedimientos para representar los requerimientos del sistema, con una técnica de diagramación, visual, compacta y no tan ambigua como el texto. Las *herramientas de CASE* informatizan las técnicas diagramáticas, algorítmicas y textuales: no sólo formalizan la documentación, sino que contienen los mecanismos rutinarios para verificar las consistencias de las técnicas soportadas. Según el alcance de las técnicas que cubra, la herramienta será común a varios métodos.

A continuación se detallarán algunas de las principales técnicas.

1.1.2.1. Diagrama de flujo de datos (DFD).

Esta técnica, también conocida por sus siglas DFD o por diagramas de burbujas (Bubble Diagrams), fue propuesta por Yourdon a finales de los años setenta. El objetivo fundamental es la descomposición de un problema complejo en otros más sencillos y manejables, facilitando la modularidad del sistema, además separa la estructura física de la lógica, aumentando la facilidad de mantenimiento de los análisis. Los elementos que lo constituyen son:

- **Procesos.** Son los *lugares* donde se transforma o descompone la información de un flujo de datos. Se representan con un círculo indicando su nombre dentro.
- **Entidades externas.** Son aquellas organizaciones, departamentos, personas o entidades en general que no pertenecen al sistema de información, pero que tiene relación con él. Es de donde proceden los datos que necesita el sistema o hacia donde van los generados por él. Sólo aparecen en el diagrama de contexto inicial. Se representan mediante un rectángulo con su nombre en el interior.
- **Almacén de datos.** Son aquellos *lugares* donde se almacena de forma estática la información dentro del sistema. Su nombre no indica cómo está organizado. Se representa gráficamente con dos líneas paralelas.
- **Flujo de datos.** Son las *tuberías* o *caminos* por donde fluye la información entre los diferentes elementos del sistema. Se representan a través de un arco orientado.

En la fig.1.1.2.A se observa un DFD según la notación de Yourdon.

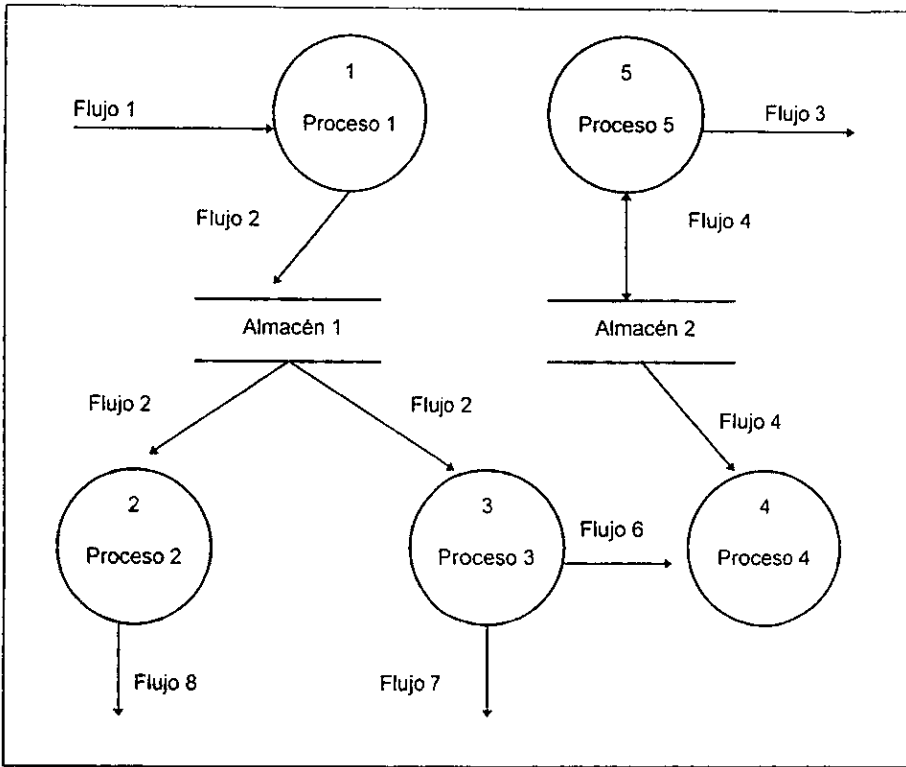


Fig.1.1.2.A Ejemplo de diagrama de flujo de datos.

1.1.2.2. Diagrama entidad - relación (DER).

El modelo entidad-relación es una de las técnicas más populares y utilizadas dentro del diseño de sistemas, concretamente en las bases de datos. Este modelo fue propuesto por Peter Chen en los comienzos de los años setenta, pero desde entonces han aparecido muchas variantes y extensiones.

En un DER se pueden identificar los siguientes conceptos: entidad, relación (o asociación), cardinalidad, atributos, ocurrencia, identificador, tipos de datos y, dependiendo del nivel de desarrollo de los modelos, se utilizarán otros conceptos como subtipos, supertipos, opcionalidad, dominios, etc. A continuación se presentan los elementos básicos del DER:

- **Entidad.** Es todo objeto del que se desea guardar información; es decir, se pretende *almacenar* una serie de datos para su posterior recuperación y tratamiento.
- **Relación.** Es básicamente, cualquier asociación o interdependencia entre varias entidades. En el ejemplo de la fig. 1.1.2.B, las entidades son *Cliente*, *Pedido*, *Línea*, *Producto*, *Tipo de cliente*, etc. Las relaciones son *solicita*, *contiene*, *hace referencia a*, etc.
- **Cardinalidad.** Son las reglas del negocio por las cuales las entidades existen en el modelo; es decir, definen la manera en número, de asociar entidades. En el ejemplo de la fig. 1.1.2.B, encontramos la cardinalidad que refleja que un cliente puede hacer muchos (M) pedidos y que un pedido tiene muchas líneas, y que una línea se relaciona con un producto.
- **Atributo.** Son las características o propiedades que describen a las entidades. Los atributos toman distintos valores. En algunos casos, hay atributos que se construyen por propia necesidad en el modelo para poder identificar valores de formas ordenadas. Ejemplos de este tipo de atributos son *código del cliente*, *dirección del cliente*, *nombre del cliente*, etc.

- **Ocurrencia.** En una entidad, una ocurrencia es un dato definido siguiendo el modelo diseñado, son los distintos valores que puede tomar, por ejemplo, la entidad Cliente como *Imagen* y *Producción*, etc.
- **Identificador.** Es un atributo que reúne una serie de condiciones y que tiene la utilidad de poder definir una ocurrencia de manera única. Un identificador debe tener un solo valor, que no se puede repetir y no puede ser nulo. En el caso de Cliente, existe el atributo de Código de cliente, el cual sería el candidato ideal para poder ser definido como identificador, y tomaría un valor distinto para cada cliente, y por ello sería único.
- **Tipos de datos.** Son la manera en la que los atributos se representarán de forma física en el diseño de las bases de datos. Estas formas físicas son propias del entorno sobre el cual se están desarrollando, y por lo tanto son una característica física. Ejemplos pueden ser los tipos numéricos, booleanos, reales, alfanuméricos, etc.

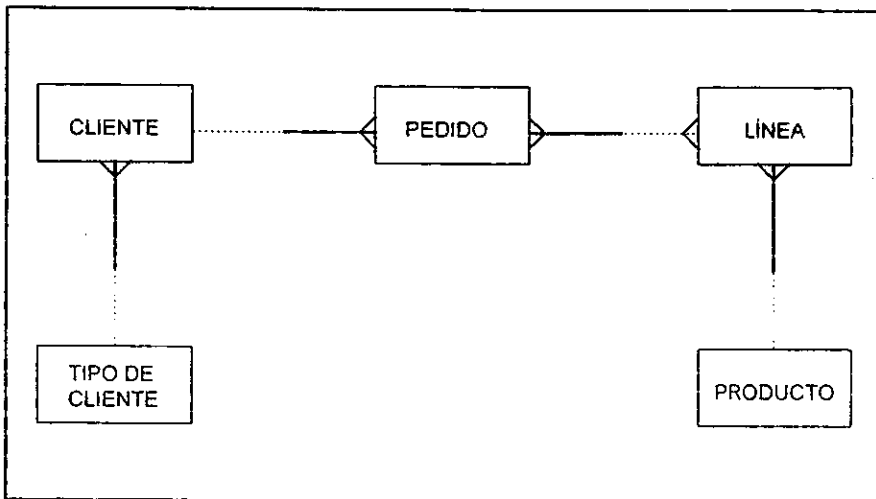


Fig. 1.1.2.B. Ejemplo de un Diagrama Entidad - Relación.

1.1.2.3. Diagrama de estructura (DE) o jerárquico funcional (DJF).

El diseño estructurado expone las guías de las que carece el diseño *top-down*, y es propuesta inicialmente por Yourdon y Constantine en los años setenta. Esta técnica es utilizada para representar la estructura funcional de un sistema, mostrando las interacciones que existen entre cada una de las funciones encontradas. Los elementos principales del diagrama de estructura se muestran en la fig. 1.1.2.C. y son:

- **Módulo.** Un módulo es un número de sentencias que hacen una actividad. El módulo se representa mediante una caja rectangular con su nombre dentro. El nombre del módulo es una sentencia de su función; es decir, lo que hace cada vez que es llamado. Un módulo predefinido es un módulo que no se tiene que escribir porque ya existe en el sistema, se representa añadiendo dos líneas paralelas e internas a ambos lados del rectángulo.
- **Conexión y comunicación entre módulos.** Un sistema está compuesto por módulos, organizados jerárquicamente, cooperando y comunicándose entre sí para realizar una tarea. Los signos de la comunicación son los datos y las banderas. Los datos se procesan y están relacionados con el problema; las banderas no se procesan, y sólo son importantes para la comunicación de información.
- **Cohesión.** Es la medida del grado de asociación de los elementos dentro de un módulo. Los elementos pueden ser una instrucción, un grupo de instrucciones o una llamada a otro módulo. Existen tres tipos de cohesión:
 - Cohesión funcional: todos los elementos contribuyen a una y sólo a una tarea.
 - Cohesión secuencial: los datos de salida de un elemento sirven como datos de entrada de otro elemento.
 - Cohesión comunicacional: los elementos contribuyen a tareas diferentes, pero cada tarea tiene los mismos parámetros de entrada y salida.

• **Acoplamiento.** Es la medida del grado de la interdependencia entre dos módulos. Existen tres tipos de acoplamiento:

- Acoplamiento por datos: dos módulos están acoplados por datos si se comunican por datos que no sean banderas, ni arreglos ni registros.
- Acoplamiento por estampilla: dos módulos están acoplados por estampilla si se comunican mediante registros o arreglos.
- Acoplamiento por control: dos módulos están acoplados por control si se comunican al menos por bandera.

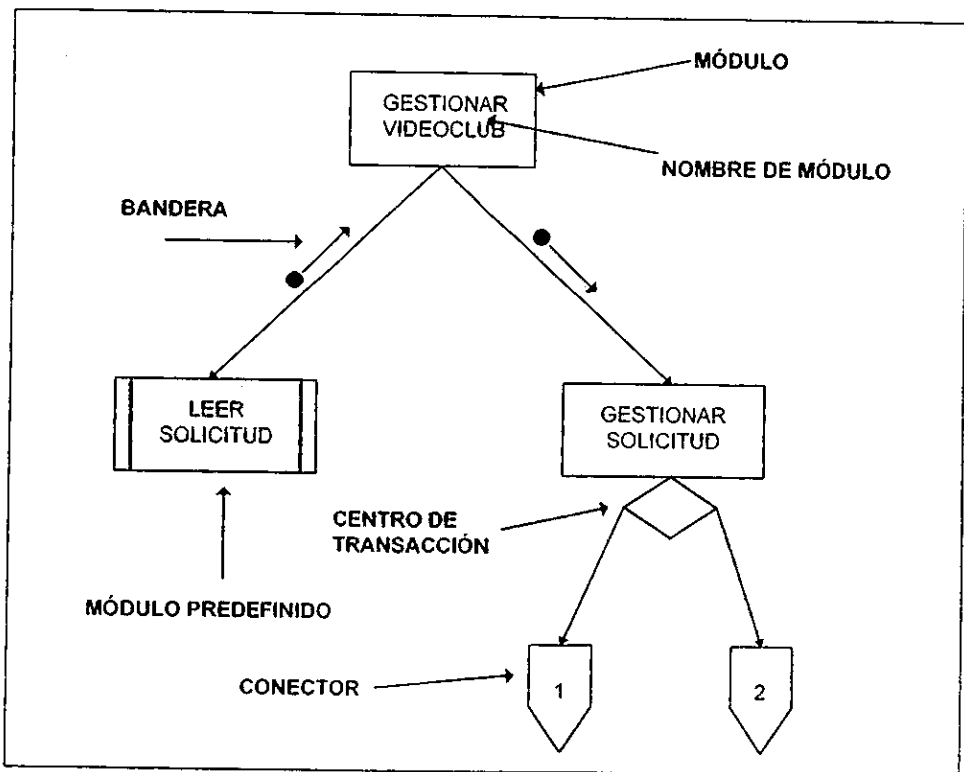


Fig. 1.1.2.C. Diagrama de estructura.

1.1.2.4. Matrices.

La técnica denominada genéricamente *matrices*, ayudan principalmente a realizar la verificación y validación entre el mundo de los datos y el de los procesos (matriz de función vs entidad); asimismo, sirven para que se realicen con éxito otras tantas (matriz entidad vs atributos) y para comprender mucho mejor el sistema de estudio (matriz de roles de usuarios vs funciones). Un punto de interés es que este tipo de técnicas pueden realizarse y aplicarse sobre cualquier clase de actividad que se realice a lo largo del ciclo de vida del sistema. Algunas de las matrices que son de gran utilidad en el desarrollo de sistemas son:

- **Matriz función/entidad.** La columna de la tabla corresponde a todas las entidades identificadas en el modelo lógico de datos. Las filas de la tabla corresponden a todos los eventos que se hayan identificado a partir de los DFD y del DJF (el nivel de los DFD lógicos más bajos muestran los eventos o funciones elementales). Una vez identificadas las entidades y las funciones, se relacionan mediante la matriz: se reconoce qué efecto (creación C, modificación M o borrado B) tiene cada evento sobre las entidades y se ha de marcar en la casilla correspondiente, como se muestra en el ejemplo de la fig. 1.1.2.D.

Entidad	CLIENTE	PEDIDO	PRODUCTO
Función			
1.1. Mantener cliente	C M B		
1.2. Crear producto		C	
1.3. Aceptar pedido		C	
1.4. Enviar factura		C	
1.5. Guardar pedido		C	
1.6. Mantener producto			C M B

Fig. 1.1.2.D. Ejemplo de matriz función vs entidad.

- **Matriz entidad/entidad.** Una vez identificadas las entidades, una forma de identificar las relaciones entre dichas entidades es usando la matriz entidad/entidad. Cuando existe una relación entre dos entidades, se marca con una X en la intersección correspondiente. A veces se puede ser más riguroso indicando la cardinalidad en la intersección correspondiente al nombre de la relación existente entre las dos entidades. Esta matriz es útil sólo cuando el sistema de análisis es demasiado grande para ser representado directamente en el DER con toda seguridad, y en casos donde se cuente con poca experiencia en el análisis de entidades y relaciones. La fig. 1.1.2.E. nos muestra un ejemplo de esta matriz.

Entidad	CLIENTE	PEDIDO	PRODUCTO
Entidad			
CLIENTE		solicita muchos	
PEDIDO	es solicitado por un		está formado por muchos
PRODUCTO		forma un	

Fig. 1.1.2.E. Ejemplo de matriz entidad vs entidad.

- **Matriz de funciones/ rol del usuario.** La matriz muestra qué papel desempeña el usuario y qué funciones puede realizar. Se ha de verificar esta matriz con los usuarios para asegurarse de que se han identificado todos los diálogos (intersecciones de la matriz) que se quieren y que ninguno es superfluo. La fig. 1.1.2.F nos muestra un ejemplo.

Usuario Función	Atención al Cliente	Compras	Encargado de Facturación
1.1. Mantener cliente	X		
1.2. Crear producto		X	
1.3. Aceptar pedido		X	
1.4. Enviar factura			X
1.5. Guardar pedido		X	
1.6. Mantener producto		X	

Fig. 1.1.2.F Ejemplo de matriz funciones vs rol del usuario.

1.1.2.5. Diagrama de transición de estados (DTE).

El diagrama de transición de estados (DTE) es una técnica de modelado que se base en el comportamiento dependiente del tiempo de un sistema. Normalmente se utiliza para representar el comportamiento de sistemas de tiempo real, donde el software debe responder a sucesos del mundo real en un tiempo muy limitado. Los componentes mostrados en la fig. 1.1.2.G., son:

- **Estado.** El estado representa un modo externo de comportamiento. El nombre del estado es el nombre del comportamiento exhibido por el sistema. Uno de los estados de un DTE será el inicial, y se representa con una flecha de transición apuntando hacia él y sin ningún estado origen; esta flecha puede tener o no condiciones o acciones. El estado inicial puede considerarse como el

comportamiento del sistema antes de que haya ocurrido ninguna transición. Puede haber uno varios estados finales.

- **Transiciones.** Representan el paso de un estado a otro. También puede ocurrir que una transición termine en el mismo estado del que parte, o que haya múltiples transiciones desde o hacia otro estado.
- **Condiciones.** Las condiciones provocan una transición en el sistema. Una vez que se cumple la condición, se toman las acciones indicada y se cambia de estado. Se pueden dar varias acciones independientes en una transición sencilla. Se asume que estas acciones se producen instantáneamente y al mismo tiempo, a no ser que se indique una secuencia explícitamente.

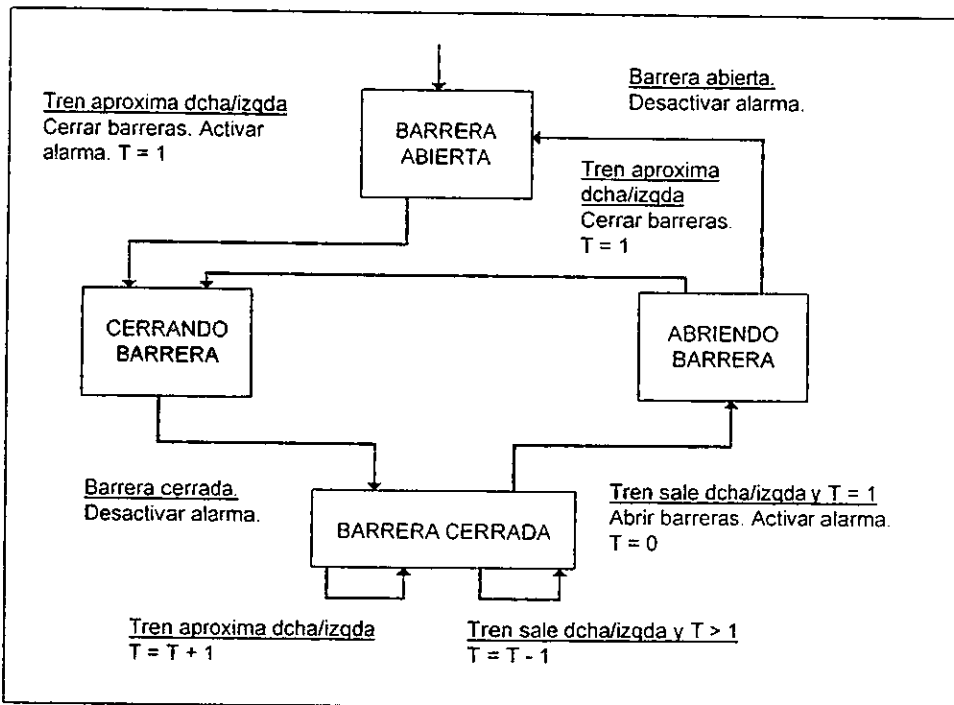


Fig. 1.1.2.G Ejemplo de un diagrama de transición de estados.

1.1.3. METODOLOGÍA DE YOURDON.

Debemos a las aportaciones de Yourdon gran parte de las técnicas de desarrollo de sistemas, principalmente el diagrama de flujo de datos; así como la afinación del diseño estructurado. Su metodología se cimenta fundamentalmente en el DFD, cuyo proceso de análisis es sencillo. Consiste en un refinamiento progresivo a través de la siguiente fases:

1. Construcción del diagrama de contexto. Es aquel DFD compuesto por un solo proceso que representa todo el sistema, las entidades externas que participan en él y los flujos de entrada y de salida.
2. Descomposición de cada proceso en los subsiguientes niveles de DFD, hasta conseguir todos los procesos primitivos (aquellos que ya no se pueden seguir descomponiendo).
3. Descripción de todos los elementos obtenidos en el diccionario de datos.

Diagramación.

El modelado gráfico utilizado por Yourdon agrega a los productos entregados en cada etapa del ciclo de vida de un sistema, diagramas de contexto y de datos a los cuales se les da mayor peso antes de pasar a los DER y DJF. El diagrama de datos y su representación gráfica ya fue cubierta en el punto de herramientas.

Básicamente, la diagramación de Yourdon está basada en procesos que son representados por círculos, y almacenadores de información representados por un rectángulo simple o doble dependiendo si es externa o interna. Comenzando con el Diagrama de Contexto o 0 donde sólo existe un proceso general y sus entradas y salidas, se va detallando éste hasta llegar a funciones o procesos elementales. De aquí, pasar a DER y DJF es relativamente simple.

En la Fig. 1.1.3.2.A se ve un diagrama de contexto característico.

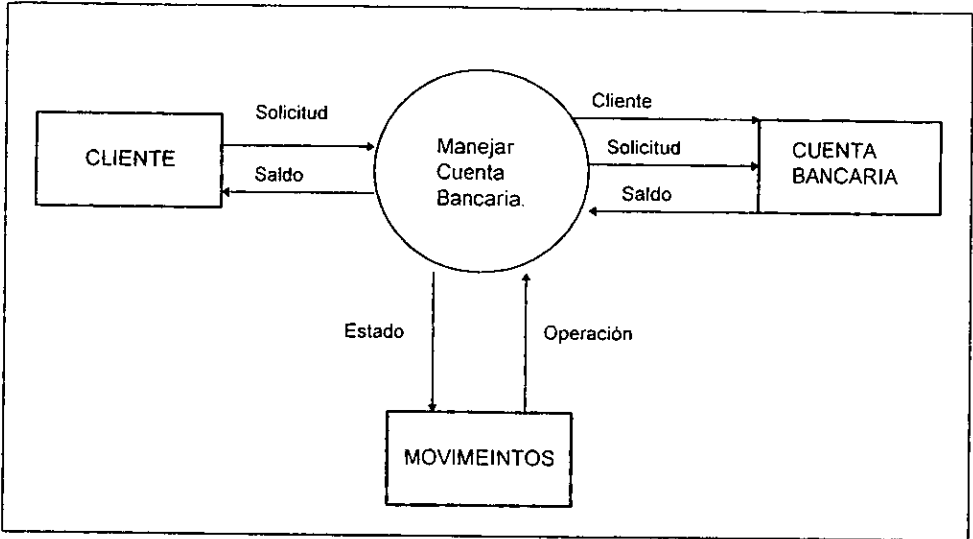


Fig. 1.1.3.2.A. Ejemplo de diagrama de contexto.

Hablar de una metodología única de Yourdon es prácticamente imposible, al ser una de las bases de casi toda metodología posterior. Para efectos de este trabajo, se generalizará en las siguientes fases:

- Definición del problema.
 - Inicio de la construcción del diagrama de contexto o nivel cero.
 - Inicio de la construcción del diagrama de transición de estados.
- Planificación.
 - Elaboración de un modelo de Empresa. Abarca la información abstracta de los datos, procesos, organización y estrategia de la empresa.
 - Elaboración de un plan de trabajo.
- Análisis.
 - Formalización de requisitos de información de un área de negocio.
 - Construcción de modelo conceptual que lo soporte.
 - Depuración detallada de diagramas DFD, DER y DTE.

- **Diseño.**
 - Transformación del modelo conceptual a la aplicación detallada de datos y procesos.
 - Definición de entornos y pruebas de la aplicación.
- **Construcción.**
 - Codificación de la nueva aplicación.
 - Realizar pruebas al sistema.
 - Implementarla en la organización.

Ventajas.

- Las necesidades de una dirección para tener la información son normalmente muy rápidas de identificar.
- La atención primaria se concentra de forma ordenada sobre los objetivos, problemas y factores críticos de éxito.
- La dirección consigue una mejor comprensión del negocio mediante las técnicas utilizadas.
- Excelente metodología para hacer estudio de procesos.

Desventajas.

- Asigna poca participación del usuario en el desarrollo del ciclo de vida del sistema.
- Centra su atención al DFD restándosela al DER quienes deben ser complementarios.

1.1.4. METODOLOGÍA SSADM.

Es una metodología desarrollada por el gobierno británico a principio de los años 80 para resolver o aminorar los problemas típicos de los departamentos informáticos, dando como resultado el SSADM (Structured Systems Analysis and Design Method). La estructura de la metodología SSADM comprende 6 etapas agrupadas en 5 módulos abarcando: el estudio de viabilidad, análisis y diseño. A continuación se presentan estos módulos y sus etapas.

1. Módulo FS (Feasibility Study). Estudio de viabilidad. Sus etapas son:

- Etapa 0. Viabilidad:
 - Preparar el estudio de viabilidad. Definir el problema. Seleccionar opciones. Componer el informe de viabilidad.

2. Módulo RA (Requirements Analysis). Análisis de requerimientos.

- Etapa 1. Investigación del sistema actual.
 - Establecer el marco de trabajo. Investigar y definir requerimientos. Investigar procesos del sistema actual. Investigar datos del sistema actual. Obtener una visión lógica del sistema actual. Componer los resultados de la investigación.
- Etapa 2. Estudio de opciones del sistema
 - Definir opciones del sistema. Seleccionar una de las opciones.

3. Módulo RS (Requirements Specification). Especificación de requerimientos.

- Etapa 3. Definición de requerimientos.
 - Definir procesos del sistema requerido. Definir el modelo de datos requerido. Obtener las funciones del sistema. Refinar el modelo de datos requerido. Desarrollar prototipos. Desarrollar especificaciones de procesos. Confirmar objetivos del sistema. Componer la especificación de requerimientos.

4. Módulo LS (Logical System Specification). Especificación del sistema lógico.

- Etapa 4. Opciones del sistema técnico.

- Definir opciones técnicas. Seleccionar opción técnica.
 - Etapa 5. Diseño lógico.
 - Definir diálogos de usuario. Definir procesos de actualización. Definir procesos de consulta. Componer el diseño lógico.
5. Módulo PD (Physical Design). Diseño físico.
- Etapa 6. Diseño físico.
 - Preparar el diseño físico. Crear el diseño físico de datos. Crear mapas de implementación de funciones. Optimizar el diseño físico de datos. Completar la especificación de funciones. Consolidar interfaces de proceso de datos. Componer el diseño físico.

Diagramación.

En la fig. 1.1.4.A podemos observar los estándares de diagramación utilizados por este método. Observamos que los subtipos, identificadores únicos y relaciones no transferibles no se muestran en el diagrama. En esta metodología, los subtipos son modelados con el uso de arcos exclusivos (esto se observa en la entidad ORDER).

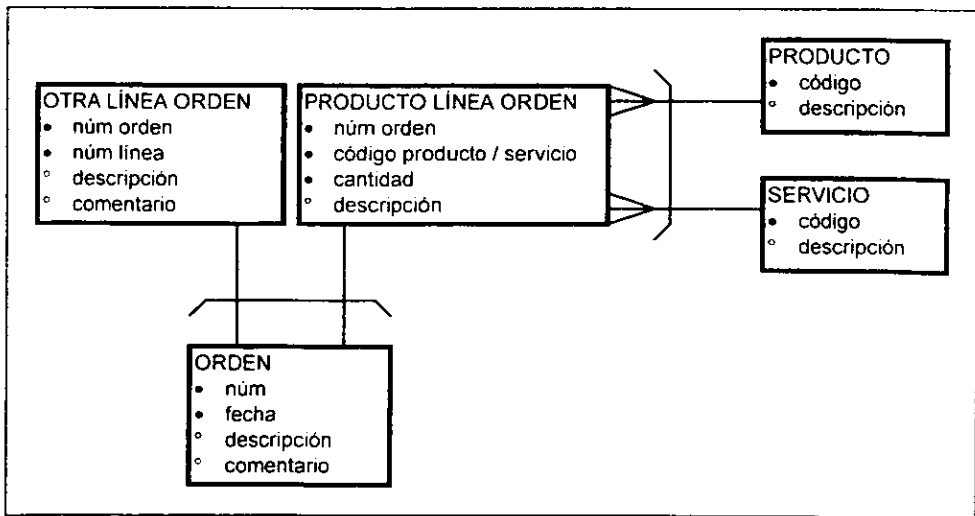


Fig. 1.1.3 4.A. Ejemplo de diagramación del método SSADM.

Ventajas.

- La alta participación del usuario en la definición de los medios asegura que el sistema diseñado coincida con el sistema requerido.
- Nos proporciona tres vistas independientes del sistema:
 1. La funcionalidad, a través de los diagramas de flujo de datos (DFD), donde vemos los procesos que se desarrollan en nuestro sistema.
 2. Los datos con un estudio exhaustivo del modelo (técnicas top-down y bottom-up).
 3. La lógica de nuestro sistema con una técnica que prácticamente es utilizada sólo por esta metodología: la historia de vida de la entidad.
- La separación de conceptos lógicos y físicos facilitan en cierto modo la implementación del sistema analizado en distintos entornos.
- SSADM produce gran cantidad de documentación a lo largo de todas sus etapas, lo que supone una importante ayuda para el mantenimiento y producción del sistema diseñado.

Desventajas.

- El carácter rígido y exigente de su origen británico. Cada paso de cada etapa está rigurosamente definido así como los subproductos que se van a ir obteniendo en el camino.
- Las técnicas de estudio de datos y procesos son similares a las de otras metodologías, pero aporta las suyas propias, lo que requiere una formación y entrenamiento adicionales si se pretende seguir el método al pie de la letra.

1.1.5. METODOLOGÍA MÉTRICA VERSIÓN 2.

Esta es una metodología de origen español que nació para marcar las directrices en materia de tecnología de la información y comunicaciones, creada en los años ochenta. En la fig. 1.1.5.A se muestran las fases que la metodología propone:

FASES	MÓDULOS
1. Planificación.	PSI. Plan de sistemas de información ARS. Análisis de requisitos del sistema.
2. Diseño	EFS. Especificación funcional del sistema. DTS. Diseño técnico del sistema.
3. Construcción	DCS. Desarrollo de componentes del sistema. DPU. Desarrollo de procedimiento de usuario.
4. Implementación	PIA. Pruebas, implantación y aceptación del sistema.

Fig. 1.1.5.A Fases de la metodología Métrica Versión 2.

Las técnicas incluidas son:

- Diagramas de flujo de datos.
- Modelado de datos.
 - Modelo conceptual: Diagrama de estructura de datos.
 - Modelo lógico: Normalización.
- Historia de la vida de las entidades.
- Diseño estructurado.
- Entrevistas.
- Análisis Costo-Beneficio.
- Pruebas.
 - Unitarias y de integración.

- Del sistema y de aceptación.
- Factores críticos de éxito.
- Técnicas matriciales.

Los productos que esta metodología entrega están esquematizados en la fig. 1.1.5.B.

FASES	MÓDULO	PRODUCTOS
Planificación.	PSI	Plan de acción. Nuevos sistemas a desarrollar y prioridades.
Análisis	ARS	Descripción del sistema actual. Datos y procesos. Catálogo de requisitos de usuarios. Modelo lógico de procesos de la alternativa seleccionada.
	EFS	Modelo de procesos del nuevo sistema. Modelo de datos del nuevo sistema. Modelo de eventos del nuevo sistema. Descripción de pantallas e informes.
Diseño	DTS	Diseño de la arquitectura física del sistema. Descripción de los componentes del sistema. Descripción de las interfaces entre componentes. Diseño físico de datos. Diseño detallado de pantallas. Especificación del entorno físico.
Construcción	DCS	Codificación de los componentes del sistema. Pruebas unitarias de componentes. Pruebas de integración entre componentes.
	DPU	Procedimientos de usuario (manual).
Implementación	PIA	Pruebas del sistema completamente realizadas. Pruebas de aceptación del sistema realizadas por los usuarios. Implantación final del sistema.

Fig. 1.1.5.B. Productos de la metodología Métrica Versión 2.

Ventajas.

- Puede utilizarse en el desarrollo de proyectos de distinta envergadura; al ser una metodología general y abierta utilizable en diversos entornos físicos y lógicos.

Desventajas.

- Al ser un producto relativamente nuevo en nuestro entorno, cabe la posibilidad de mal interpretarlo o no adecuarlo a las necesidades reales y actuales con las que nos enfrentamos.

1.1.6. METODOLOGÍA MERISE.

Esta metodología de hechura francesa, nace en los años setenta y considera a la empresa como un todo a ser estudiado, tratando su automatización como un acto estratégico que maneja un recurso básico como lo es la información, y por lo tanto debe acometerse de manera reflexiva y ordenada. Las etapas que considera son:

- Estudio previo. Definición global de las soluciones conceptuales, organizacionales y técnicas en relación con el dominio objeto de estudio.
 - Estudio de lo existente.
 - Concepción del nuevo sistema.
 - Conclusión del estudio previo.
- Estudio detallado. Definición explícita de las especificaciones funcionales de la aplicación del objeto de estudio.
 - Diseño funcional
 - Diseño detallado.
 - Conclusión del estudio detallado.
- Estudio técnico. Distribución de datos en archivos físicos, y el tratamiento en módulos de programas en función del sistema informático a usar.
 - Diseño técnico.
 - Modelo físico de datos.
 - Modelo operativo de tratamientos.
 - Estrategia de producción / entrega del software.
- Realización y puesta en marcha. Producción de los módulos de programación e implementación de los medios técnicos necesarios, así como formación del personal, y recepción definitiva por parte del usuario.
 - Producción del software.
 - Pruebas individuales.
 - Pruebas de integración.
 - Preparación de los recursos de puesta en marcha.

- Entrega y lanzamiento.
- Mantenimiento. Abarca el resto de la vida del sistema hasta que sea sustituido por otro nuevo.
 - Estudio del impacto.
 - Análisis de las modificaciones.
 - Realización de las modificaciones.
 - Puesta en marcha del sistema.

Ventajas:

- Aproximación racional, que abarca el ciclo de vida completo de un sistema de información asegurando su coherencia.
- Amplia utilización de diversas técnicas de modelado.
- Cronológica de los procesos.

Desventajas:

- Definición estricta del método de trabajo.
- Inclusión de nuevas técnicas de modelado no practicadas en nuestro medio.
- Necesario un proceso de actualización conceptual.

1.1.7. METODOLOGÍA DE CASE*METHOD.

Esta metodología conjuga las principales propiedades de la Ingeniería de Software propuestas por Yourdon principalmente, y que cubren totalmente el ciclo de vida de un sistema. La metodología está dividida en diferentes fases, cada una con un conjunto de tareas y productos específicos que llevados a cabo, garantizan un mejor entendimiento y desarrollo del sistema. La metodología une cada fase de tal manera que los resultados de una son la información base de la siguiente, por esto, es importante mantener un nivel de calidad y consistencia de información.

Un punto esencial de la metodología es que deben existir, en todas las etapas, entrevistas entre el usuario y el analista, permitiendo así una retroalimentación en ambos sentidos, evitando pérdida de información en este proceso. Esta metodología utiliza la técnica top-down y bottom-up. Al analizar un sistema el objetivo es obtener la mayor información posible del sistema desde las cosas más genéricas hasta las más detalladas (top-down) , y si se combina con el bottom-up, donde se realiza un análisis detallado, lograremos asegurar que no falte información. A continuación señalaremos las características y productos principales de cada una de las etapas de esta metodología.

- **Estrategia.** El objetivo de esta etapa es producir junto con el usuario, el modelo del negocio, un conjunto de recomendaciones y un plan de desarrollo de sistemas, los cuales cubran los requerimientos actuales y futuros del negocio, tomando en cuenta las restricciones organizacionales, financieras y técnicas. El análisis que se tiene que desarrollar en esta etapa debe ser completo pero a nivel general, puesto que la intención en este punto es modelar la situación actual del negocio y definir la dirección del mismo, indicando los factores críticos de éxito de la solución. Los principales productos entregables de esta etapa son:

- Términos de referencia. Se definen objetivos, prioridades, restricciones, factores críticos de éxito y términos del negocio, que se usarán como lineamientos de trabajo.
 - Diagrama de contexto. Diagrama general de la situación actual y de la situación esperada en donde se representan las áreas y procesos principales que se ven involucrados .
 - Diagrama Entidad-Relación. Técnica de modelado para definir la información que necesita la organización. Se definen las *entidades* (personas, lugares o cosas de importancia de la organización) , *atributos* (sus propiedades) y *relaciones* (cómo están relacionadas unas con otras).
 - Diagrama de descomposición funcional a primer nivel (Jerárquico-Funcional). Es un conjunto de funciones organizadas en forma jerárquica donde se concentran el qué y posteriormente el cómo, se realiza en el negocio.
 - Plan de desarrollo.
- **Análisis.** En esta etapa se toman y se verifican los resultados de la etapa de estrategia y los detalla y refina al nivel suficiente para especificar los requerimientos de la organización, planteando modelos que reflejen adecuadamente la situación de la organización y desarrollando alternativas válidas de solución. Esta etapa del ciclo de vida es muy importante, pues la información resultante será la base para el éxito o fracaso del sistema, por tanto es recomendable que el tiempo que se asigne a esta etapa sea largo, de tal forma que los productos que se obtengan puedan ser utilizados para el desarrollo del sistema. Los principales entregables de esta etapa son:
- Diagrama Entidad-Relación completo. Con la información obtenida de la estrategia y análisis, se consolida el diagrama Entidad-Relación, el cual debe modelar fielmente la estructura del negocio. El diagrama debe contener las entidades y sus relaciones a su nivel más detallado.
 - Diagrama Jerárquico-Funcional detallado. Con la misma información, se integra un diagrama consolidado de las funciones que realiza el negocio,

detallado hasta la función más elemental, de manera que se puedan determinar los módulos que compondrán al sistema.

- **Diagrama de Flujo de Datos.** Diagrama que nos permite identificar cómo fluye la información y las relaciones que existen con otros entes externos.
 - **Matrices de validación.** Matrices que cruzan elementos del sistema contra otros tantos (p.ej. funciones vs entidades o atributos vs funciones) para corroborar la integridad y correspondencia de la información.
 - **Volúmenes de datos.** Empiezan a considerarse en esta fase para ir dimensionando la base de datos que se creará.
 - **Plan de desarrollo.**
- **Diseño.** En esta etapa se toman y se verifican los resultados de la etapa de análisis y se encuentra el mejor camino para cubrir los requerimientos, dado el ambiente técnico y los niveles requeridos de automatización. El objetivo es producir un diseño apropiado a las necesidades del negocio, dentro de las restricciones técnicas y considerado el enorme impacto que tiene dicho diseño, en el rendimiento final del sistema. Los principales entregables de esta etapa son:
- **Arquitectura del sistema.** Se entrega el proyecto del arreglo del equipo así como de los elementos modulares, bajo la arquitectura que será utilizada para el desarrollo del sistema.
 - **Diseño de módulos.** Se mapean las funciones elementales del DJF y se detalla el funcionamiento de los procesos puntuales, pantallas y reportes que se construirán en la siguiente etapa.
 - **Diseño de la base de datos.** Se mapean las entidades con sus atributos y relaciones del DER, a tablas, columnas y constraints (*reglas de validación*) así como las vistas e índices que emergen de fases anteriores.
 - **Dimensionamiento detallado.** Del análisis se obtiene un tamaño aproximado de cada una de las entidades y por ende de las tablas, de esta información se puede entonces, crear la Base de Datos lo más cercano a la realidad.
 - **Procedimientos operativos o manuales.** Los procesos que necesariamente requieren de una logística manual para llevar a cabo el buen funcionamiento

del sistema, también deben ser documentados para asegurar el correcto funcionamiento de lo automatizable.

- **Construcción.** El objetivo de esta etapa es el codificar y probar los programas utilizando las herramientas adecuadas al problema, que serán escogidas dependiendo del ambiente técnico y del tipo de programa. Los principales entregables de esta etapa son:
 - Diseño de programas
 - Base de datos afinada
 - Programas trabajando y probados
 - Plan de transición
 - Resultados de la prueba del sistema
 - Hardware y software instalado
 - Indicadores de rendimiento para la aplicación.
- **Documentación.** El objetivo de esta etapa es el armar los manuales del usuario y la documentación técnica. Cabe señalar que durante todo el ciclo de vida del sistema, la documentación de cada fase ha quedado registrada en documentos, reportes y diagramas. Los manuales técnicos y del usuario debe ser una ayuda a técnicos como usuarios facilitándoles el conocimiento y el posterior control del producto por liberarse. Los principales entregables de esta etapa son:
 - Documentación del usuario
 - Documentación técnica
- **Transición.** El objetivo de esta etapa es el realizar todas las tareas necesarias para la implementación del sistema proporcionando un período inicial de soporte y apoyo. La transición no debe demandar una nueva organización al usuario y sí, la capacitación de éste para utilizar y controlar autónomamente el sistema. Los principales entregables de esta etapa son:
 - Material de entrenamiento
 - Instalación completa del sistema
 - Conversión de datos. De sistemas viejos o archivos requeridos.
 - Activación de archivos para análisis, clasificación y corrección de errores

- Documentación completa del sistema.
- **Producción.** El objetivo de esta etapa es asegurar que el sistema funcione adecuadamente, con la mínima intervención del personal de operación y soporte, y monitorear su rendimiento. También el incluir los cambios necesarios de manera que no interrumpan y que aseguren el entusiasmo y compromiso del usuario. Los principales entregables de esta etapa son:
 - Archivos de respaldo y recuperación
 - Registro de control de cambios
 - Reporte de fallas
 - Cambios/Adiciones al sistema
 - Estadísticas de rendimiento
 - Nuevos requerimientos
 - Resultados de auditoría al sistema.

Diagramación.

Las convenciones aportadas por Gane and Sarson que de hecho, son las adoptadas por la metodología son:

- **Entidades:** se representan en el diagrama como un rectángulo con esquinas redondeadas (softbox), su nombre debe ser especificado en singular y la definición de cada entidad debe ser única.
- **Relaciones:** una relación es binaria en el sentido de que siempre habrá una relación entre dos entidades o entre una entidad y ella misma. Cada relación tiene dos terminaciones cuyas características son: nombre, grado y opcionalidad. Una relación se encuentra representada por una línea que une dos entidades o recursivamente una entidad con sí misma. Las relaciones más comunes son aquellas que son muchos a uno y obligatorias. Para el grado de muchos, la línea de la relación tiene tres puntos llamada "pata de gallo" (rowsfoot), para el grado de uno, la línea se une en un punto. Cuando la relación terminal es obligatoria, se debe dibujar una línea continua hasta la mitad de esa relación y cuando sea opcional, se dibujará una línea

punteada. También existen las relaciones recursivas que denotan un nivel jerárquico.

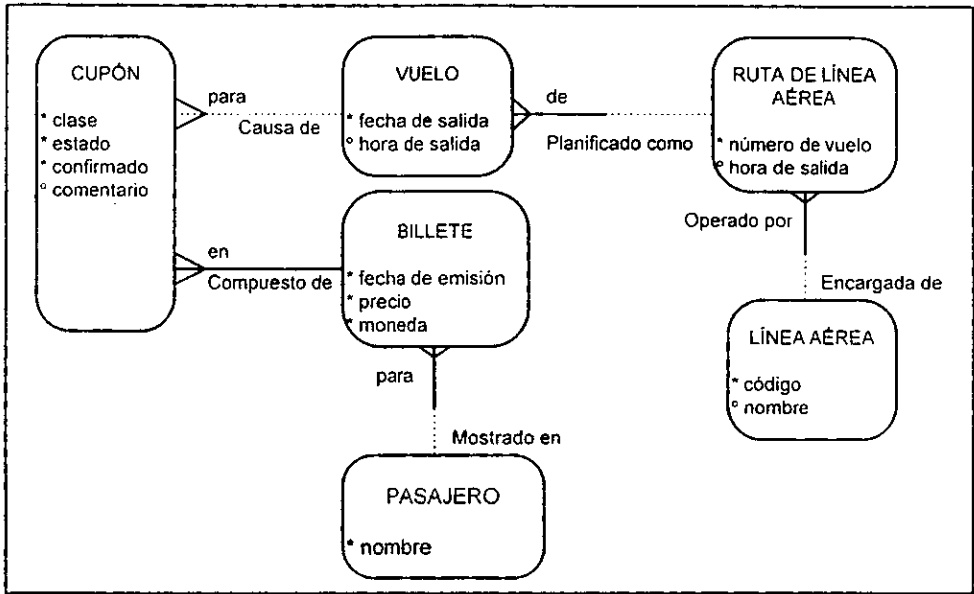


Fig. 1.1.7.A Diagrama entidad - relación bajo los estándares de Gane y Sarson .

1.2. BASES DE DATOS

1.2.1. CONCEPTOS

El desarrollo de la Base de Datos colectiva es sin duda una de las actividades más importantes en el campo de la informática en lo que va del presente decenio. Los datos se observarán con mayor interés como un recurso vital para la empresa que merece ser debidamente organizada con el fin de maximizar su valía. Aparte de las bases de datos propias de empresas y diversos organismos, día a día va acrecentándose la demanda por los servicios de información para el público en general, capacitados para el acopio, la organización y la venta de datos.

En la actualidad existe mucha redundancia en los datos utilizados por las organizaciones y es frecuente que el mismo ítem de datos se defina diferentemente por los departamentos dentro de la organización. En muchas ocasiones, sólo después de instalada la computadora se advierte que eran erróneas algunas de las ideas que sirvieron de base para su diseño o cuán imprecisos eran los métodos previstos. La eliminación de toda imprecisión en la definición y el uso de los datos constituye un objetivo primordial que debe tenerse a la vista en todas las etapas del diseño y la integración de la Base de Datos. En varios casos, los métodos directos de diseño no proveen las respuestas adecuadas porque no toman en cuenta las intrincadas concertaciones a que hay que llegar entre los diversos aspectos de diseño.

En el transcurso del desarrollo de la Base de Datos se observa que pronto tendremos nuevos dispositivos de almacenamiento, nuevas técnicas de software y nuevos tipos de bases de datos. Los detalles cambiarán, pero los principios fundamentales seguirán siendo los mismos. Por lo tanto, el lector debería concentrarse en esos principios.

Sistema de Base de Datos.

Un sistema de procesamiento de datos tiene básicamente cuatro componentes: los datos, el hardware (la máquina donde corre), el software (los programas) y los usuarios.

Un sistema de Base de Datos no es más que un sistema de mantenimiento de registros basado en computadoras, es decir, un sistema cuyo propósito general es registrar y mantener información (datos).

Una definición de Base de Datos es: *Una colección de datos, integrada y generalizada, con su descripción, que se gestiona de tal forma que pueda satisfacer las distintas necesidades de sus usuarios.*

Una definición más elaborada y completa de Base de Datos es: *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 los 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 bases de datos cuando éstas son totalmente independientes desde el punto de vista estructural.*

Una base de datos es una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una o más aplicaciones; los datos son independientes de los programas que lo usan; se emplean métodos bien determinados para incluir datos nuevos y modificar o extraer los datos almacenados.

El uso de Bases de Datos nos permite:

- Controlar la redundancia
- Mantener la consistencia
- Lograr la integración de los datos
- Compartir los datos entre las diferentes aplicaciones
- Cumplir con los estándares
- Tener facilidad en el desarrollo de aplicaciones
- Uniformar los controles de seguridad, privación e integridad
- Independencia entre los datos y los programas
- Reducir el mantenimiento a los programas

Un sistema de manejo de Base de Datos (DBMS Data Base Management System) consiste en un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a esos datos. El objetivo primordial de una DBMS es crear un ambiente en que sea posible guardar y recuperar información de la base de datos en forma conveniente y eficiente.

Tipos de Bases de Datos

Existen enfoques alternativos para visualizar y manejar datos a un nivel lógico independientemente de cualquier estructura física de soporte en que se basen.

Los modelos de base de datos que existen son:

- Modelo Jerárquico
- Modelo de Red
- Modelo Relacional

1.2.2. MODELO JERÁRQUICO O DE ÁRBOL

La estructura lógica en la cual se sustenta la base de datos jerárquica o de árbol.

Un árbol se compone de nodo un raíz y varios nodos sucesores, ordenados jerárquicamente. Cada nodo representa una entidad (tipo de registro) y las relaciones entre entidades son las conexiones entre los nodos.

El nodo colocado en la parte superior es llamado padre y los nodos inferiores son los hijos.

En el sistema jerárquico, las conexiones entre archivos no dependen de la información contenida en ellos; se define al inicio y es fijo.

La característica sobresaliente de este modelo es el manejo de la conexión uno a muchos, entre un padre y varios hijos, en otras palabras, cada hijo sólo tiene un padre.

Como ejemplo podemos tener una representación de cursos escolares donde observamos la configuración arborécea de la información (Fig. 2.2.A.).

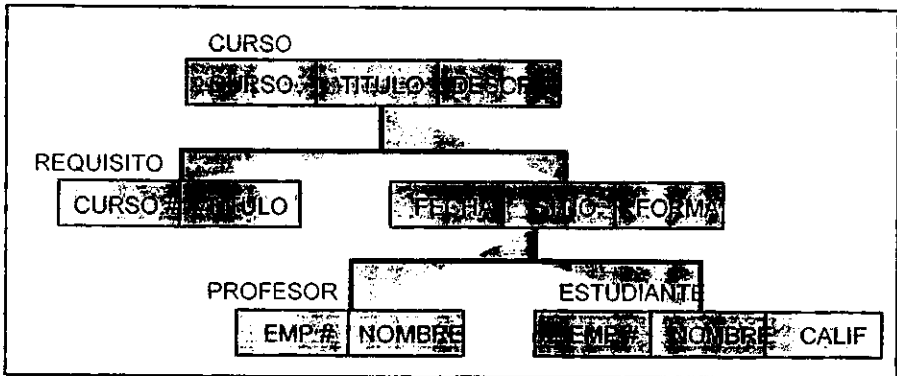


Fig. 1.2.2.A. Enfoque Jerárquico.

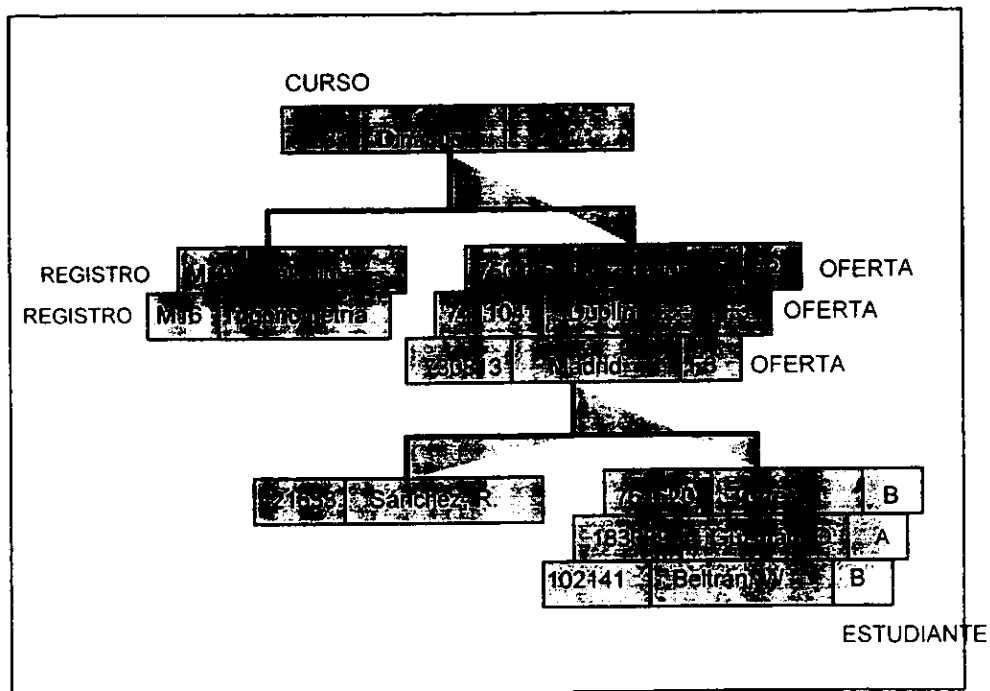


Fig. 1.2.2.B. Enfoque Jerárquico.

Desventajas en el enfoque jerárquico:

- No modela sencillamente las relaciones Muchos a Muchos.
- Anomalías de inserción.
- Anomalías de borrado.
- Anomalías de actualización
- Se pueden dar consultas inconscientes.

1.2.3. MODELO DE RED.

Los datos se representan como registros ligados formando un conjunto de datos interceptados.

La base de datos de red, a diferencia de las jerárquicas, permite cualquier conexión entre entidades, es decir, se pueden representar relaciones de muchos a muchos.

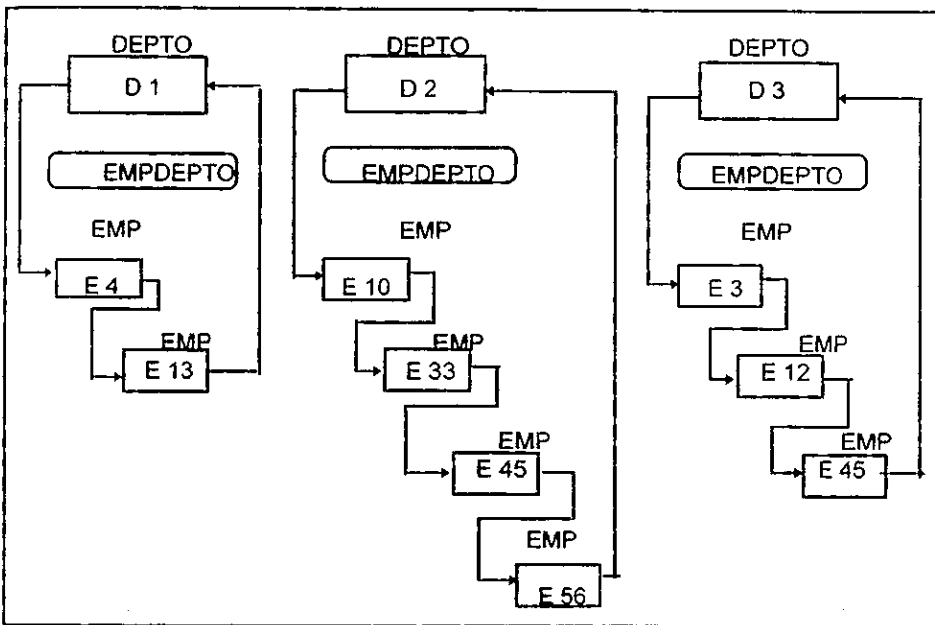


Fig. 1.2.3.A. Enfoque de red.

En una red, un hijo puede tener varios padres y varios hijos a la vez.

Como ejemplo podemos tener 2 entidades: Empleado y Departamento, cada Departamento puede tener varios empleados adscritos (Fig. 1.2.3.A).

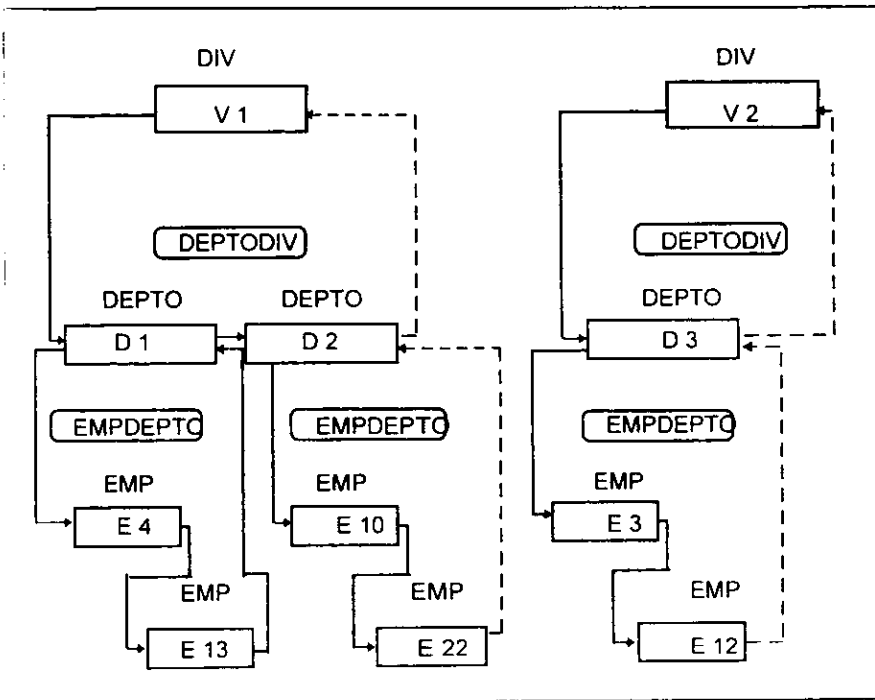


Fig. 1.2.3.B. Enfoque de Red.

A su vez, el departamento puede pertenecer a una división (Fig. 1.2.3.B).

Desventajas en el modelo de red:

- Resulta difícil hacer nuevas relaciones.
- Es complicado darle mantenimiento ya que cualquier cambio en la estructura requiere de una descarga de datos.
- Representa un desperdicio de recursos.
- Anomalías de inserción
- Anomalías de borrado.

1.2.4. MODELO RELACIONAL.

La estructura de una base de datos relacional está basada en la representación de entidades mediante tablas, las cuales constan de columnas (campos) y renglones (registros). Las relaciones entre tablas se llevan a cabo a través de un conjunto de columnas que se tengan en común. Logrando una conexión dinámica entre un número ilimitado de ellas a través del contenido de esas columnas.

La ventaja de los sistemas relacionales es el poder modificar la información sin la preocupación de especificar las combinaciones entre registros.

Como ejemplo podemos tener 2 entidades: cuentahabiente y cuenta, relacionadas entre sí con una sola relación mucho a muchos, es decir, un cuentahabiente puede tener varias cuentas, y una cuenta puede pertenecer a varios cuentahabientes (Fig. 1.2.4.A.).

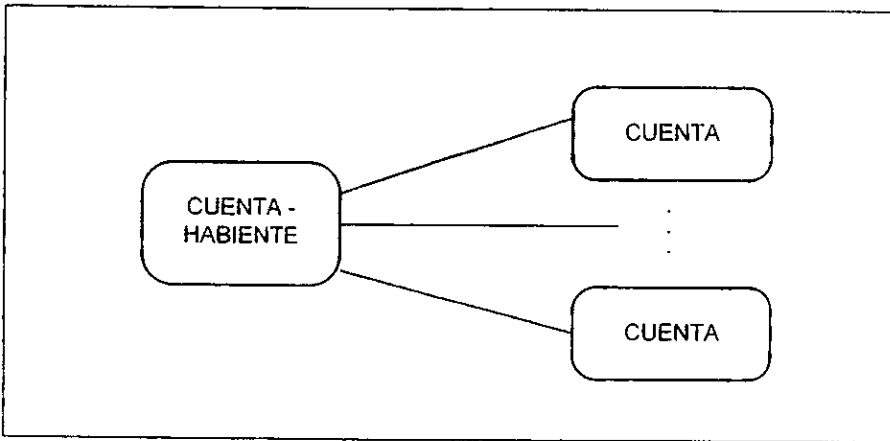


Fig. 1.2.4.A. Enfoque Relacional.

1.2.4.1. Modelo Entidad-Relación.

El modelo Entidad-Relación es una técnica para definir las necesidades de información de cualquier empresa. Esta técnica involucra conceptos que se identifican con varios objetos de importancia para la empresa, a los cuales se les denomina ENTIDADES, a las características de dichos objetos se les denomina ATRIBUTOS y a como se relacionan estos objetos entre sí se le denomina RELACIONES.

Todos estos conceptos se modelan a través de cierto tipo de esquemas gráficos, los cuales muestran a los usuarios una manera más sencilla y práctica de visualizar sus necesidades de información.

Una entidad es una persona, cosa o lugar, que cae dentro del alcance del sistema, acerca de la cual el sistema debe mantener, correlacionar y desplegar información. La entidad se representa por medio de una caja con las esquinas redondeadas y dentro de éstas se escribe el nombre de la entidad, el cual debe estar en singular. Cada entidad debe tener un nombre único dentro del sistema, lo que indica que no pueden existir dos entidades dentro del mismo sistema con el mismo nombre.

ORDEN
Número de Orden
PK
5
6
7
...

PRODUCTO
Código del Producto
PK
M23
S45
M47
...

Fig. 1.2.4.C. Entidad

La entidad dentro del modelo relacional se presenta por medio de una tabla, donde el nombre de la tabla debe contener una columna que identifique de forma única a cada renglón de ésta. Esta columna recibe el nombre de llave primaria (PK), la cual no puede contener valores nulos, ni duplicados, como se muestra en la fig. 2.4.C.

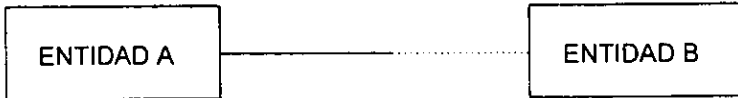
Una relación requiere de una o más entidades, la cual debe caer dentro del alcance del sistema, acerca de la cual el sistema debe mantener, correlacionar y desplegar información.

Una relación es un término matemático para representar una tabla bidimensional, como una colección de registros. Se define mediante filas y columnas, siendo cada entrada o elemento un valor de atributo. A lo anterior se le llama una relación y no una matriz porque falta homogeneidad en sus entradas.

Cada columna de una relación contiene los valores de un atributo con nombre, y cada fila se llama *tupla*. La palabra *tupla* se toma de la definición de grupos tales como quintuplas, séxtuplas, etc. Así pues, un grupo de n elementos es una n -tupla.

Las relaciones se presentan en varias modalidades:

- Relaciones Uno a Uno con las combinaciones de obligatorio a opcional, opcional a opcional y obligatorio a obligatorio.



- Relaciones Uno a Muchos con las combinaciones de obligatorio a opcional, opcional a opcional y obligatorio a obligatorio.



- Relaciones Muchos a Muchos con las combinaciones de obligatorio a opcional, opcional a opcional y obligatorio a obligatorio.



Las RELACIONES dentro del modelo relacional se representan de las siguientes formas:

- Relación uno a uno entre dos entidades se modela dibujando la llave primaria (PK) de la tabla con más renglones y como llaves foráneas (FK) en la otra tabla, como lo muestra la fig. 1.2.4.D.

PRODUCTO
CÓDIGO DEL PRODUCTO
PK
7
8
9
10
...

PRECIO	
CLAVE DE PRECIO	CÓDIGO DE PRODUCTO
PK	FK
1	FX45
2	GT11
3	GT17
4	GT76
...	...

Fig. 1.2.4.D. Relación Uno a Uno.

- Una llave foránea es una o más columnas que son llave primaria en otra tabla, una llave foránea permite nulos y valores duplicados.
- Relación Uno a Muchos o Muchos a Uno entre dos entidades se modela dibujado la llave primaria (PK) de la tabla que tiene la correspondencia de una como llave foránea (FK) en la otra tabla, como se muestra en la fig. 1.2.4.E.

CLIENTE
NÚMERO DE CLIENTE
PK
11
12
13
14
...

ORDEN	NÚMERO DE CLIENTE
NÚMERO DE ORDEN	NÚMERO DE CLIENTE
PK	FK
300	11
400	12
500	13
600	14
...	...

Fig. 1.2.4.E. Relación Uno a Muchos o Muchos a Uno.

- Relación Muchos a Muchos entre dos entidades se modela dibujando una tercer tabla, la cual se compone de una llave primaria (PK) compuesta de dos columnas, las cuales son llaves primarias y foráneas a la vez, como lo muestra la fig. 1.2.4.F.

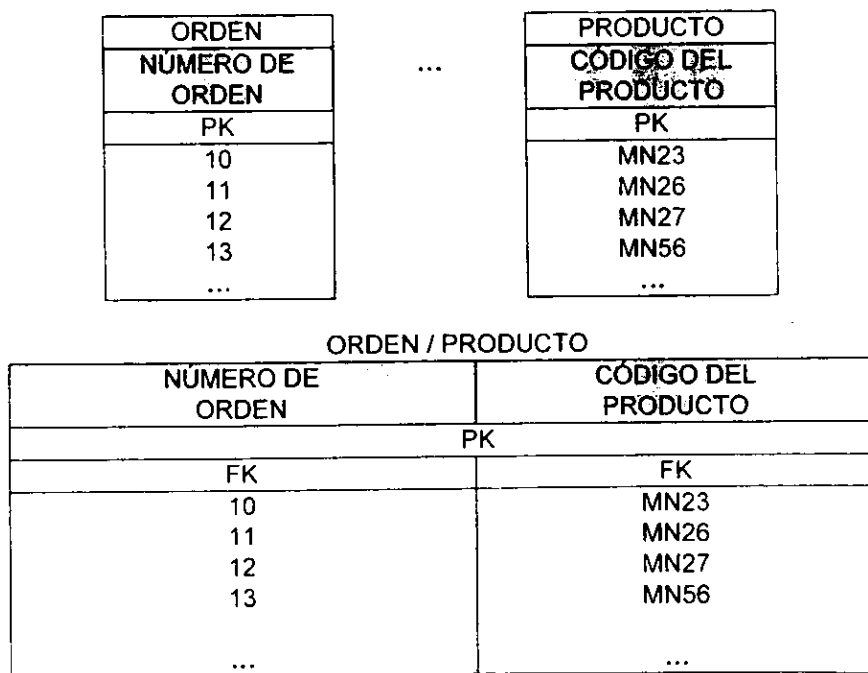


Fig. 1.2.4.F. Relación Muchos a Muchos.

Atributos

Un atributo es una características o cualidad de una entidad o relación, que cae dentro del alcance del sistema, acerca del cual el sistema debe mantener, correlacionar y desplegar información.

Para representar uno o varios atributos, se escribe el nombre del atributo dentro de la entidad.

Los atributos dentro del modelo relacional se presentan por medio de columnas dentro de una tabla.

1.2.4.2. 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
- La estructura debe ser tal que haya lugar para todos los datos requeridos.
- La redundancia que puede existir deberá ser usada por los elementos que son identificadores o llaves. Por lo que se debe tener cuidado de elegir aquellos que no estén sujetos a actualizaciones.
- Los efectos indeseables son las anomalías que pueden presentarse en las operaciones de actualización, inserción y eliminación.

Anomalía de inserción: No se debe 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 característica de adaptabilidad a los cambios maximizan la independencia de uso 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

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 de la semántica del contenido de la base de datos según la interpretación del diseñador de la base de datos.

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 de R es funcionalmente dependiente del atributo A de R si en cualquier instante, cada valor A no tiene más de un valor B asociado con él 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 sólo valor de cada uno de los atributos y no múltiples valores de éstos, si posee múltiples valores se dice que existen grupos repetitivos.

Se dice que una relación está en la **Primera Forma Normal** cuando 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.

La primera forma normal se alcanza cuando se quitan todos los grupos de repetición, de forma que un registro tenga longitud fija. Un grupo de repetición, es decir, la aparición repetida de un dato o grupo de datos dentro de un registro, es en realidad otra relación. Por lo tanto, se quita del registro y se le considera como una parte del mismo o como una relación adicional.

La redundancia puede reducirse separándola en otro grupo aparte de la entidad, debiendo contener el identificador de la entidad original para mantenerse relacionadas.

En la fig. 1.2.4.G. Se muestra la entidad Cliente con sus datos, donde observamos que existe un campo de repetición, delegación o municipio, que puede ser separado en otro grupo aparte (evitando tener tablas de longitud variable en campos repetitivos), quedando en la fig. 1.2.3.H. en primera forma normal.

No. de Factura	Cve Cliente	Cve Producto	Cantidad	Precio
A13	7	M13	8	200
A13	7	S16	2	200
A14	3	M13	4	200
A14	3	M34	6	400
A14	3	S16	10	200

Fig. 1.2.4.G. Entidad de Facturas

No. de Factura	Cve Cliente	Nombre del Cliente	Costo
A13	7	Miguel Ruiz	2,000
A14	3	Susana Robles	5,200

Fig. 1.2.4.H. Primera forma normal de Clientes.

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, y que además cumpla con la 1FN.

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 estas nuevas entidades el atributo del cual depende para mantenerla relacionada con la original. En las fig. 1.2.4.I y J se muestra lo anterior.

Cve	Nombre	Dirección	Del. O Mpio.
1	Juan Sánchez	Calle 23 Núm 45	Álvaro Obregón
2	María Rojas	Insurgentes Sur 345	Álvaro Obregón
3	Susana Robles	Av. Centenario 16	Coyoacán
4	Martín Reyes	Av Montevideo 160	Gustavo A. Madero
5	Juan Sánchez	Tres Cruces Núm 3	Coyoacán
6	José López	Volcán Tajín 14	Gustavo A. Madero
7	Miguel Ruiz	Delta Núm 378	Coyoacán

Fig. 1.2.4.I. Entidad Clientes

Cve.	Nombre Delegación o Mpio.
1	Alvaro Obregón
2	Coyoacán
3	Gustavo A.Madero

Fig. 1.2.4.J. Segunda forma normal

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. En la fig. 1.2.1.K se muestra esta conversión.

La tercera forma normal se alcanza cuando se quitan las dependencias transitivas de un diseño de registro:

- A, B y C son tres datos en un registro
- Si C es funcionalmente dependiente de B y
- B es funcionalmente de A entonces
- C es funcionalmente dependiente de A.
- Por lo tanto existe una dependencia transitiva.

Factura

Número de Factura	Cve Cliente	Precio
A13	7	2000
A13	7	5200

Cliente

Cve	Nombre	Precio
M13	Sala de Postproducción	200
S16	Copiado	200
M34	Uso de Rank	400

Cliente

Cve	Nombre	Dirección	Del. O Mpio.
1	Juan Sánchez	Calle 23 Núm 45	Álvaro Obregón
2	María Rojas	Insurgentes Sur 345	Álvaro Obregón
3	Susana Robles	Av. Centenario 16	Coyoacán
4	Martín Reyes	Av Montevideo 160	Gustavo A. Madero
5	Juan Sánchez	Tres Cruces Núm 3	Coyoacán
6	José López	Volcán Tajín 14	Gustavo A. Madero
7	Miguel Ruiz	Delta Núm 378	Coyoacán

Factura - Producto

No. Factura	Cve Producto	Cantidad
A13	M13	8
A13	S16	2
A14	M13	4
A14	M34	6
A14	S16	10

Fig. 1.2.4.L. Tercera forma normal

Finalmente se resume en este cuadro los pasos para normalizar nuestra base de datos en la fig. 1.2.4.M.

FORMA	PASOS
1a forma	Cambiar todas las estructuras que no sean bidimensionales (es decir, grupos de repetición) en estructuras de registro bidimensionales.
2a forma	Eliminar los datos que no dependan totalmente de las llaves de registro.
3a forma	Eliminar los datos que dependan transitivamente de las llaves primarias.

Fig. 1.2.4.M. Pasos en la normalización de la base de datos.

Lenguaje SQL

SQL es una abreviatura de "Structured query language" (lenguaje de consulta estructurado). Pero SQL es algo más que un lenguaje de consulta, a pesar de la palabra "consulta" que figura en su nombre. Primero, no sólo ofrece funciones de recuperación sino también una diversidad de operaciones de actualización,; por lo tanto es un lenguaje de DML (Data Management Language) y de DDL (Data Definition Language).

El acceso a la base de datos vía SQL, este es un lenguaje no procedural que contiene extensiones significativas que incluyen operaciones con fechas, funciones numéricas y con caracteres y subconsultas. Las instrucciones de SQL pueden ligarse con lenguajes de tercera generación tales como C, Cobol, Fortran y Pascal.

En una forma muy simplificada, SQL es un lenguaje para tener acceso a datos. Como a cualquier lenguaje, se le usa para entablar comunicación, por ejemplo con un administrador de base de datos, como DB2, SQLBase, ORACLE, INFORMIX, INGRES, o con cualesquiera sistemas relacionales de administración de bases de datos (RDBMS, por las palabras en inglés) que soporten el lenguaje SQL. Estos sistemas de

base de datos entienden el SQL, y cuando se aprende a conversar en SQL, puede hacerse un proceso de datos bastante elaborado.

A pesar de que SQL es un lenguaje de programación, es mucho más sencillo que los lenguajes como el COBOL, BASIC, FORTRAN o APL. Su aprendizaje es fácil. El SQL usa una forma libre para sus instrucciones, con una estructura como la del inglés; su sintaxis no es matemática. Se escogen (SELECT) los datos (DATA), de (FROM) una tabla, donde (WHERE) se satisfacen algunas condiciones. O se escriben (INSERT) los valores (VALUES) en (INTO) una tabla, o se borran (DELETE) los datos de (FROM) alguna tabla, donde (WHERE) se satisfagan las condiciones. Es muy lógico y fácil de comprender.

El SQL decide cómo manejar los datos: sacarlos de, o insertarlos en la base de datos. Solamente hay que decirle qué se desea, y el SQL hace lo demás. Esto se llama el *no ser navegante*, y promete ganancias en productividad para la industria del procesamiento de datos. A veces es posible la sustitución completa de algunos programas escritos en lenguajes convencionales mediante una sola consulta de SQL.

La mayoría de los administradores de bases de datos relacionales (DBMS) soportan el procesamiento de SQL tanto interactivo como estático. Es decir, las instrucciones del SQL pueden ejecutarse en una forma interactiva - cuando hay una comunicación directa con el DBM- o pueden insertarse en lenguajes de programación tradicionales como el COBOL. Esto es necesario, pues en sus comienzos se pensó en usar al SQL junto con otros lenguajes. El SQL no posee instrucciones para un diálogo en la pantalla, o para algo más que un muy básico esquema de dar formato a un reporte; por ello, la característica dual que se menciona es muy importante en cualquier clase de aplicación formal de procesamiento de datos. Los programas en línea, o por lotes, pueden tener acceso a los datos del DB2 mediante la inclusión de instrucciones de SQL. Éstas son muy similares a sus contrapartes interactivas. Sin embargo, en este escrito no se discutirán las instrucciones de SQL que se insertan en otros programas.

Por último, el SQL hace el proceso de datos al nivel de "conjunto", con lo que los ajustes transforman conjuntos completos de valores (renglones), y la consulta produce todo un conjunto de valores (una tabla de resultados). Esto es conceptualmente opuesto a la manera tradicional de proceso de archivos de datos: valor1...valor2...valor... . Aunque la manera de "conjunto" incrementa la potencia de, y simplifica los sistemas relacionales, las personas acostumbradas a la manera tradicional deben incorporar algunas consideraciones en su manera de trabajar.

A continuación se describen algunas de las capacidades del SQL.

Las capacidades de proceso del SQL.

El SQL se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de control de datos. Con estos tres lenguajes se cubre todo el espectro de la actividad del proceso relacional de datos. De hecho, en la mayoría de los productos basados en el SQL, todo el acceso a los datos se hace mediante el SQL, y solamente con él.

La *definición de datos* permite la creación, el borrado y la modificación de las estructuras de datos que necesite el sistema. Estas estructuras incluyen tablas, bases de datos, índices - las partes "estructurales" del sistema de la base de datos.

La *manipulación de datos* se divide en tres categorías: la recuperación, la manipulación y la actualización. La recuperación de datos implica la consulta con la base de datos: en la pantalla se presenta una selección de los datos almacenados en tablas. La manipulación de datos se refiere a las capacidades del SQL para hacer cálculos estadísticos, como la determinación de promedios y la suma y multiplicación de columnas. La actualización de datos se refiere al "mantenimiento" de la base de datos.

El *control de los datos* permite la definición de mecanismos de seguridad para evitar que se tenga acceso sin autorización a los datos de un sistema. Consta de las instrucciones *autorización* y *revocación* que afectan a otros usuarios de los objetos del sistema. Estas instrucciones de acceso permiten crear controles de la complejidad necesaria para la seguridad del sistema.

Una característica importante de las instrucciones del SQL es que pueden ejecutarse dinámicamente, cada vez que funcione el sistema. Por ejemplo, el borrado de una base de datos y la definición de una estructura completamente nueva es directa: Las bases de datos relacionales son fáciles de definir, y luego son fáciles de modificar.

1.3 REDES

1.3.1 DEFINICIÓN

Dependiendo del nivel de red (ver fig. 1.3.1.A) podemos tener diferentes definiciones.

Red de aplicaciones: es un arreglo de usuarios y/o sistemas, físicos y/o lógicos, que intercambian información a través de computadoras que forman una red. Internamente la red de computadoras esta conectada a una red de datos, y los usuarios desconocen la manera en que se transmiten los datos.

Red de computadoras : es un arreglo de computadoras que intercambian datos a través de los puntos de acceso de una red de datos. A este nivel, para los usuarios la red de datos es un servicio, y desconocen la manera como el servicio es realizado.

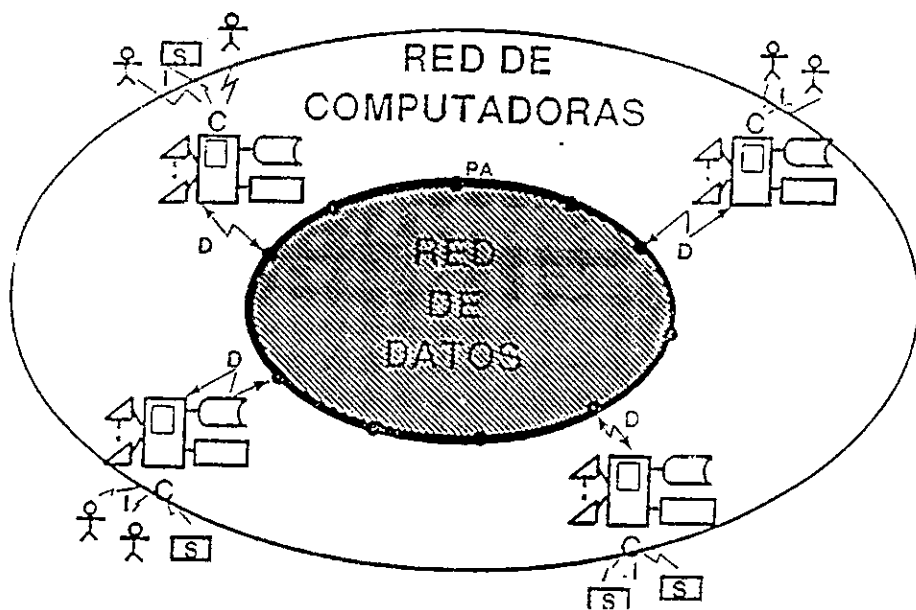


FIG. 1.3.1.A RED DE APLICACIONES Y COMPUTADORAS

Red de datos : Arreglo de medios , equipos y técnicas (protocolos) que permiten el intercambio de datos a través de sus puntos de acceso (ver fig. 1.3.1.B). Una red nos da varios beneficios entre los que tenemos :

- Conexión a una diversidad de equipos
- Compartir recursos de alto costo entre miembros de un grupo de trabajo
- Flexibilidad de crecimiento
- Control de procesos distribuidos. Establecer sistemas cliente servidor.
- Compartir información.
- Proveer mayor integridad de la información.

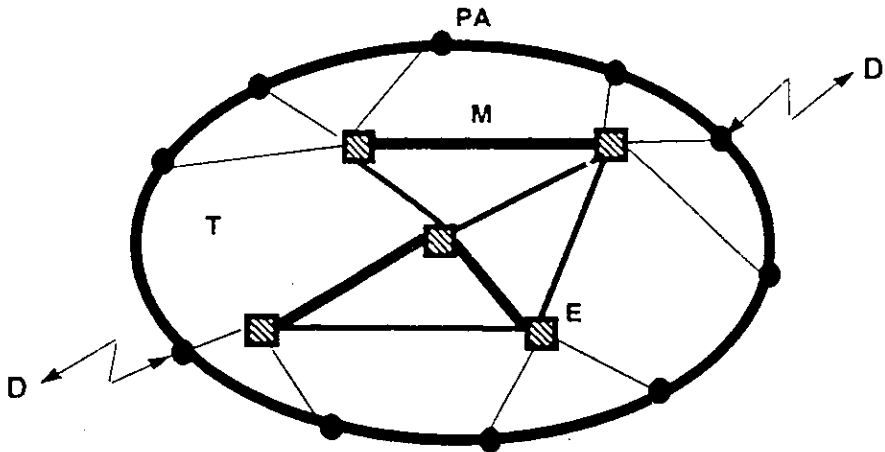


FIG1.3.1.B RED DE DATOS

1.3.2 TIPOS DE REDES.

Existen diferentes clasificaciones para las redes, estas dependen de su naturaleza.

Arquitectura

La arquitectura de una red es el conjunto de medios, equipos, topologías y técnicas definidas por los fabricantes u organismos.

Las redes de arquitectura propietaria son aquellas que sus elementos son definidos por los fabricantes. Ejemplo son SNA (*System Network Architecture - Arquitectura de Sistemas de Red*) de IBM, DECNet de Digital.

Las redes de arquitectura normalizada son aquellas definidas por organismos. Ejemplo TCP/IP (*Transport Control Protocol / Internet Protocol - Protocolo de Control de Transporte / Protocolo de Red de redes*).

Infraestructura de comunicación

Dependiendo de la infraestructura de comunicación que se use en la red se clasifican como:

Red Pública de datos : son compañías a nivel nacional y/o internacional que ofrecen su servicio de transporte de datos a sus abonados. Existen diferentes compañías que ofrecen estos servicios con calidades en capacidad de transmisión y seguridad. Los equipos de comunicación son propiedad de las compañías.

Las ventajas de este tipo de red son :

- Su cobertura (regional, nacional e internacional).

- Economía, no se requiere inversión en infraestructura de comunicación.

Las desventajas son :

- Costo alto si se tiene grandes y constantes volúmenes de información.
- Seguridad no 100 % confiable.
- Tiempos de respuestas dependientes de un tráfico externo.

Ejemplos de compañías: Telmex, Datapac, Internet.

Red privada de datos : son infraestructuras de comunicaciones creadas para las necesidades específicas de una institución. Su cobertura puede ser local a un edificio, regional, nacional e internacional.

Las ventajas de este tipo de red son :

- La adecuación de la red a las necesidades del usuario.
- Mayor seguridad.
- Tiempos de respuestas dependientes de su propio tráfico.

Sus desventajas son:

- Costo alto en infraestructura.
- Contratación de personal calificado para el mantenimiento y administración de las redes.

Ejemplos de estas redes son: BANAMEX, PEMEX.

Dimensión

Dependiendo del área que cubra una red se pueden clasificar en:

Red Local: El IEEE (*Institute of Electrical and Electronics Engineers - Instituto de Ingenieros eléctricos y electrónicos*) define a una red local como un sistema de comunicaciones de datos que permite a un número de dispositivos conectarse directamente entre ellos, en una área geográfica de tamaño moderada sobre un canal de comunicación a velocidades de datos moderadas. Por lo tanto, una red que cubre hasta un edificio puede ser considerada como una red local (ver fig. 1.3.2.A).

Las redes de área local pueden ser usadas para compartir datos, aplicaciones y dispositivos, en correo electrónico, en monitoreo de procesos en una fabrica, así como en sistemas de seguridad y alarmas. Sin embargo, la característica mas interesante en la que es usada una red local es su habilidad para soportar aplicaciones cooperativas cliente servidor en la que una aplicación corre parcialmente en un estación y parcialmente en servidor.

Existen diferentes topologías de redes locales (bus, anillo, estrella). Estas serán explicadas más adelante.

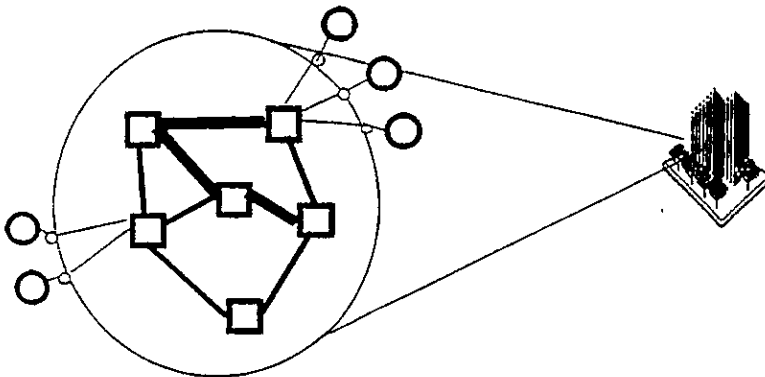


FIG 1.3.2.A RED LOCAL

Red de área extendida : la cual puede ser urbana, nacional o internacional (ver fig. 1.3.2.B). Este tipo de red se interconecta a través de enrutadores con conmutación por paquetes, enlaces digitales, satelitales, de radio, etc.

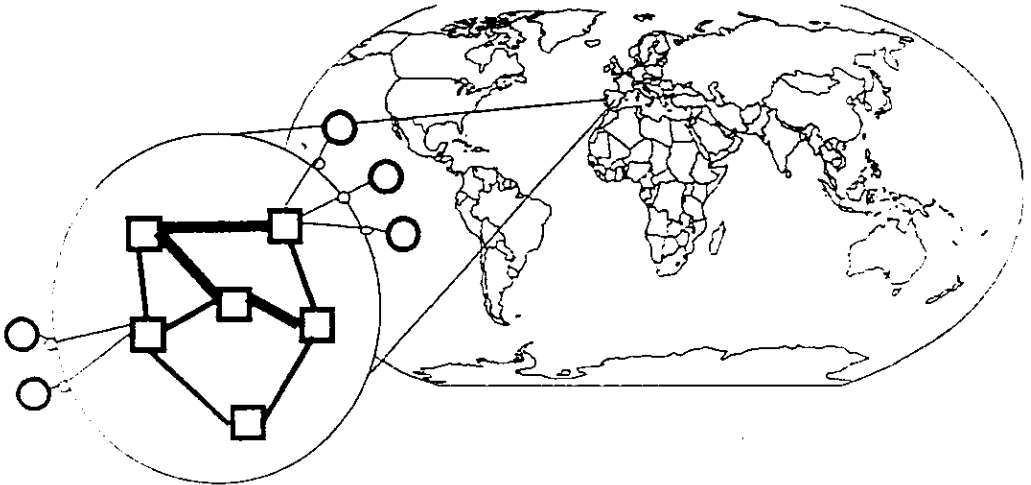


FIG1 3.2.B RED DE ÁREA AMPLIA

1.3.3 MODELOS DE RED

Modelo simple de referencia de red local.

Este modelo fue desarrollado por IEEE como parte del proyecto 802. Este modelo se enfoca principalmente al medio físico y su acceso a él.

		Control lógico de Enlace				IEEE
Medio de acceso		CSMA/CD 802.3	Token Ring 802.5	Token Bus 802.4	FFDI	
Medio		Cable Coaxial	Par Trenzado	Par Trenzado Blindado	Fibra Optica	

FIG. 1.3.3.A MODELO DE RED

En el control lógico de enlace bajo este modelo simple se incluyen los protocolos de comunicación (TCP/IP, SPX/IPX, Appletalk, etc.) y los sistemas operativos de red (Lan Manager, Novell Netware, Banyan Vines, Appletalk, etc.).

Modelo OSI

El modelo OSI (*Open Systems Interconnection - Interconexión de Sistemas Abiertos*) tiene las siguientes características :

- Publicado en 1980 por ISO (*International Standards Organization - Organización Internacional de Estándares*).
- Esquema de comunicación modular, constituido por 7 capas
- Especifica como deben fluir los datos entre sistemas
- Permite la intercomunicación con dispositivos de varios fabricantes.
- Es independiente a la topología y los medios físicos de comunicación.

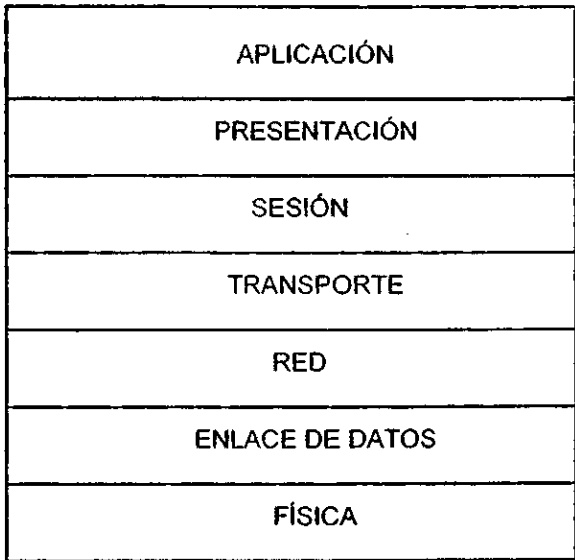


FIG. 1.3.3.B MODELO OSI

- Nivel físico : Envía y recibe bits sobre un medio de transmisión (cable coaxial, fibra óptica) a través de señales eléctricas.
- Nivel de enlace : Organiza y estructura bits en forma de tramas, para ser enviados o recibidos a través del medio físico.
- Nivel de red : Especifica medios de comunicación necesarios para establecer enlaces entre dos estaciones.
- Nivel de transporte : Establece y mantiene el transporte de comunicaciones simultáneas a través de la red. Este nivel asegura la confiabilidad de la información a través de la conexión.
- Nivel de sesión : Establece y mantiene la conexión de sesiones simultáneas, sincroniza y maneja el diálogo entre aplicaciones que se comunican, restablece el diálogo en caso de falla de transporte o caída de la red y libera la conexión de sesiones.
- Nivel de presentación : Realiza un intercambio de información usando formatos de datos que pueden ser enviados o recibidos por cada servicio de aplicación, transfiere diferentes formatos de datos sobre la conexión de una misma sesión (información como voz digital encriptada y comprimida e información texto en formato no encriptado y no comprimido).
- Nivel de Aplicación : Provee servicios de aplicación en forma individual entre los que se encuentran transferencia de archivos, envío y recepción de correos electrónicos.

Modelo DoD

El Departamento de Defensa (DoD) de los Estados Unidos al necesitar estandarizar el acceso a archivos a través de múltiples sistemas operativos y plataformas de hardware, fundó la creación de un conjunto de protocolos que han tenido gran influencia en los Estados Unidos y en el mundo. Los protocolos DoD son derivados de

un modelo de servicios como se muestra en la figura 1.4.3.C El nivel mas bajo de los servicios es prácticamente ignorado, abandonado a ser definido por otros organismos de estándares como el IEEE. La mayoría de los servicios basados en los protocolos DoD residen sobre el nivel físico. Estos protocolos basados en el modelo DoD son conocidos como TCP/IP.

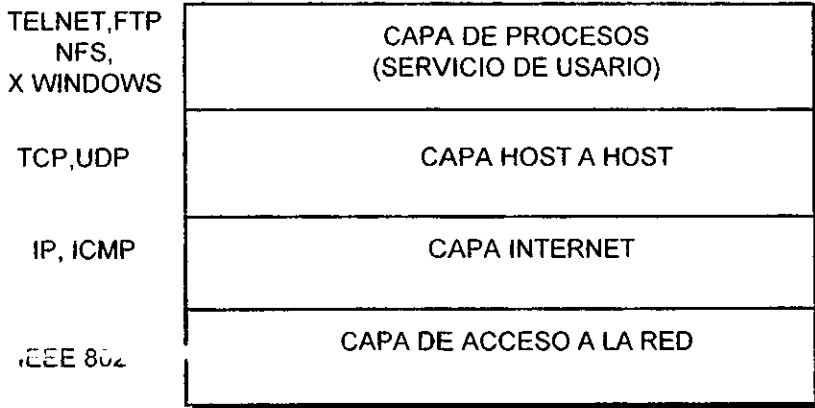


FIG.1.3.3.C MODELO DOD

En la figura 1.3.3.D se muestra una relación entre los tres modelos mencionados para que se pueda ver su correspondencia.

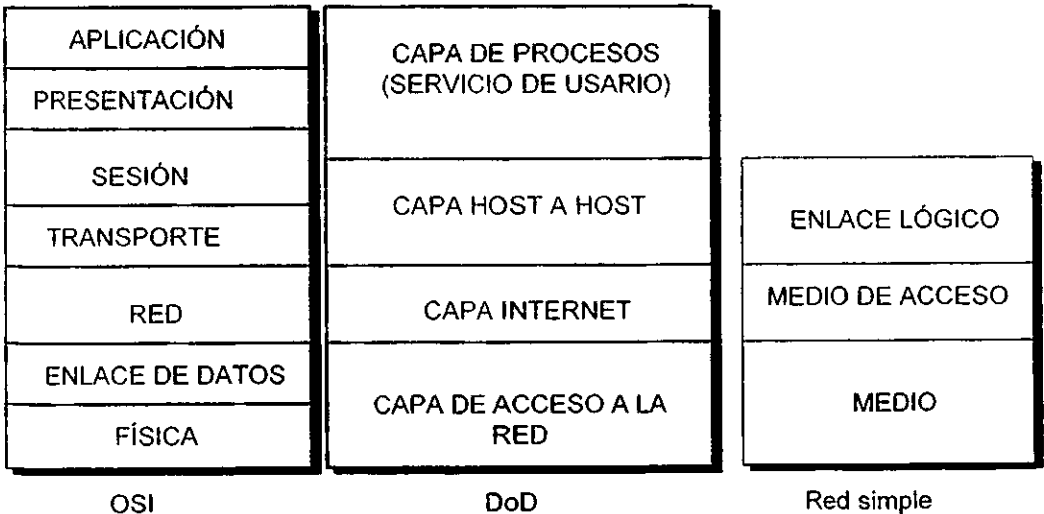


FIG 1.3.3.D MODELO DE CORRESPONDENCIA

Como se puede ver, no es una correspondencia uno a uno entre los modelos OSI y DoD. En consecuencia, no es sencillo mover aplicaciones entre redes basadas en estos dos modelos. Sin embargo, el gran uso que ha tenido TCP/IP ha hecho que varios vendedores desarrollen convertidores de protocolos que faciliten los servicios entre estos dos modelos. El modelo simple de red encaja muy bien en el nivel de acceso de red del modelo DoD. Así, la combinación de los modelos de red simple y DoD proporcionarán efectivamente todos los servicios definidos en el nivel siete del modelo OSI.

Modelo SNA

SNA es un modelo desarrollado por IBM para conectar sus equipos. SNA define un modelo completo de protocolos y direccionamiento basado en unidades lógicas (UL) y unidades físicas (UF), las cuales permiten escalabilidad y extensibilidad. La fig 1.3.3.D muestra el mapeo con el modelo OSI. SNA tiene también siete capas, pero tienen muchas diferencias entre ellas.

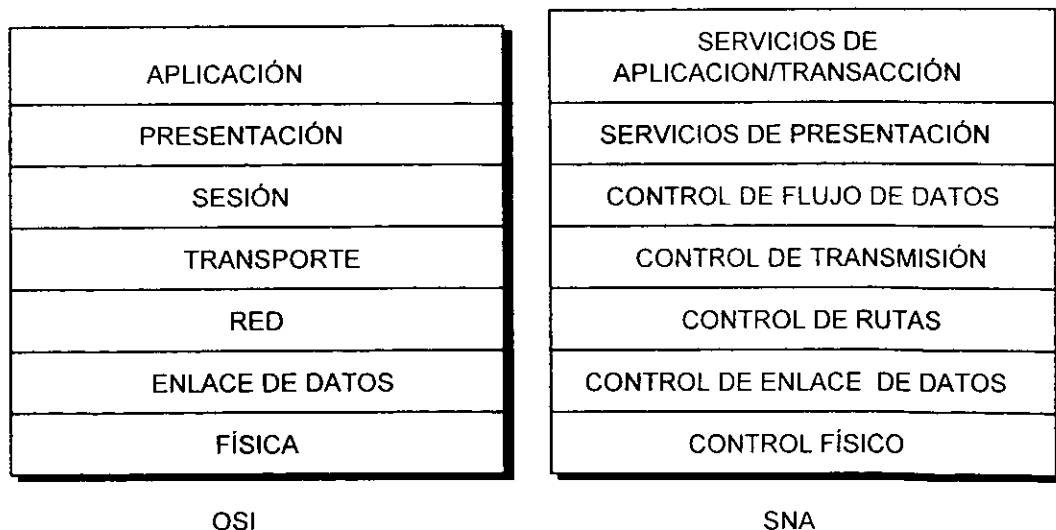


FIG. 1.3.3.E MODELOS OSI Y SNA

- Servicios de aplicación/transacción. Análogo a la capa de aplicación OSI, esta capa proporciona servicios como correo electrónico, así como de transferencia de archivos.
- Servicios de presentación. Este servicio mapea datos en formatos y representaciones que son entendidas por aplicaciones. El programa de interface común CPI-C es el estándar para servicios de presentación.
- Control de flujo de datos. Esta capa utiliza elementos de protocolos llamados cadenas y llaves para organizar y coordinar el flujo de datos y recuperación de errores entre puntos.
- Control de transporte. Los servicios proporcionados por esta capa son similares a la capa de transporte de OSI en las funciones de garantizar el paso y entrega extremo a extremo. Además, el control de transporte proporciona servicios de compresión e encriptamiento.
- Control de rutas. Proporciona servicios confiables orientados a conexión. Provee servicios punto a punto y administra las interfaces a varias subredes, incluyendo fragmentación y ensamblado.
- Control de enlace de datos. El protocolo de enlace de datos de SNA es SDLC, proporciona transmisión confiable entre sistemas. SDLC implementa un mecanismo primario / secundario en el cual un sistema toma el rol de maestro.
- Control físico. Como en todas las arquitecturas de red, maneja las características eléctricas y mecánicas de las líneas entre las estaciones.

1.3.4 TOPOLOGÍAS DE REDES LOCALES

Bus :

- Es la topología mas simple. Un cable lineal con varios dispositivos conectados a lo largo de el.
- Las transmisiones de un nodo viajan en ambos sentidos.
- Los nodos no retransmiten la información.
- Si un nodo falla no afecta el funcionamiento de la red.
- La ruptura del cable afecta a toda la red.

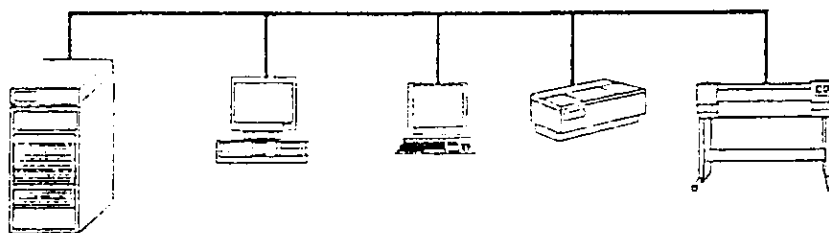


FIG 1.3.4 A TOPOLOGÍA DE BUS

Anillo / Estrella

- Los nodos se conectan en forma circular.
- Cada nodo retransmite la información al siguiente.
- Si un nodo falla, afecta el funcionamiento de la red.
- La ruptura de un cable afecta a toda la red.

- Los nodos se conectan a un concentrador central
- La falla de un nodo no afecta la red.
- La ruptura de un cable afecta solo al nodo conectado a el.

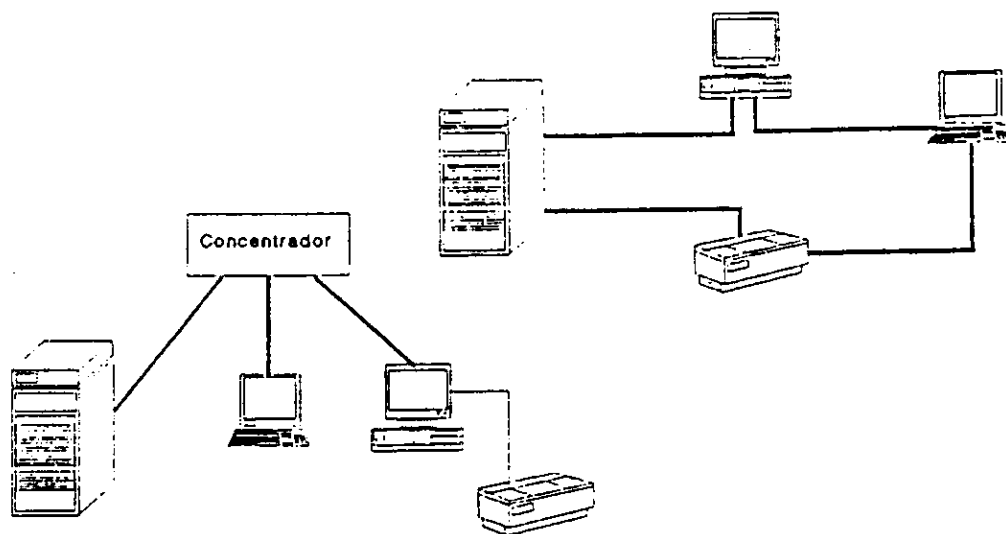


FIG. 1.3.4.B TOPOLOGÍA DE ANILLO Y ESTRELLA

Ventajas entre bus y anillo / estrella

Para un usuario con un gran número de dispositivos y gran capacidad de requerimientos la topología de bus es la más conveniente. Sin embargo, para requerimientos más moderados el anillo puede satisfacerlos bien.

La topología bus es más simple. Utiliza puntos pasivos en lugar de repetidores activos. No tiene la complejidad que implica los puentes y el cableado hacia el concentrador.

El beneficio mas importante de utilizar el anillo es que utiliza enlaces de comunicación punto a punto. Esto da ciertas ventajas. Primero la señal transmitida es regenerada en cada nodo, la transmisión de errores son minimizados y grandes distancias puedan ser

cubiertas. Segundo, la electrónica y mantenimiento de líneas punto a punto son más simples que líneas multipunto.

Red Ethernet

La red Ethernet o CSMA/CD (*Acceso Múltiple por Detección de Portadora con Detección de Colisión*) es la red más popular que se utiliza para transportar el protocolo TCP/IP. Esta red fue desarrollada por Xerox a principios de 1970.

Esta red local es basada en una topología de un medio único de comunicación llamado bus que permite el intercambio de mensajes entre cualquier par de equipos conectados al bus (conexión lógica punto a punto).

Entre otras características de la red ethernet tenemos :

- Es una arquitectura que provee servicio de datagramas, con detección de error pero no corrección de este.
- Utiliza topología de bus, multiacceso, con conmutación de paquetes y un medio de difusión pasivo.
- Las unidades de datos son transmitidas y llegan a todas las estaciones de red.
- Las estaciones son responsables de reconocer las direcciones contenidas en una unidad de datos.
- El servicio que ofrece esta red corresponde exactamente a los estándares OSI-ISO nivel 1 y 2, nivel 1 del modelo DoD, y son equivalentes a los 802.3 de IEEE.
- Esta red es utilizada por diferentes niveles de transporte tales como OSI nivel 3, IP (deTCP/IP), IPX (de Netware), etc.

El soporte físico de comunicación que utiliza es :

- cable: par trenzado, coaxial, fibra óptica.

- componente electrónico: repetidor, puente, "hub" (concentrador).

Ethernet puede utilizarse bajo cuatro tipos distintos de medios físicos(cables).

- 10Base5 : coaxial grueso.
- 10Base2: coaxial delgado.
- 10BaseT: par trenzado.
- Fibra Ethernet : fibra óptica.

Red Token Ring

Token ring (*Paso de Testigo en Anillo*) es el fruto de una investigación de los laboratorios de IBM en Zurich, Suiza y en Trangle Park Carolina del Norte, que fue anunciada comercialmente en 1985. Es una combinación entre la topología física de estrella con el flujo lógico de datos en forma de anillo. Esta red necesita un MAU (*Multiple Access Unit - Unidad de Acceso Multiple*) para poder realizar el anillo lógicamente y la estrella físicamente

Las ventajas de que tiene token ring son :

- Permite conexión de estaciones separadas hasta 300 mts, o en edificios separados.
- Permite conexión de estaciones y terminales a computadoras host sobre la misma red.
- Permite hacer fácilmente reconfiguraciones de la red, cambios en las localizaciones de estaciones o de otros recursos.
- Nodos defectuosos fácilmente detectados
- Cada nodo regenera la señal
- Cableado fácilmente modificado.

Y sus desventajas :

- Ocupan mas cable que otras topologías
- Conexión/desconexión de nodos genera errores.

Los tipos de cable que utiliza esta red son :

- Tipo 1 : STP 2 pares blindados.
- Tipo 2 : STP 2 pares blindados, 4 pares fuera del blindaje.
- Tipo 3 : UTP 2 pares.
- Tipo 5 : 2 fibras de 100/140 micras.
- Tipo 6 : STP 2 pares.

1.3.5 MEDIOS DE COMUNICACIÓN

Es un medio físico sobre el cual se transmiten los mensajes. Los medios tienen las siguientes características :

- No son 100 % confiables por el ruido y ruptura
- La transmisión requiere un tiempo (no es instantánea)

Los principales errores debidos a los medios de comunicación son :

- Perdida de mensajes
- Alteración de mensajes
- Retrasos
- Alteración de orden transmision-recepcion.
- Sobreflujo en las áreas de almacenamiento temporal de los mensajes.

Existen dos tipos de medios de comunicación:

- Cable
 - Par trenzado
 - Cable coaxial
 - Fibra Óptica
- Espectro Electromagnético
 - Vía Satélite
 - Estaciones Terrenas
 - Estaciones Personales (vsat)
 - Estaciones móviles
 - Vía Microondas(Radio)
 - Estaciones fijas
 - Estaciones móviles

- Infrarrojo

Par Trenzado

Características :

- El medio mas común
- de capacidad reducida para transmisión digital
- hasta decenas de Mbps/ hasta 100 metros.
- Conectores RJ-45, RJ-12, RJ-11 (8,6 y 4 hilos)

Tipos:

- Par trenzado no protegido (UTP)
- Par trenzado protegido (STP)

Cable coaxial

Características:

- Mayor ancho de banda (mayor capacidad)
- Excelente inmunidad al ruido
- Es usado para transmisión de video

Tipos:

- Banda base (baseband) 50 ohms
 - Transmisión digital
 - Hasta 10 Mbps/ 1 Km

- Banda ancha (Broadband) 75 ohms
 - Transmisión analógica
 - Hasta 300Mhz (150Mbps) / 100 Km

Fibra Óptica

Características :

- Transmisión a través de un rayo luminoso
- Un gran ancho de banda $10E8$ Mhz
- Hasta 1000 Mbps / 1 Km.

Tipos :

- Transmisión por
 - Diodo emisor de luz (led). Foto-diodo
 - Laser (ILD). Foto-transistor.
- Propagación
 - Multimodo (LED) con capacidad mínima de 3 Km.
 - Monomodo (ILD) con capacidad máxima hasta de 30 Km.

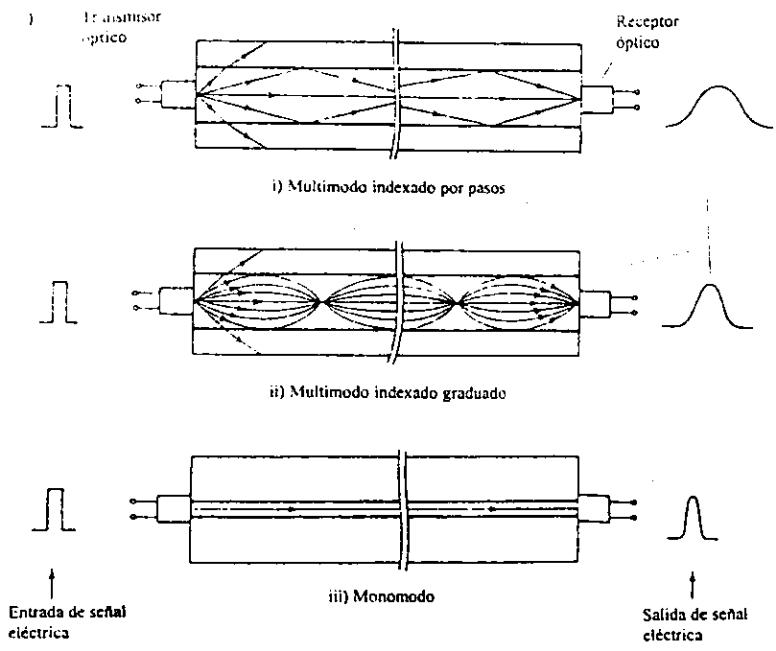


FIG. 1.3.5.A FIBRA ÓPTICA

Satélite

Características :

- Sensible a las condiciones atmosféricas
- Rango de frecuencias de 1 a 14.5 Ghz
- Existen tres bandas
 - C de 3.7 a 6.4 Ghz
 - Ku de 11.7 a 14.5 Ghz
 - L de 1.64 a 1.66 Ghz
- Requiere permiso de SCT. Alquiler del canal. Alto costo
- Transmisión por broadcast (difusión)
- Gran cobertura de transmisión (nivel nacional e internacional)
- Tiempo de propagación considerable (aprox 250 msg)
- Velocidades entre 64 Kbps-50Mbps.

Tipos:

- Estaciones terrestres
- Microestaciones terrestres (VSTA: Very Small Aperture Terminals)

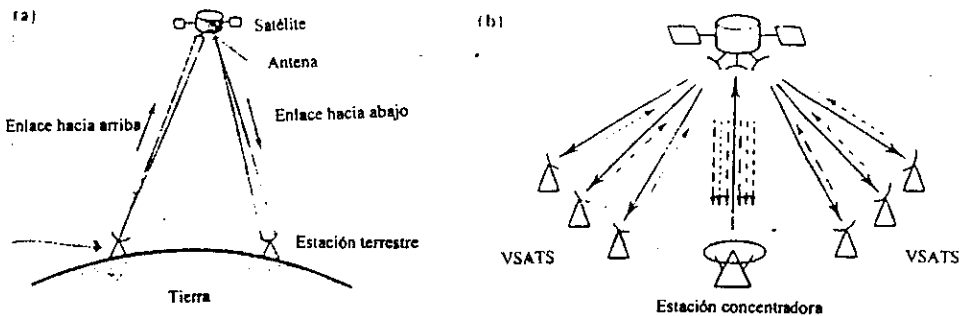


FIG. 1.3.5.B SATÉLITE

Radio

Características:

- Ancho de banda limitado
- Transmisión de banda limitado
- Alcance limitado a decenas de kilómetros
- Velocidades entre 100 Kbps-2Mbps
- Sensibles a los cambios atmosféricos
- Para conexiones punto a punto se requiere de una línea de vista sin obstáculos.

Tipos:

- Punto a punto
- Multipunto
- Difusión.

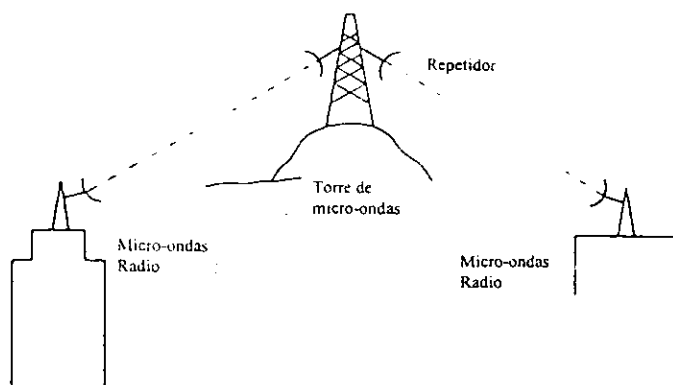


FIG. 1.3.5.C RADIO

Infrarrojo

Características:

- Tecnología reciente
- Alcance hasta decenas de Kms
- Conexión punto a punto
- Conexión a línea de vista

Tipos de transmisores:

- Láser con alcance hasta decenas de Kilómetros.
- Led con alcances hasta decenas de metros.

1.3.6 ELEMENTOS DE INTERCONEXIÓN

Repetidor :

- Permite la extensión de un medio
- El intercambio de realiza a nivel 1 físico.
- Con un repetidor se extiende un medio
- Con dos repetidores de utiliza otro medio intermedio

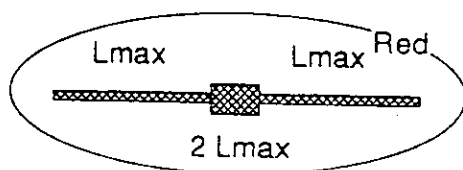


FIG. 1.3.6.A REPETIDOR

Puente

- Permite la extensión física (como repetidor) y lógica a nivel 2 OSI de enlace
- Sobre un mismo medio permite conectar/separar tráfico
- Sobre dos medios diferentes permite la interconexión de dos subredes físicas

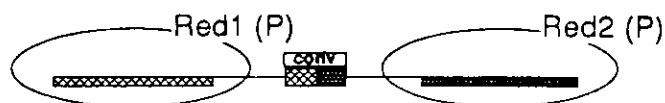


FIG. 1.3.6.B PUENTE

Enrutador

- Permite la interconexión de redes que pueden funcionar con medios y protocolos de enlaces diferentes.
- El intercambio de realiza a nivel 3 OSI de enrutamiento.
- Interconectan subredes para crear una *internet* (red de redes)
- Cuentan con la capacidad de escoger entre rutas alternas.



FIG. 1.3.6.C ENRUTADOR

Convertidor de protocolo

- Permite que se comuniquen dos aplicaciones soportadas por protocolos diferentes. Ejemplo TCP/IP <-> SNA
- El intercambio de realiza a nivel 7 de aplicación.

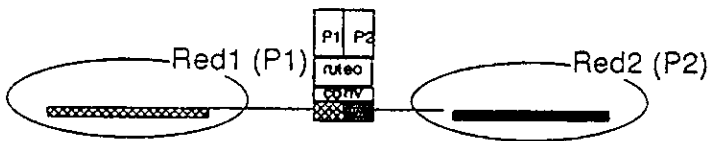


FIG. 1.3.6.D CONVERTIDOR DE PROTOCOLOS

1.3.7 PROTOCOLOS

Un protocolo es un conjunto de reglas de un dialogo entre entidades comunicantes, expresado a través del envío y recepción de mensajes.

TCP/IP

Es una colección de protocolos de alto nivel, el cual puede operar sobre diferentes medios, incluyendo par trenzado, coaxial o fibra óptica, usando varios protocolos de bajo nivel como son X.25, ethernet o token ring. El protocolo de bajo nivel mas usado por TCP/IP es ethernet.

Varios servicios forman parte de TCP/IP como son :

- FTP (*File Transfer Protocol - Protocolo de transferencia de archivos*). Transferencia de archivos
- TELNET . Acceso remoto de terminales.
- SMTP .Transferencia de correo electrónico
- NFS (*Network File System - Sistema de archivos en red*). Este servicio proporciona a multiples usuarios acceso del mismo archivo desde diferentes sistemas en donde el usuario aparentemente lo esta accedando en forma real y local.
- Servidor de nombres que administra un sola base de datos la cual mantiene contraseñas, inicializa usuarios, nombres y direcciones de redes para una red completa.
- SNMP (*Simple Network Management Protocol - Protocolo simple de administración de la red*) que es usado para administrar dispositivos conectados de manera remota. Este protocolo utiliza a UDP como transporte para proveer comunicación entre la estación administradora de la red y un agente administrador de red en el objeto administrado.

- Ejecución remota. Permite que un programa sea ejecutado desde otra maquina. de manera remota.

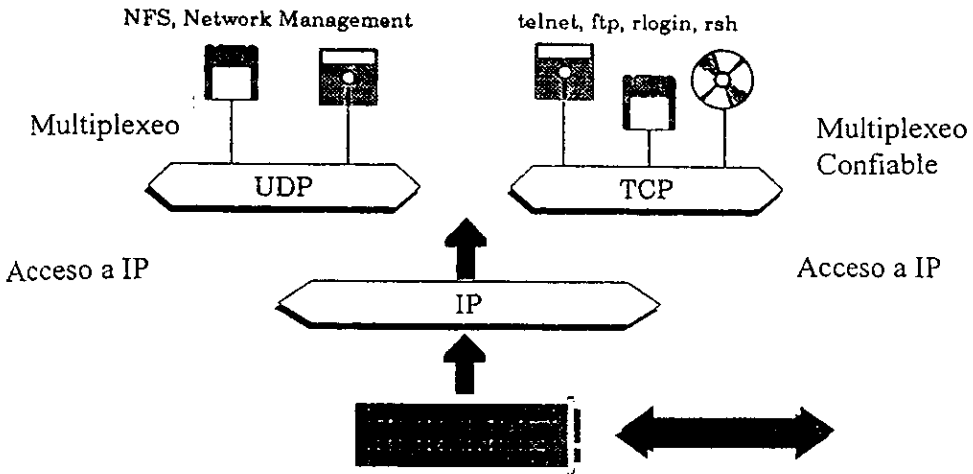


FIG. 1.3.7.A SERVICIOS DE TCP/IP

Los principales protocolos son IP a nivel de red y, TCP y UDP (User Datagram Protocol - Protocolo de Datagramas de Usuario) a nivel de transporte.

Las principales funciones de IP son :

- Encaminar una unidad de datos de una computadora fuente a una computadora destino dentro de una red de redes.
- Detectar encabezados IP corruptos a través de un algoritmo de revisión. Esto hace seguro que un paquete no pueda llegar a otro destinatario.
- Rutear y recibir datagramas a la capa más alta correcta. El encabezado IP contiene un identificador de protocolo. Este identificador podría especificar TCP pero hay otros protocolos que utiliza IP, así IP puede informar el protocolo destino de la llegada de un datagrama.

- Fragmentación de datagramas

Las principales funciones de TCP son :

- Fragmentación de mensajes a segmentos.
- Da secuencia a segmentos para que el receptor pueda ensamblar el mensaje. Los segmentos por la red pueden tomar diferentes rutas y llegar al receptor en diferente orden.
- Detección de mensajes corruptos o perdidos usando los numeros de secuencia, temporizadores y revisión de errores.
- Retransmisión de segmentos perdidos y corruptos.
- Control de flujo.

TCP y UDP permiten el intercambio de mensajes entre dos procesos ubicados en computadoras diferentes conectadas en una red. A este nivel el enrutamiento de mensajes es transparente. TCP es un protocolo orientado a conexión, mientras que UDP no necesita establecer una conexión. UDP es una alternativa a TCP, con UDP un mensaje siempre cabra en un solo datagrama. UDP es menos poderoso que TCP ya que no maneja segmentación o reensamblado de mensajes, ni detección de errores.

IPX/SPX

Es el protocolo usado por el popular sistema de red Netware. IPX (*Internet Packet Exchange - Intercambio de Paquetes en la Red de redes*) es el protocolo a nivel red, mientras que SPX (*Sequenced Packed Exchange - Intercambio de Paquetes en Secuencia*) da servicios de transporte.

Sobre IPX/SPX existen 4 protocolos punto a punto: Netbios, Pipes nombrados, TLI (*Transport Layer Interface - Interface de la Capa de Transporte*) y apis de IPX/SPX. Estos protocolos son soportados en ambientes DOS, Windows 95/NT y OS/2.

NetBios y NetBeui

Netbios fue el primer protocolo para redes locales, introducido por IBM en 1984. Netbios es usado como interface de una variedad de stacks de redes incluyendo: Netbeui (de IBM y Microsoft), TCP/IP, Vines, OSI y IPX/SPX.

Netbeui es el protocolo que viene generalmente con los productos de redes locales de IBM y Microsoft, entre estos productos están : Windows para trabajo en grupo, Lan-Manager, Lan-Server y SQL-Server.

1.3.8 TENDENCIAS DE LAS REDES.

El Objetivo básico de una red es transmitir información entre las computadoras conectadas a ellas. Gracias a la tecnología de los sistemas abiertos ahora es posible ofrecer este servicio básico independientemente de la tecnología de los medios de comunicación y computadoras que conforman una red.

La tendencia actual es que las redes ofrezcan nuevos servicios de valor agregado tales como :

- Teleconferencia
- Redes inteligentes
- Redes de transacciones (EDI).
- Sistemas de información mundiales(Internet).
- Redes de servicios integrados (ISDN B-ISDN)
- conexiones no fijas (computación móvil).



FIG. 1.3.8.A TENDENCIAS DE LAS REDES

1.4 ARQUITECTURA CLIENTE/SERVIDOR

1.4.1 DEFINICIÓN CLIENTE/SERVIDOR

Como su nombre lo indica, clientes y servidores son entidades lógicas separadas que trabajan juntas sobre una red para completar una tarea. El computo cliente/servidor se refiere a la automatización de procesos de negocio a través de recursos de computo que no se encuentran confinados en una sola localidad. El beneficio que nos proporciona esta arquitectura es que las aplicaciones pueden ajustarse de mejor manera a los requerimientos del negocio y potencialmente pueden desarrollarse de una manera más rápida y barata que las aplicaciones basadas en los tradicionales centralizados multiusuario.

Como regla general, una aplicación cliente/servidor es un software ligado entre una estación "cliente" (con su usuario humano asociado) y un servidor que puede ser una base datos. El modelo también incluye computadoras servidores que pueden suministrar a clientes con otras clases de servicios, por ejemplo, un servidor de aplicación, un servidor de seguridad, etc. El software de cliente proporciona una interfaz de usuario efectiva (usualmente emplea gráficos y un ratón) en coordinación con software sobre el servidor (típicamente una base de datos relacional) para satisfacer una necesidad de negocio.

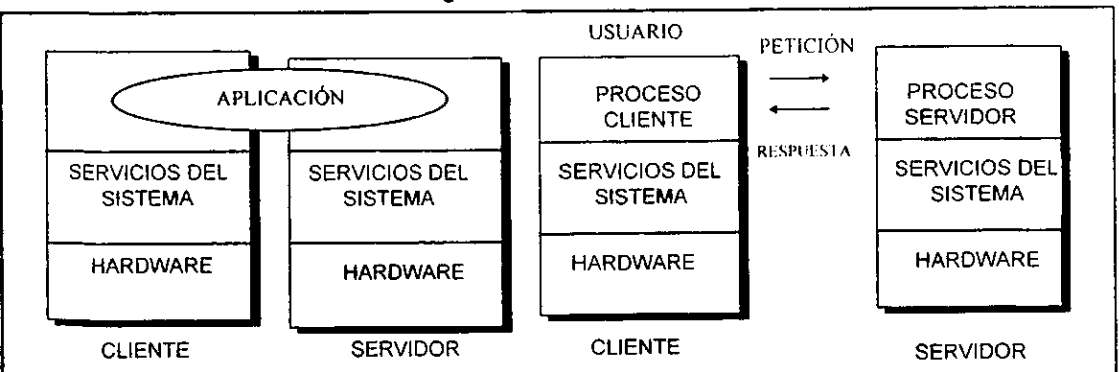


FIG 1.4.1.A SISTEMA CLIENTE/SERVIDOR

1.4.2 SISTEMAS TRADICIONALES Y SISTEMAS CLIENTE/SERVIDOR

Los sistemas tradicionales generalmente se han basado en sistemas multiusuarios o mainframes usando una terminal tonta como interfaz hacia el usuario. Este esquema no maneja capacidades de computo cooperativo, pues toda la aplicación reside en el sistema central.

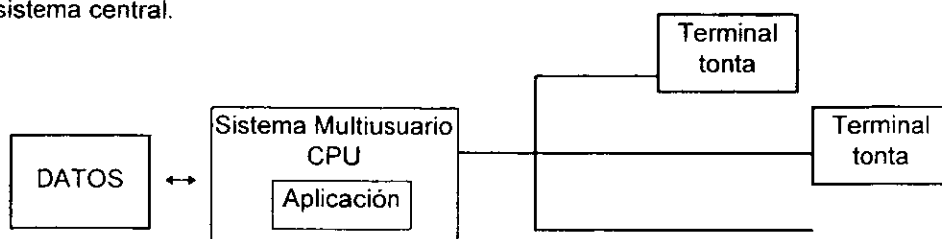


FIG. 1.4.2.A SISTEMAS TRADICIONALES BASADOS EN SISTEMAS MULTIUSUARIOS.

El computo cliente/servidor es un movimiento irresistible que esta cambiando la manera en que las computadoras están siendo utilizadas. Ver FIG. 1.4.2.A

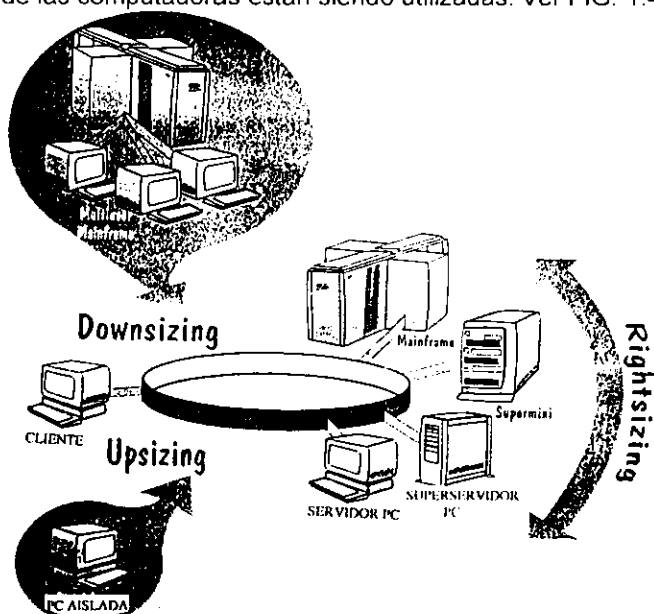


FIG. 1.4.2.A TRANSFORMACIÓN DE SISTEMAS TRADICIONALES HACIA CLIENTE/SERVIDOR

Los cambios tecnológicos y las necesidades empujan hacia el computo cliente/servidor desde tres diferentes direcciones: downsizing, upsizing y rightsizing.

Downsizing

Es la migración hacia abajo de aplicaciones tradicionales de mainframes y minicomputadoras multitarea hacia PCs y estaciones de trabajo Unix. El proceso de downsizing rompe las grandes aplicaciones de mainframes en módulos de programas que corren sobre uno o mas servidores. Las funciones de interfaz hacia el usuario se mueven hacia estaciones de trabajo clientes y reemplazan las terminales tontas tradicionales. Los procesadores centralizados para compartir terminales son reemplazadas por equipos de red cliente/servidor..

Upsizing

Es la migración de PC's aisladas hacia un nivel departamental de trabajo en grupo en red. Las primeras PC's que eran conectadas en red solo compartían dispositivos muy caros como impresoras láser y digitalizadores. En la actualidad las redes locales son usadas principalmente para envío de correo electrónico y para compartición de archivos y bases de datos (incluyendo repositorios de imágenes y documentos). Además, ya existe un nueva generación de software cliente/servidor para trabajo en grupo que ofrece grandes ventajas.

Rightsizing

El rightsizing mueve las aplicaciones a la plataforma de servidor mas apropiada. Los clientes piden servicios sobre la red y el mejor servidor proporciona el servicio. En este modelo abierto un servidor puede ser una PC, una minicomputadora, o un mainframe. Los servidores de diferentes proveedores pueden convivir.

1.4.3 ARQUITECTURA TRES CAPAS.

La mayoría de las organizaciones se han comenzado a mover hacia esta arquitectura. Su raíz se basa en la arquitectura maestro esclavo con mucho mas capacidades de proceso cooperativo, los recursos de computo son distribuidos verticalmente. Especificamente la capa mas alta es usualmente ocupada por el sistema mas poderoso que ocupa los datos corporativos, es decir el mainframe. La segunda capa contiene el poder de los servidores de red, los cuales tienen dos propiedades: ellos actúan como clientes de la capa mas alta enviando peticiones al mainframe. Al mismo tiempo, ellos funcionan como servidores de las estaciones de trabajo que residen en la tercera capa..

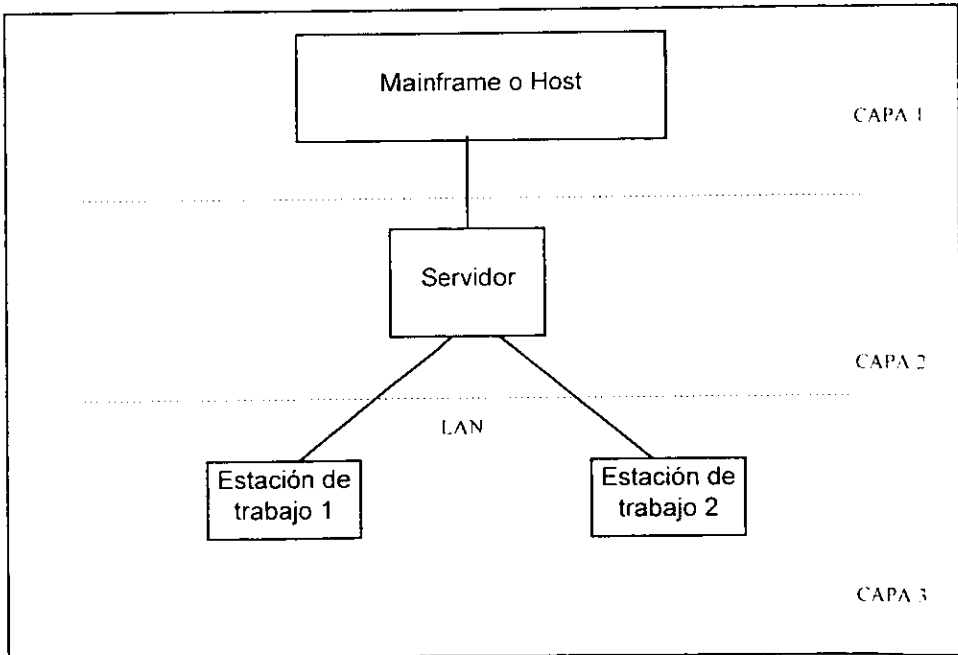


FIG. 1.4.3.A ARQUITECTURA TRES CAPAS

Además esta arquitectura puede ser extendida horizontalmente agregando mainframes a la capa superior, y redes locales y servidores a la segunda y tercera capa.

En una arquitectura de este tipo generalmente se originan las siguientes preguntas:

- En cual capa (o capas) deberán estar los datos.
- En cual capa (o capas) deberá ser puesta la lógica de la aplicación.
- En cual capa (o capas) deberá ser puesta la interfaz del usuario.

Una aplicación típica consiste de los siguientes componentes:

- **Lógica de proceso de presentación :** Esta es una parte del código de la aplicación que interactúa con un dispositivo tal como una estación de trabajo o terminal. Esta lógica ejecuta tareas como formato de pantallas, lectura y escritura de la información sobre la pantalla, manejo de ventanas, teclado y ratón. Existen algunas interfaces de usuario gráficas que proporcionan grandes facilidades., entre estas interfaces tenemos Presentation Manager de OS/2, Windows de Microsoft, X Windows y Motif de OSF para ambientes Unix.
- **Lógica de proceso del negocio:** Esta parte de aplicación utiliza la entrada de datos (de la pantalla y/o base de datos) para ejecutar tareas del negocio. Típicamente, esta parte es código escrito en lenguajes de tercera o cuarta generación.
- **Lógica de proceso de la base de datos:** Es una parte de la aplicación que manipula datos en la aplicación. Los datos son administrados por un DBMS (sistema administrador de base de datos). La manipulación de datos en una base de datos relacional es hecha usando algún dialecto del sql (Structured Query Language - Lenguaje estructurado de Consultas). DML (Data Manipulation Language - Lenguaje de manipulación de datos) es típicamente incrustado dentro del código de la aplicación.

- **Proceso de la base datos:** Este es el proceso actual de la base de datos que es ejecutado por el DBMS. Idealmente el proceso de la base es transparente a la lógica del negocio de la aplicación, Sin embargo, desde el punto de vista arquitectura, este proceso es una parte esencial del las interacciones del proceso cooperativo, y deberán ser consideradas como un componente de la aplicación completa.

La arquitectura cliente / servidor emplea el proceso cooperativo distribuido para distribuir componentes de la aplicación entre clientes (típicamente presentación y alguna parte de la lógica del negocio) y servidores (regularmente algunas partes de la lógica del negocio, lógica de la base datos y el DBMS).

Para responder las preguntas que se hicieron con anterioridad respecto a la distribución de los componentes de una aplicación de tres capas o cliente/servidor se sugieren algunas recomendaciones:

- En general, el componente de la presentación con sus facilidades de entrada y salida a una pantalla son puestas en el sistema cliente, y estos clientes son típicamente puestos en la capa mas baja.
- Dada la capacidades de poder de las estaciones de trabajo, y el hecho de que la presentación reside en los sistemas cliente, también se puede poner alguna parte de la lógica del negocio en el cliente.
- Si la lógica de proceso de la base datos es incrustada en la logica del negocio, y si clientes mantienen alguna baja interacción, datos casi estáticos, entonces alguna parte de la lógica del proceso de la base puede ser puesta en el cliente.
- Dado que una red típica conecta clientes en un grupo de trabajo común, y asumiendo que el grupo de trabajo comparte una base de datos, fragmentos compartidos de la lógica del negocio y de la base datos, así como el DBMS deberán ser puestos en el servidor

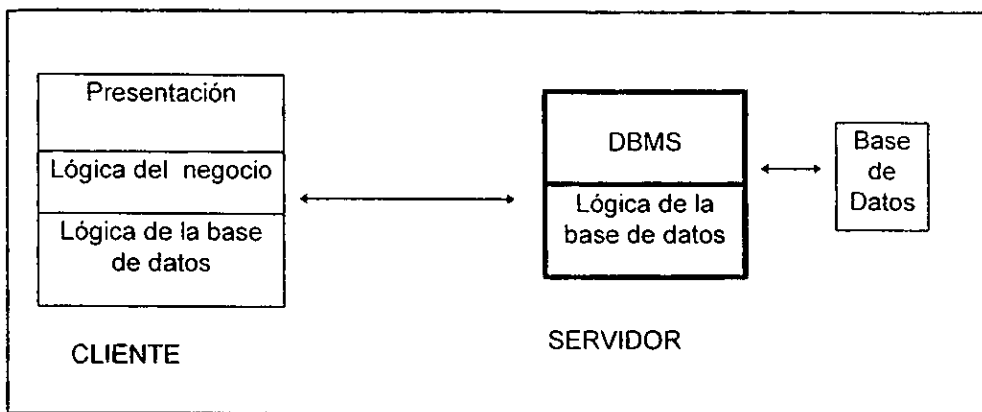


FIG. 1.4.3 A CLIENTE/SERVIDOR CON DATOS UNICAMENTE EN EL SEVVIDOR

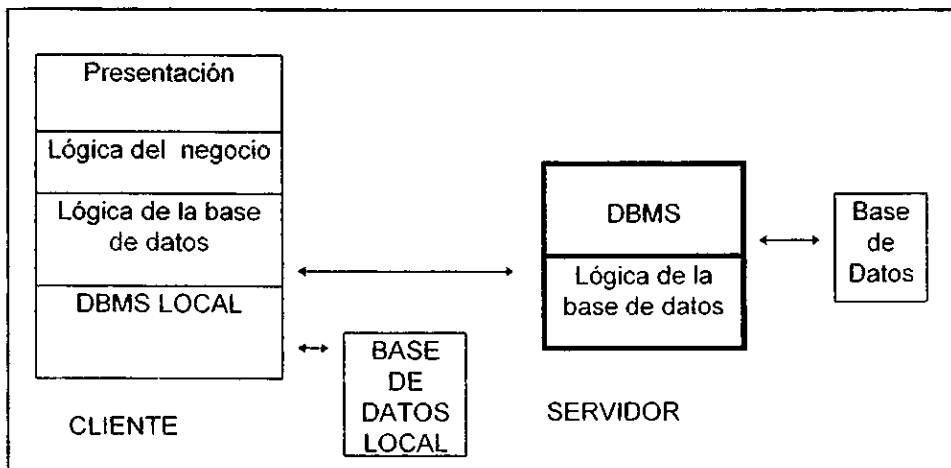


FIG. 1.4.3.B CLIENTE/SERVIDOR CON DATOS EN AMBOS LADOS

1.4.4 CARACTERÍSTICAS DE LA ARQUITECTURA CLIENTE/SERVIDOR

Los sistemas cliente/servidor tienen las siguientes características:

- **Servicio** : Cliente/servidor es principalmente una relación entre procesos corriendo sobre diferentes maquinas. El proceso servidor es un proveedor de servicios. El cliente es un consumidor de servicios. En esencia, cliente/servidor proporcionan una clara separación de funciones basadas en la idea de servicio.
- **Recursos compartidos** : Un servidor puede atender a muchos clientes al mismo tiempo y regular accesos a los recursos compartidos.
- **Protocolos asimétricos** : Existe una relación muchos a uno entre clientes y servidor. Los clientes siempre inician el dialogo pidiendo un servicio. Los servidores están pasivos esperando peticiones de los clientes.
- **Transparencia de localidad** : El servidor es un proceso que puede residir sobre la misma maquina que el cliente o sobre diferentes maquinas a través de la red.
- **Intercambios basados en mensajes** : Clientes y servidores son sistemas acoplados que interactuan a través del mecanismo del paso de mensajes. Este mensaje es el mecanismo de entrega de peticiones y respuestas.
- **Escalabilidad**: Los sistemas cliente servidor pueden ser escalados horizontalmente o verticalmente. Horizontalmente significa agregar o quitar estaciones de trabajo clientes con el impacto tal vez del desempeño. Verticalmente es migrar a un servidor mas grande y rápido.

Un cliente por si solo tiene las siguientes características :

- ProActivo, envía peticiones al servidor
- Inicio/Fin Fijo
- Mantiene dialogo con el usuario
 - Manejo de pantallas y ventanas a través de interfaces gráficas de usuario (GUI)
 - Interpretación de comandos y menús
 - Entrada por teclado y ratón
 - Captura de datos y validación
 - Procesamiento de ayudas
 - Recuperación de errores

Un servidor por si solo tiene las siguientes características :

- Reactivo, disparado por peticiones de clientes
- Corre por siempre
- No interactua con otros servidores
- Tiene una función especifica
- Administración de acceso y control de las bases de datos.
- Administra recursos compartidos :
 - Comunicaciones
 - Discos/datos
 - Procesador
 - Impresoras

Específicamente, las aplicaciones cliente/servidor para su implementación requieren de alguno o de todos los siguientes elementos de computo distribuidos:

- Conectividad de red entre clientes y servidores
- Interoperatividad entre clientes y servidores
- Un mecanismo de llamados a procedimientos remotos (RPC)
- Procedimientos almacenados y triggers (*disparadores*)
- Soporte de sql
- Replicación de datos
- Procesamiento transaccional
- Administración de datos remotos
- Seguridad (autenticación y control de acceso)

1.4.5 TIPOS DE SERVIDORES

- **Servidor de archivos** : el cliente pasa peticiones de registros de archivos sobre una red al servidor. Esta es una manera muy primitiva del servicio de datos que necesita mucho intercambio de mensajes para encontrar los datos pedidos. Los servidores de archivos son útiles para compartir archivos a través de la red. Ellos son indispensables para crear repositorios compartidos de documentos, imágenes y otros grandes objetos de datos.
- **Servidor de base de datos:** el cliente pasa su petición sql como mensaje al servidor. El resultado de cada comando sql es regresado sobre la red. El código que procesa la petición sql y los datos residen en la misma maquina. El servidor utiliza su propios recursos para encontrar los datos pedidos y pasar todos los registros encontrados de regreso al cliente. En esta arquitectura el resultado es mucho mas eficiente, ya que existe una mejor distribución del procesamiento.
- **Servidor de transacciones:** el cliente invoca procedimientos remotos que residen en el servidor sobre un engine (*motor*) de base de datos. Estos procedimientos remotos en el servidor ejecutan un grupo de sentencias sql (en ocasiones llamados Store Procedure (*procedimientos almacenados*)). El intercambio en la red consiste de un mensaje de petición / respuesta. El grupo de sentencias sql todas son exitosas, o bien todas fallan como una sola unidad. Este grupo de transacciones también son llamadas transacciones. En un servidor de transacciones se puede crear una aplicación escribiendo código en ambos componentes (servidor y cliente).
- **Servidor de trabajo en grupo:** esta enfocado a la administración de información semi-estructurada como texto, imágenes, correo y el flujo de trabajo. Este tipo de servidores ponen a personas de manera directa con otras personas.

- **Servidor de objetos:** Aquí la aplicación cliente/servidor es escrita como un conjunto de objetos que se comunican. Los objetos clientes se comunican con objetos servidores usando ORB (*Object Request Broker - Despachador de peticiones de objetos*). El cliente invoca un método soportado por una clase en un servidor de objetos.

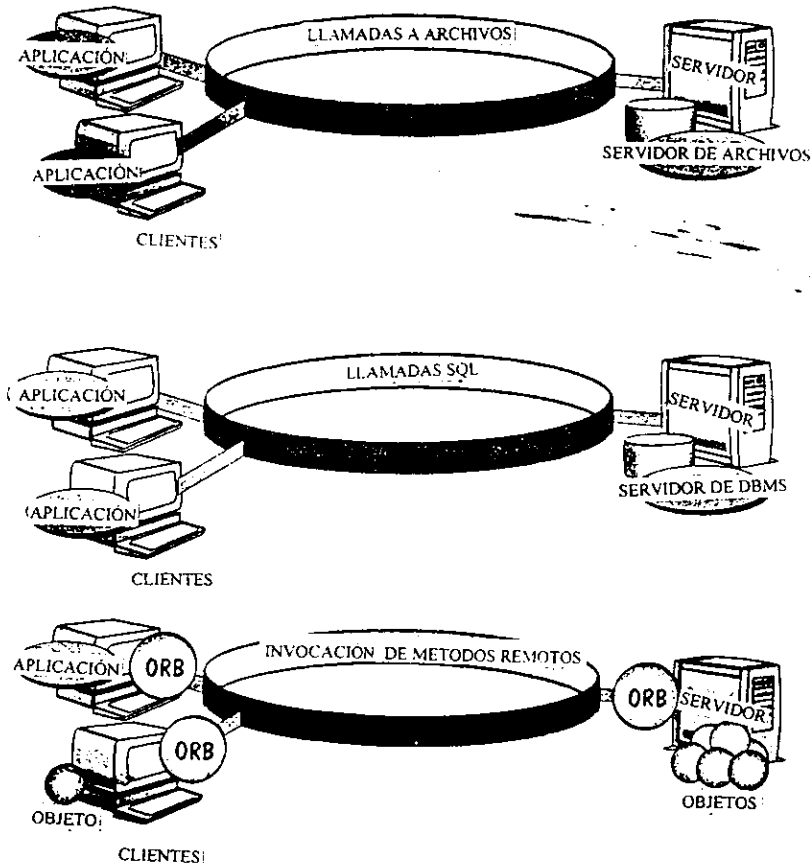


FIG. 1.4.5.A ALGUNOS TIPOS DE SERVIDORES

1.4.6 MIDDLEWARE (INFRAESTRUCTURA CLIENTE/SERVIDOR)

Middleware se refiere al software que se encuentra en medio del sistema cliente/servidor. Para hablar de middleware lo podemos dividir en tres capas: Los stacks (pilas) de transporte, el sistema operativo de red y los servicios específicos de middleware.

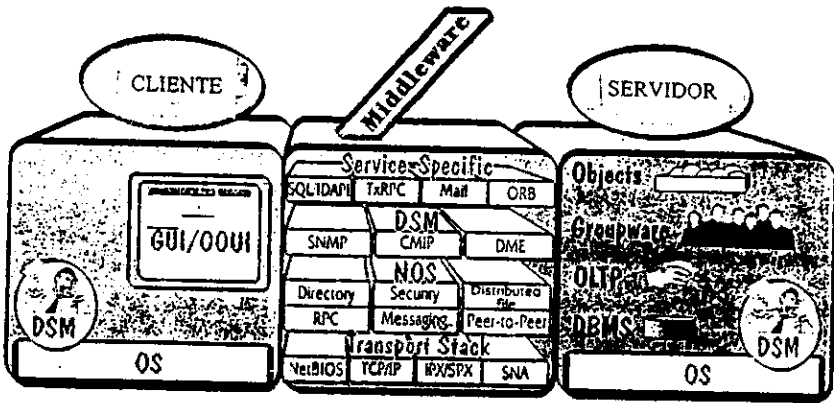


FIG. 1.4.6.A INFRAESTRUCTURA CLIENTE/SERVIDOR

El Middleware de stacks de transporte

El stack de transporte consiste de protocolos de transportes, incluyendo TCP/IP, NetBios, IPX/SPX, DECNet, OSI, que proporcionan comunicación punto a punto

confiable a través de redes de área amplia y redes locales a través de sus dispositivos de interconexión como enrutadores, puentes y convertidores de protocolos.

El middleware de sistema operativo de red

Tiene las siguientes características:

- Extiende el alcance local del sistema operativo para incluir dispositivos como impresoras, archivos de directorios y pilas de modems.
- Soporta la coordinación entre aplicaciones en la red. Esto se hace a través de RPC (*Remote Procedure Call - Llamados a Procedimientos Remotos*) y de middleware orientado a mensajes. La figura 1.4.6.B muestra una implementación de un RPC.

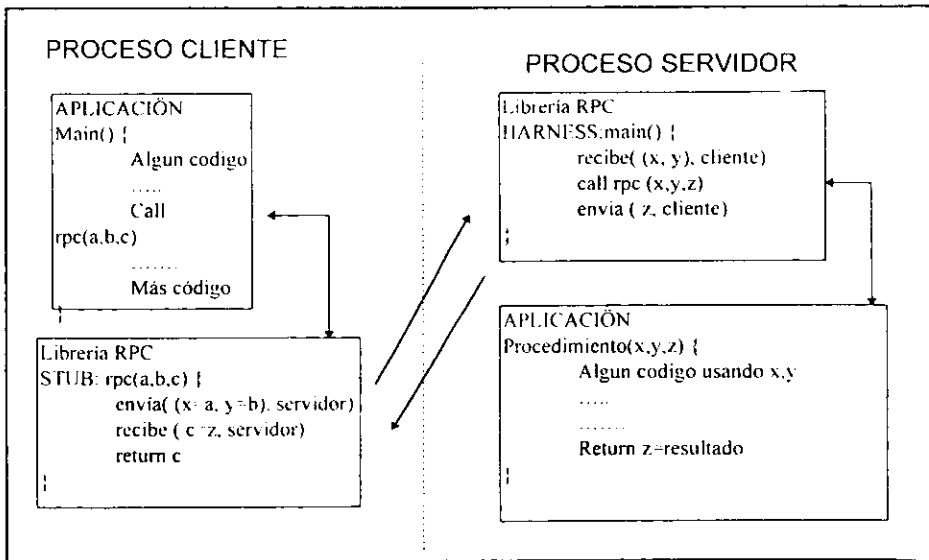


FIG. 1.4.6.B IMPLEMENTACIÓN SENCILLA DE UN RPC

Los RPC tienen las siguientes características :

- No hay espacio común de direcciones(no hay variable globales, no se pueden pasar apuntadores o direcciones)
- Existen problemas con listas, arboles y otras estructuras recursivas.
- Se necesita de una descripción completa de cada estructura con lenguajes especiales como Courier o clases de C++.
- Su overhead es de 6,000 msec. Un procedimiento local lo tiene de 10 msec, mientras que un acceso a disco es 15,000 msec.

El middleware de servicios específicos

Existen cinco servicios específicos:

- **Middleware de base de datos:** permite a clientes invocar servicios basados en sql a través de diversos proveedores de bases de datos. El middleware de base de datos es definido por estándares de facto como ODBC, IDAPI, DRDA, RDA, Oracle glue.
- **Middleware transaccional :** permite a clientes invocar servicios a través de múltiples servidores de transacciones. Los monitores de transacciones garantizan la integridad de todas las actividades en los servidores.
- **Middleware para trabajo en grupo:** permite a clientes invocar servicios sobre un servidor de trabajo en grupo. Entre estos servicios esta el correo electrónico, calendarización de actividades, etc. El estándar para correo electrónico es MAPI, para calendarizar es VIC.
- **Middleware de objetos :** permite que clientes invoquen métodos que residen sobre servidores remotos. El principal middleware es CORBA.
- **Middleware de administración de sistemas distribuidos:** permite administrar y monitorear estaciones y dispositivos. El estándar usado en esta área es SNMP y CMIP.

1.4.7 MONITOR DE TRANSACCIONES

El primer monitor de transacciones apareció sobre mainframes para proporcionar un medio ambiente robusto en tiempo real que pudiera soportar aplicaciones OLTP (*On Line Transaction Processing - Procesamiento de Transacciones en Línea*) de gran escala (reservaciones aéreas y de hotel, bancarias, autorizaciones de tarjetas de crédito, etc.).

Históricamente los monitores de transacciones fueron introducidos para correr aplicaciones que atendían a cientos y a veces a miles de clientes. Ya que si cada uno de estos miles de clientes necesitaba recursos (una conexión de red, medio megabyte de memoria, uno o dos procesos y una decena de archivos abiertos) se necesitaría un mainframe como servidor, el cual podría llegar a caerse.

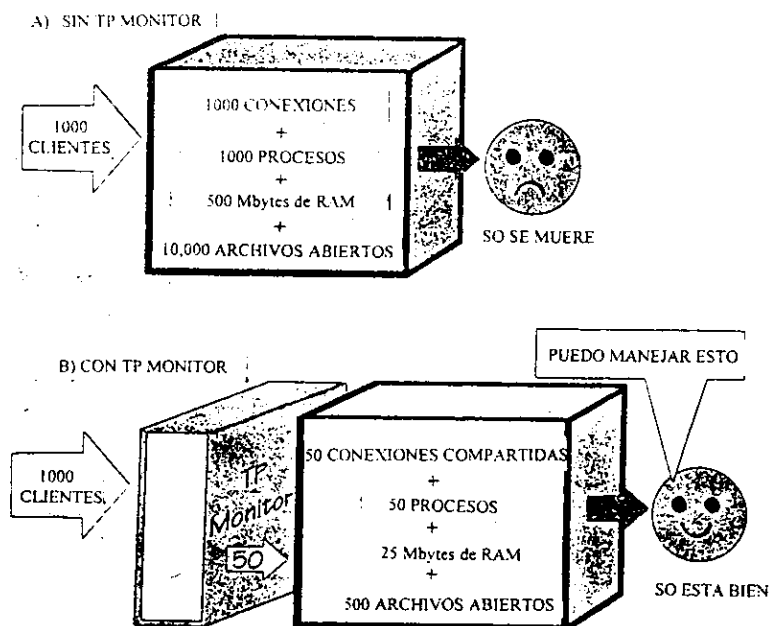


FIG. 1.4.7.A JUSTIFICACIÓN DE UN MONITOR DE TRANSACCIONES

Un monitor de transacciones puede ser definido como un sistema operativo para el procesamiento de transacciones en tiempo real. Un monitor de transacciones se caracteriza por manejar muy bien los siguientes conceptos:

- Administración de procesos: incluye el inicio de procesos en los servidores, balanceo del trabajo entre ellos y monitoreo de su ejecución
- Administración de transacciones: significa que va a garantizar las propiedades de todos los programas que corren bajo su protección.

Modelo del monitor de transacciones

Los monitores de transacciones necesitan estándares, ya que las aplicaciones que ellos coordinan pueden estar corriendo en diferentes plataformas con accesos a diferentes bases de datos y administradores de recursos. Estos estándares vienen de dos fuentes :

- ISO con sus especificaciones OSI-CCR y OSI-TP
- DTP "Distributed Transaction Processing" (Procesamiento de Transacciones Distribuidas) con su especificación X/Open.

X/Open tomo la iniciativa de definir apis en un marco de trabajo general para procesamiento de transacciones. Juntos, X/Open DTP y OSI-TP forman la fundación para el modelo de administración de transacciones abiertas. Este modelo define cuatro componentes :

- Un administrador de recursos: es una pieza de software que administra recursos compartidos. Por ejemplo un administrador de bases de datos que permite su actualización coordinandose vía el protocolo commit de dos fases.
- Un administrador de transacciones: es el componente que coordina y controla al administrador de recursos a través de apis establecidas para esta función

- Un administrador de programas: utiliza apis de propósito general suministradas por el administrador de recursos (ejemplo, sql)
- Un administrador de recursos de comunicación: que controla la comunicación entre aplicaciones distribuidas. También existen una serie de apis para interactuar entre el administrador de comunicaciones con el administrador de transacciones. Estas apis permiten conocer sobre cual administrador de transacciones remoto se encuentra la transacción.

En la figura 1.4.7.B se muestra la interacción entre estos cuatro componentes.

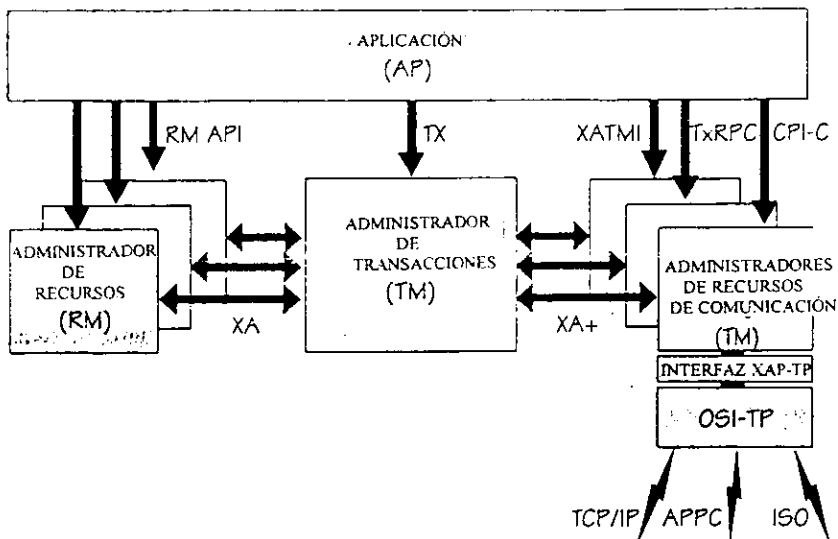


FIG. 1.4.7.B MODELO DE REFERENCIA DEL PROCESAMIENTO DE TRANSACCIONES.

1.4.8 CARACTERISTICAS DE ALGUNOS MONITORES DE TRANSACCIONES.

Tuxedo

Tuxedo es un monitor de transacciones de USL (*Unix System Laboratories - Laboratorios de Sistemas Unix*) basado principalmente para correr sobre Unix en sistemas con arquitectura cliente/servidor, sin embargo en la actualidad opera en medios ambientes heterogéneos cliente/servidor que incluyen estaciones de trabajo, servidores y computadoras mainframes. Actualmente Tuxedo es soportado por varias plataformas de hardware incluyendo Amdahl, AT&T, Pyramid, Sequent, Tandem y Unisys. La mayoría de vendedores de bases de datos en Unix han anunciado su compatibilidad con Tuxedo. Entre estos bases de datos se encuentran Oracle, Informix, Ingres, y Sybase. Tuxedo tiene además las siguientes funciones:

- Envío y recepción de mensajes entre clientes y servidores a través de la red.
- Administración del flujo de transacciones.
- Interacción con administradores de recursos (por ejemplo sistemas de bases de datos "DBMS")
- Definición y soporte de transacciones globales que abarcan múltiples sistemas distribuidos
- Maneja un modelo de tres capas.

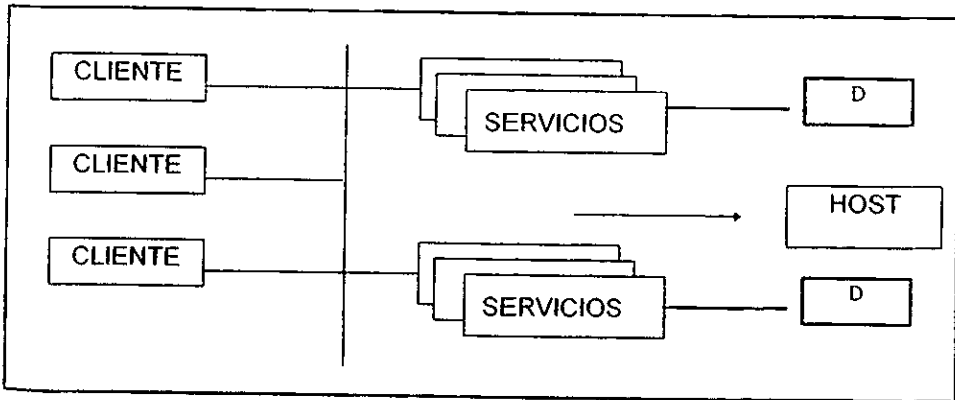


FIG. 1.4.8.A MODELO DE TRES CAPAS DE TUXEDO.

- Las reglas del negocio son accesadas como servicios disponible al cliente. El cliente envía una petición al servicio apropiado enviando un mensaje. Entonces el cliente recibe y procesa las respuestas del componente que dio el servicio.
- En aplicaciones de gran escala, muchos clientes concurrentemente se conectan y comparten los componentes y sus servicios encapsulados.
- Las rutinas de acceso a datos se encapsulan en el servidor en forma de sentencias sql incrustadas o en lógica de procedimientos almacenados. Los clientes consultan y actualizan datos en forma de petición de servicio.

Top End

Este monitor es diseñado por NCR Corporation y es el principal competidor de Tuxedo. Algunos expertos consideran que este monitor eventualmente remplazara a Tuxedo por ser mas robusto. Top End tiene las siguientes características :

- Soporta la administración de servicios de transacciones de una manera que requiere un mínimo de programación.
- Soporta el modelo X/Open DTP.
- Integra la mayoría de los productos de proveedores de bases de datos en Unix, desarrollando una estrategia de relación con ellos.
- Implementación de cualquier servicio de aplicación en lenguajes como : C, C++ y Cobol. Además tiene facilidades para depuración de aplicaciones distribuidas.
- Sistemas de administración amigable, incluyendo definición interactiva del sistema con definición de integridad y control.

La arquitectura TOP END esta basada en un modelo cliente/servidor. Sin embargo, TOP END simplifica el proceso de transacciones dividiendolas en dos entidades separadas: procesos de petición de servicios y procesos proveedores de servicios.

Para manejar el desempeño de grandes sistemas y aislamiento de fallas, TOP END incrementa la arquitectura cliente/servidor con un diseño modular del sistema.

Los módulos de TOP END son usados para facilitar la implementación de procesamiento distribuido de las transacciones. TOP END despedaza el proceso de transacciones en grandes componentes que pueden ser distribuidos a través de la red para mayor flexibilidad y eficiencia. Para poder soportar esta distribución, TOP END proporciona un sistema sofisticado de administración el cual incluye autorización de seguridad y autenticación a través de la red. Además como en tuxedo TOP END asegura una integridad global de la transacción a través del protocolo "two-phase commit". TOP END también incluye un administrador de transacciones, instancias de aplicaciones en servidor, un administrador de recursos, un administrador de recursos de comunicaciones y una interfaz de red.

TOP END también proporciona una funcionalidad sofisticada de administración la cual permite que usuarios autorizados administren cambios en la red desde cualquier punto. Estas funciones incluyen levantamiento y baja de TOP END, cuantos clientes pueden estar en la red, distribución de la carga, número de transacciones, rutas de transacciones alternas en caso de caídas de nodos.

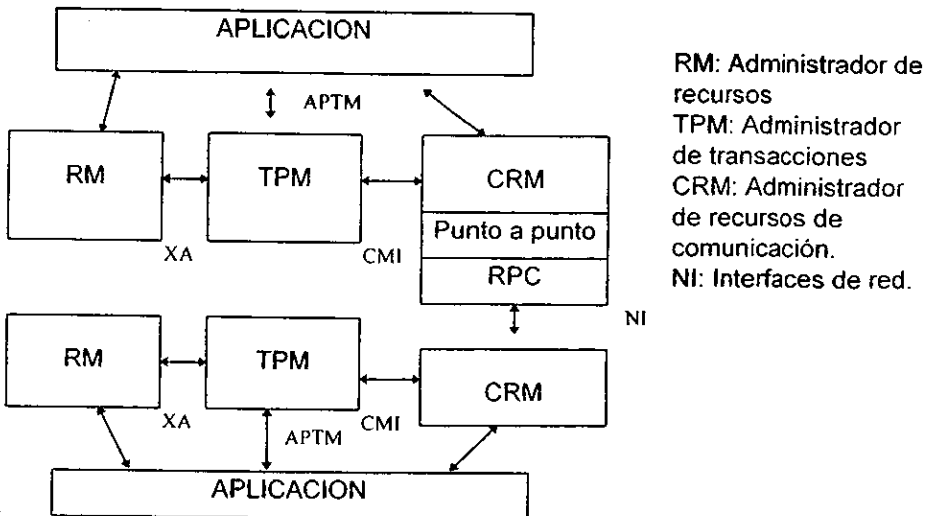


FIG. 1.4.8.B COMPONENTES DE TOP END.

CICS

CICS (*Customer Information Control System - Sistema de Control de Información del Cliente*) es el monitor propietario de IBM. Originalmente nació como un producto del sistema 370. En un ambiente heterogéneo CICS puede comunicarse con otros sistemas a través de sus productos en otras plataformas (CICS/ESA, CICS OS/2, CICS/MVS, CICS OS/400, CICS/AIX).

CICS tiene dos maneras de comunicar un sistema local con un sistema remoto: una es a través de la comunicación entre sistemas (ISC) y la otra es por la operación multiregional(MRO).

MRO es diseñada para comunicar CICS a CICS habilitando a sistemas que están corriendo en el mismo host en diferentes espacios de direcciones. MRO no soporta comunicación con sistemas que no sean CICS. Facilidades de red de SNA no son ocupadas por MRO, todas las transferencias de datos entre los espacios son manejadas por un programa de comunicación intraregional o por el sistema operativo MVS a través de memoria.

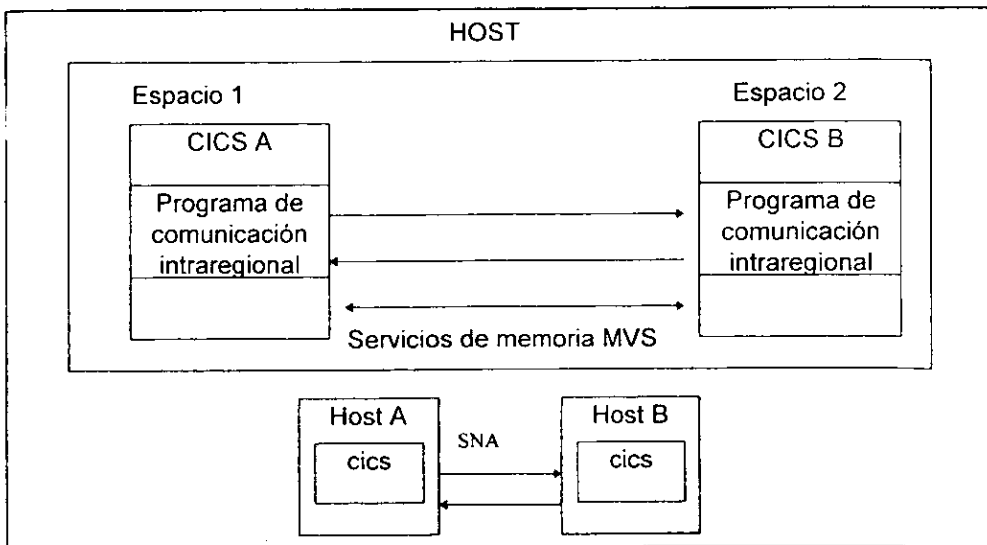


FIG. 1.4.8 C OPERACIÓN MULTIREGIONAL Y OPERACIÓN ENTRE SISTEMAS

ISC se realiza por la comunicación entre sistemas que se encuentran en diferentes localidades. Aunque también puede realizar la comunicación entre sistemas que se encuentran en el mismo Host a través del protocolo ACF/TAM. Para sistemas remotos utiliza las facilidades de SNA para poder realizar la comunicación.

ISC proporciona cuatro tipos básicos de facilidades de intercomunicación: Función de embarque , proceso asincrono, ruteo de transacciones y proceso distribuido de transacciones (DTP):

- Función de embarque : esta facilidad está disponible solo en comunicación CICS a CICS vía MRO. Esta función fue diseñada para habilitar a un programa de aplicación que corre en un sistema CICS y esté pueda acceder un recurso propiedad de otro sistema CICS. El recurso remoto puede ser un archivo o una base de datos, una cola de datos, una cola de datos temporales. Los programas de aplicación pueden ser diseñados y codificados como si estos recursos remotos fueran locales al sistema. CICS utiliza esta función para pasar peticiones de un sistema local a una transacción especial en el sistema remoto.
- Proceso asincrono : esta facilidad habilita a una transacción para distribuir el proceso requerido entre sistemas en un medio ambiente ISC iniciando una transacción en el sistema remoto y pasándole datos a él. La respuesta no tiene que ser regresada a la tarea que inicio la transacción.
- Ruteo de transacciones : Esta facilidad está disponible solo en comunicación CICS a CICS via MRO. Habilita a una terminal que es propiedad de un sistema CICS a iniciar una transacción que es propiedad de otro sistema CICS .
- Proceso de transacciones distribuidas: Esta es la facilidad que actualmente implementa la distribución SAA. Habilita a que una transacción se comunique con

otra transacción corriendo en otro sistema con el propósito de distribuir el proceso requerido entre dos o mas sistemas en un ambiente ISC.

TransaRC

Es un monitor desarrollado por TransaRC Corporation este monitor nació de un proyecto de investigación. TransaRC incorpora tecnología de las organizaciones X/Open y OSF (*Open System Foundation* - Fundación de Sistemas Abiertos), por lo que resulta en soporte de varios proveedores de hardware y software (IBM, Hewlett-Packard, Stratus, Sybase, Informix). De hecho IBM intento utilizar TransaRC como un componente de su producto CICS/Aix. Otras características básicas de TransaRC son :

- Utiliza el paradigma orientado a objetos para encapsular tanto código como datos en el proceso servidor. Además múltiples servidores o múltiples instancias del mismo servidor pueden operar sobre cada uno de los nodos distribuidos.
- Soporta el protocolo commit dos fases sobre múltiples procesadores y nodos de red. Un protocolo commit dos fases local optimizado es usado para transacciones no distribuidas.
- Este monitor es soportado en ambientes de espejo para incrementar su confiabilidad en sistemas distribuidos.

Los componentes de Transarc pueden ser usados para construir administradores de transacciones distribuidas, administradores de recursos, o partes integrales de otros sistemas. Como Transarc utiliza los estándares X/Open sus interfaces aseguran que sus módulos interoperen con otros sistemas heterogéneos a través de la red.

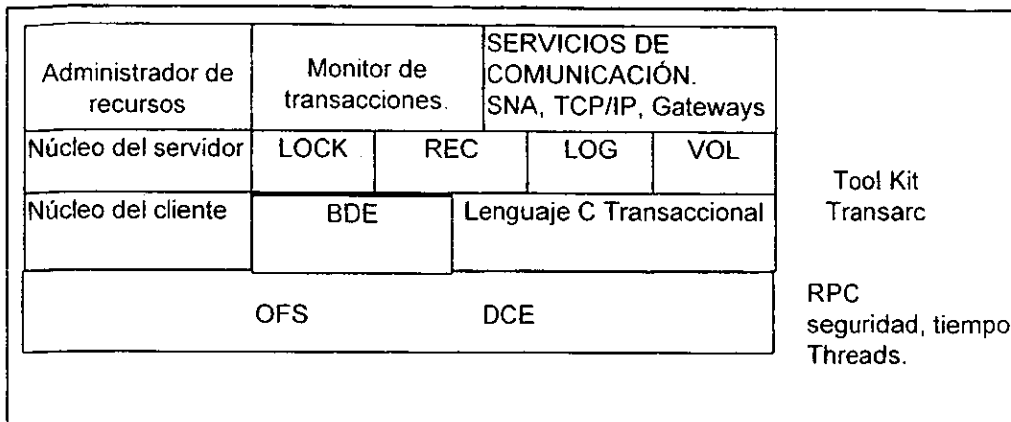


FIG. 1.4.8.D COMPONENTES DE TRASARC

Los componentes de Transarc son diseñados para trabajar junto con la arquitectura DCE de OSF. Transarc proporciona interfaz DCE con su AFS (sistema de archivos distribuidos).

El núcleo del cliente ejecuta el inicio y fin de una transacción, maneja condiciones de excepción y procedimientos remotos transaccionales. El núcleo del cliente incluye:

- **Ambiente base de desarrollo (BDE):** este permite la portabilidad del sistema, soporta sincronización de procesos, manejo de señales y servicios de alarmas y tiempo.
- **Lenguaje transaccional C :** Son librerías en lenguaje C que soportan aplicaciones transaccionales, threads de control, candados, variables recuperables y operaciones de "prepare/commit/abort".

El núcleo del servidor habilita la construcción de servidores recuperables, proporciona una integridad completa de transacciones. Este núcleo incluye lo siguiente:

- **Servicios de bitácora (LOG)** para un almacenamiento de datos confiable.

- Servicios de recuperación (REC) que proporciona la lógica de deshacer/rehacer, administración de buffers, bitácora física/lógica de cambios en datos.
- Servicios de candados (LOCK) para poder tener transacciones compartidas o exclusivas.
- Servicios de discos (VOL) que proporciona administración de almacenamiento en grandes discos a través de múltiples volúmenes, con reducida fragmentación del disco, soporte de alta concurrencias de acceso y espejo de discos.

1.5. HERRAMIENTAS DE DESARROLLO

1.5.1. HERRAMIENTAS DE APOYO PARA EL DESARROLLO DE SISTEMAS.

Introducción.

El front-end de un sistema es un programa de computadora que oculta detalles del acceso a los datos de otra computadora. En un sentido, todo programa es un front-end que evita que los usuarios conozcan los detalles reales del intrincado funcionamiento de la computadora. La mayor parte del tiempo, un front-end significa el uso de la computadora en mayor medida que muchos de los programas. Un ejemplo de un procesador de este tipo, es el que se utiliza como enlace de comunicaciones para un mainframe. En otras palabras, un front-end es un procesador de formatos, es decir, un procesador de entrada/salida utilizado para realzar el formato de los datos de entrada y/o procesarlos. A veces, se utiliza el término para referirse a un programa de comunicación.

El éxito de Windows en el entorno PC ha obligado a los programadores a reciclarse para poder desarrollar aplicaciones que respeten el *look* (presentación) y las necesidades de esta interface gráfica. La llamada programación visual supone un paso más en la búsqueda de nuevas herramientas que faciliten el desarrollo de software para Windows.

Empresas como Microsoft y Borland se han apresurado a lanzar al mercado versiones de sus productos con la palabra "visual", pero aún no está muy claro el significado de este término. Se podría decir que programación visual se entiende como el proceso de desarrollo de forma interactiva del entorno del usuario para sus aplicaciones. La idea es que la programación visual permite que la mayor parte del proceso necesario para crear cualquier aplicación se pueda realizar en forma automática.

Así por ejemplo, si la mayoría de las aplicaciones de Windows disponen de menús desplegables, parece razonable desarrollar una herramienta para crearlos sin tener que recurrir a páginas de código indescifrable. Pero los beneficios no acaban aquí, porque todo lo mencionado hace tiempo que estaba disponible en los editores de recursos, ahora se cuenta con mayor integración entre el desarrollo de aplicaciones propiamente dicha y el diseño de interface de usuario.

Los editores de recursos que actualmente podemos encontrar en el mercado (en los compiladores de pascal o C para Windows siempre viene alguno) siguen estas pautas:

Primero se crea el recurso deseado, que puede ser una ventana, un menú, un ícono, algún tipo de letra, un botón, un campo de texto o cualquier otro elemento típico de Windows, después se genera el código correspondiente al recurso y, por último, con más o menos pasos intermedios, se acaba uniéndolo al resto del código fuente de nuestro programa.

Con la programación visual se pretende simplificar al máximo todo el proceso, evitando cuantos pasos intermedios sea posible. Lo normal es que se parta de la creación de una interface de usuario y se llegue a la escritura del código (si es necesario) en algún lenguaje de programación.

Por el momento, la programación visual ha dado un paso gigante en la tarea de facilitar la vida a los informáticos; sin embargo, todavía no ha llegado el día en que cualquier persona sea capaz de decirle a la computadora mediante un micrófono lo que quiere y que esta lo haga. Hasta entonces tendremos que seguir trabajando con las herramientas disponibles, muchas de las cuales ya han comenzado su transición a la programación visual.

Programación de la Interface.

Con estas premisas, está claro que la programación visual ha pasado de ser una curiosa novedad metodológica, al constituirse como un conjunto de herramientas imprescindibles para el desarrollo de las complejas aplicaciones basadas en GUIs (**Graphical User Interface: Interface gráfica de usuario**) que se comercializan actualmente.

La complejidad de programar bajo una interface gráfica proviene del propio diseño de los entornos. De hecho, desarrollar el programa mas sencillo para un entorno gráfico puede convertirse en una tarea desquiciante si no se dispone de las herramientas adecuadas.

Para afrontar el desarrollo de una aplicación para una interface gráfica (Windows u OS/2 en el mundo PC) podemos seguir dos caminos. El primero consiste en armarnos de paciencia y enfrentarnos con los conceptos básicos del entorno, lo cual resulta una tarea tediosa y poco gratificante porque a veces no es efectiva ante las funcionalidades ya existentes (OLE, Object Linking and Embeding) o los nuevos sistemas (por ejemplo, Windows 95). El segundo camino se simplifica al utilizar las mencionadas herramientas de programación visual.

Estos compiladores simplifican el diseño al proporcionar formatos, tipos o patrones que podemos aplicar a nuestra aplicaciones, al mismo tiempo que ofrecen una elevada confiabilidad porque se tratan de patrones probados en multitud de otros desarrollos. Todo ello redundará en una mayor eficiencia a la hora de programar y en un gran ahorro de tiempo de codificación.

Visual Basic.

Visual Basic (VB) para Windows es un sistema de programación del que se analizarán sus características en el punto siguiente. Esta herramienta ha sido ampliamente y está ahora a la par con el lenguaje C únicamente, y en cuanto a productividad, es claramente superior.

Un programador medio necesita meses para tener una velocidad de programación para Windows usando el lenguaje C, pero con VB se pueden desarrollar aplicaciones para Windows con solo algunas horas de familiarizarse con el lenguaje tanto del paquete como de programación Basic. Se podrá estar satisfecho al ver la primera aplicación desarrollada para Windows con un trabajo fácil y tan rápido como es posible.

VB es altamente interactivo, maneja un lenguaje de programación el cual permite aprender de ambas formas: agradable y productivamente.

VB utiliza varias formas para sus presentaciones visuales de sistemas (en terminología de VB, una forma es una ventana asociada a controles, íconos, gráficos y códigos) la aplicación de COLORBAR permite trazado de gráficas. La forma GETFILE permite al usuario seleccionar un archivo desde cualquier directorio en cualquier *drive*.

Mediante VB se puede generar un sistema que soporte la instalación y uso del ratón para facilidad del propio sistema y del usuario; genera cambios de datos dinámicos (DDE) para aplicaciones basadas en ventanas; trabaja con las bases de datos en una interacción; y objetos ligados e inmersos (OLE).

PowerBuilder.

PowerBuilder es la herramienta de desarrollo de PowerSoft (Sybase) y su configuración de orientación a objetos permite crear arquitecturas Cliente/Servidor flexibles a través de un ambiente de red que permite distribuir objetos, los cuales pueden ser compilados para obtener mejor rendimiento. La capacidad total de orientación a objetos incluye encapsulamiento, polimorfismo y herencia y adicionalmente incluye un browser (herramienta para manipulación de objetos) para aumentar la productividad de programadores. PowerBuilder 5.0 incluye una nueva librería "PowerBuilder Foundation Class" (PFC), la cual permite acelerar el ciclo de desarrollo al incluir una serie de objetos reutilizables.

PowerBuilder 5.0 ofrece soporte a OLE (object Linking and embedding) de Microsoft, permitiendo acceso directo (point-and-click) para realizar automatización de servidores OLE. Las mismas aplicaciones PowerBuilder pueden ser liberadas como servidores OLE para obtener objetos de negocios abiertos y reutilizables. Incluye adicionalmente una herramienta denominada ObjectCycle™, la cual permite administrar el control de versiones de la aplicación.

A través de las características de compilación de código, las aplicaciones PowerBuilder tiene un excelente desempeño en la ejecución de funciones, expresiones matemáticas, ciclos (loops, tales como FOR, NEXT) y el procesamiento de arreglos para acceso a bases de datos (array processing). Inigualmente incluye programas nativos (drivers) para acceso a diferentes bases de datos.

Para el acceso a bases de datos, incluye además, el DataWindow que permite observar gráficamente la base de datos a través de la selección de diferentes presentaciones. Esta selección de presentación puede ser salvada como objetos y ser usada en otras aplicaciones.

Las aplicaciones PowerBuilder pueden ejecutarse en plataformas Windows, Macintosh, y Solaris. La herramienta incluye el manejador de bases de datos para escritorio (desktop) Sybase® SQL Anywhere™.

El conjunto de herramientas de PowerBuilder se determina a través de sus diferentes ediciones: PowerBuilder 5.0 Enterprise Edition (empresarial), PowerBuilder 5.0 Professional Edition (profesional) y PowerBuilder 5.0 Desktop Edition (personal).

Las diferencias básicas residen en la posibilidad que ofrece para construir aplicaciones totalmente distribuidas, multiplataforma o accesos a múltiples bases de datos, así como las herramientas de soporte adicionales.

Delphi.

Delphi es la herramienta de desarrollo de Borland, la cual permite crear aplicaciones Cliente/Servidor y que incluye drivers nativos para las bases de datos mas comerciales: Oracle, Sybase, Informix, DB2, InterBase y Microsoft SQL Server, así como drivers para manejo de Access, FoxPro, dBASE, Paradox y ODBC. Adicionalmente, cuenta con conectividad a AS/400 a través de Delphi/400.

Delphi proporciona acceso múltiple a bases de datos a través de su arquitectura DataSet architecture.

Para ayudar en el desarrollo de aplicaciones, Delphi incluye el SQL Explorer el cual permite acceder el esquema de la base de datos para la creación y modificación de tablas, alias, stored procedures (procedimientos almacenados), triggers (disparadores) y cualquier objeto de la base de datos.

Incluye un monitor de SQL para prueba, depuración y afinación de sentencias SQL que permite programar eficientemente y obtener los mejores rendimientos de la aplicación.

Para la creación de sentencias SQL, incluye el Visual Query Builder que automáticamente crea sentencias SQL ANSI SQL-92.

El módulo WEB permite la creación de HTML (Hypertext Markup Language), CGI (Common Gateway Interface), WinCGI, NSAPI (API -Application Program Interface- de programación de Netscape), ISAPI (API de programación de Internet Server de Microsoft) y ActiveX.

Delphi es una herramienta totalmente orientada a objetos y es posible utilizar los más de 130 componentes de Visual Component Library (VCL) de Borland, accedendo al código fuente. VCL encapsula la interface a Windows 95, rutinas para el manejo de datos, menús, etc. Además del uso de estos componentes, permite la creación de nuevos o la adecuación (customize) de los existentes.

La creación de componentes ActiveX puede realizarse a través de la herramienta. Estos componentes pueden reutilizarse en otras herramientas como C++, Java, Visual Basic, PowerBuilder y aplicaciones Office de Microsoft. Estos controles son 100% compilados y no requieren la inclusión de ninguna librería adicional para su distribución.

Ofrece además la posibilidad de mostrar la sintáxis de cualquier sentencia, las propiedades y eventos de un componente y las declaraciones o argumentos de un procedimiento.

Visual C++

La familia de productos de Microsoft Visual C++ consiste de varios productos: sistema de desarrollo profesional Microsoft Visual C++, Learning Edition (anteriormente conocida como Standard Edition). La diferencia fundamental entre una y otra es que la edición profesional ofrece la optimización de código y el manejo de librerías estáticas.

Visual C++ es una herramienta poderosa, totalmente orientada a objetos y sus nuevas versiones permiten acceder bases de datos a través de objetos, adicionalmente, es una excelente forma de creación de componentes ActiveX.

La versión Learning Edition está orientada a programadores iniciales de Visual C++ y que desean hacer uso de todas las bondades de Microsoft Foundation Classes (MFC). Esta edición incluye tutoriales paso-a-paso, wizards (magos), y acceso a todas las librerías MFC para la creación de aplicaciones altamente integradas a Windows.

Existe la edición Microsoft Visual C++ RISC Edition que está diseñada para el desarrollo de aplicaciones para Windows NT en la plataforma DIGITAL Alpha. Este es un producto nativo en Alpha e incluye mejoras substanciales en el manejo de código: 10% más pequeño y más rápido.

1.5.2. CASE

Introducción.

CASE (Computer Aided Software Engineering) es el acrónimo utilizado para designar a un conjunto de programas y filosofía de desarrollo que ayuda a automatizar el ciclo de vida en el desarrollo de sistemas. El mismo término CASE puede presentar diversas variaciones como se ilustra en la fig. 1.

C	Computer		
A	Aided	Assisted	Automated
S	Software	Systems	
E	Engineering		

Fig. 1 Variaciones del acrónimo CASE

CASE es una tecnología formada tanto por las herramientas como por la metodología. Mientras que las herramientas automatizan el software, la metodología define los pasos del proceso, la entrada y salida para cada paso y el control de calidad.

Es útil para programadores, analistas e ingenieros de sistemas, así como para directivos y ejecutivos de negocios de todos los niveles y para negocios de todos los tamaños y estructuras.

Proporciona herramientas para ayudar a los directores corporativos a planear y documentar su trabajo, a los analistas les da apoyo en el análisis, diseño de sistemas y en la documentación de su trabajo, además de que ayuda a hacer menos pesada la documentación haciéndola en paralelo a la construcción.

La filosofía CASE involucra el uso de la computadora como una herramienta de desarrollo para construir modelos que describen el negocio, el ambiente del negocio y

la planeación corporativa, así como para documentar el desarrollo de los sistemas de cómputo desde la planeación hasta la implementación.

Chen dice que la introducción de un CASE integrado está empezando a tener impactos significativos en el negocio y los sistemas de información de las organizaciones. Con un CASE integrado, las organizaciones pueden desarrollar sistemas de mejor calidad y mucho más rápido para soportar los procesos críticos del negocio y asistir en el desarrollo y comercialización de productos intensivos de información y servicios.

Otro punto de vista de la definición de CASE es en base al ciclo de vida del software, el cual, por definición, representa la serie de fases por las que el software pasa en su desarrollo. El tradicional ciclo de vida del software consiste en el análisis, diseño, codificación, pruebas y operación/mantenimiento. Muchas herramientas CASE han sido desarrolladas con el propósito de automatizar una fase concreta en el ciclo de vida de los sistemas. En adición, diferentes metodologías para el desarrollo de sistemas han sido creadas para proporcionar rutas específicas para el desarrollo de software. Estas metodologías están enfocadas a proporcionar una descripción paso a paso para todas las actividades necesarias en el ciclo de vida de los sistemas.

Por otro lado, las tecnologías CASE proporcionan un lenguaje común entre diseñadores y revisores permitiendo un mejor grado de entendimiento desde el diseño y durante todas las etapas.

Por supuesto existen costos y beneficios asociados con la tecnología CASE. Los beneficios incluyen una excelente documentación y posibilidad de rastreo de la fase de diseño, automatización de la ingeniería de sistemas a través del soporte del ciclo de vida del software y mejoras sustanciales en oportunidades de reingeniería. Los costos de la tecnología CASE incluyen inversión en tiempo y dinero para obtener la capacitación y las herramientas necesarias.

Primera Generación.

- El nacimiento público de CASE fue en un artículo titulado "Computer-Aided Software Engineering", presentado el 17 de abril de 1984 por John H. Manley, entonces vicepresidente de Tecnología e Ingeniería en la compañía Nastec.
- Los primeros productos CASE automatizaron, a través de software, el tiempo consumido en las tareas de analistas y programadores, en poco tiempo se automatizaron las tareas de determinar los requerimientos del sistema, el diseño de una solución y la construcción del sistema.
- Las herramientas CASE hicieron más fácil y rápido el uso de convenciones y técnicas de programación estructurada.
- Con CASE el desarrollador podía dibujar, actualizar, editar y almacenar diversos diagramas usados en las etapas iniciales del ciclo de vida del sistema.
- La computadora se convirtió en una herramienta que podía checar diagramas por seguridad.
- Estos productos CASE se conocen como herramientas UPPER-CASE.
- Las herramientas LOWER-CASE automatizan las tareas de las últimas etapas del ciclo de vida del sistema.
- Automatizan las tareas posteriores a la fase del diseño, incluyendo generación de código, prototipo, pruebas, correcciones.

- Hasta este punto las herramientas auxiliaban en la mayoría de las etapas del ciclo de vida del sistema, sin embargo, los datos y la información no necesariamente fluían automáticamente de una fase a otra.

Segunda Generación.

- Esta generación comenzó a aparecer entre 1985 y 1986 con la introducción de productos que soportaban las tareas en diversas etapas del ciclo de vida del sistema.
- Los desarrolladores usaron información de la etapa de recopilación de requerimientos para desarrollar especificaciones que posteriormente usarían en la etapa del diseño.
- Los desarrolladores de CASE utilizaron esta integración para crear productos que soportaran el ciclo completo de vida del sistema.
- Las herramientas proporcionaron la integración de requerimientos, especificaciones, diseño, código, pruebas, implementación y mantenimiento.
- El repositorio de información surgió como una facilidad de almacenamiento central para toda la información relacionada con el desarrollo del sistema.
- Posteriormente se agregaron técnicas de programación y se demostró que el código se podía generar directamente de los diagramas.
- De esta forma los usuarios de CASE podían ir desde los requerimientos del sistema hasta el sistema operacional.

Dentro de la metodología CASE se pueden identificar dos grandes etapas: PRE-CASE y POST-CASE.

Pre-CASE.

Esta etapa involucra el uso de herramientas de planeación para identificar y valorar el alcance de los proyectos. Los productos Pre-CASE incluyen herramientas para la planeación estratégica, tales como las usadas para desarrollar el modelo de datos corporativo, agrupar los datos en los sistemas de información y clasificar estos sistemas de acuerdo a su impacto en las misiones corporativas. Las funciones administrativas soportadas por herramientas Pre-CASE incluyen planeación del proyecto, calendarización y estimación.

Post-CASE

La incorporación de la tecnología de bases de datos proporcionó un lugar donde las herramientas CASE podían almacenar tanto los datos como nueva información desarrollada por herramientas Pre-CASE.

La base de datos de los productos CASE se conoce como repositorio. Algunos repositorios de CASE son RDBMS (Sistemas Manejadores de Bases de Datos Relacionales) que usan lenguaje SQL, así como bases de datos con modelos jerárquicos.

Con la tecnología de bases de datos como parte de CASE, se necesitan herramienta para transferir código existente a ambiente CASE.

El código estructurado se puede almacenar para utilizarlo como módulos de programas futuros, pero el código no estructurado no funciona en ambientes CASE. La solución es la ingeniería en reversa. Este es el proceso Post-CASE para convertir código no estructurado en código estructurado y almacenarlo en el repositorio de CASE, donde estará disponible para mantenimiento, edición o reutilización.

1.5.3. Upper CASE, Lower CASE

Recordemos que una herramienta CASE es cualquier software destinado para automatizar o asistir en la ejecución de una fase particular en el ciclo de vida de sistemas. Ante esta definición tan general, hay un gran número de formas en clasificar un herramienta CASE. Una forma de hacerlo es desde la perspectiva del sistema, esto es, a través de un CASE integrado (I-CASE). El objetivo de un I-CASE (por ejemplo Information Engineering Facility (IEF) de Texas Instruments) es el de combinar diferentes herramientas para todas las partes del ciclo de vida en un solo paquete de software el cual proporciona toda la asistencia necesaria para cubrir el ciclo de vida de sistemas.

Otra clasificación alterna para herramientas CASE incluye *toolkits*, *workbenches*, *frameworks* y *methodology companions*. Los *toolkits* (por ejemplo Analyst/DesignerToolkit de Yourdon Inc.) ofrecen herramientas integradas para automatizar una etapa del ciclo de vida de los sistemas. Los *workbenches* (por ejemplo IEW) ofrecen herramientas integradas para automatizar todo el proceso de desarrollo de software. Los *frameworks* (por ejemplo CASEdge de Hewlett Packard) proporcionan un ambiente para la integración de herramientas CASE de diferentes proveedores.

Finalmente, los *methodology companions* (por ejemplo IEF) son herramientas CASE que soportan y obligan a seguir una metodología estructurada tal como la metodología *Information Engineering*.

Finalmente, se pueden también clasificar las herramientas CASE de acuerdo a la fase en que se enfocan en el ciclo de vida del desarrollo de sistemas. Por ejemplo, herramientas **Upper CASE** (por ejemplo DesignAid de Nastec Corp.) están concentradas en la automatización de el front-end y en el análisis y diseño del ciclo de

vida del sistema. Por otro lado, las herramientas **Lower CASE** (por ejemplo Telon de Pasophic) facilitan la automatización del diseño físico, implementación y mantenimiento del desarrollo de sistemas.

Dentro de las herramientas upper CASE podemos citar a FAES el cual asiste en la ingeniería de software en el proceso de entrevistas y especificación de requerimientos. A FAES se le incorporaron técnicas de inteligencia artificial para proporcionarle una capacidad "inteligente" en la asistencia del desarrollo de sistemas.

Otro ejemplo de Upper CASE es EUOBFE (End User Oriented Business Flow Editor) el cual combina el lenguaje natural con el proceso de negocio, una base de conocimientos para validar el modelo, un esquema de predicados y objetos para el desarrollo de las reglas del negocio, un diseñador de formas para manipular las reglas del negocio así como una interface pen-based que proporciona una manera fácil y natural para usar EUOBFE.

Con respecto a herramientas Lower CASE, Church y Matthews evaluaron un gran número de herramientas lower CASE orientadas a objetos, concentrándose en 14. Su primer criterio de evaluación fue el grado de generación automática de código, pero también consideraron la generación de documentación, facilidad de uso, verificación de consistencia y solo pocos cumplieron con la generación de código y algunos menos proporcionaron un método fácil para la generación automática de documentación.

1.5.4. ANÁLISIS Y SELECCIÓN DEL SOFTWARE PARA GENERAR EL FRONT-END.

Con respecto a la metodología, la historia es muy similar a las herramientas. El mayor factor limitante en el uso es la labor manual involucrada en su uso. Muchas de las fallas en el desarrollo de sistemas es el resultado de una inadecuada planeación, análisis insuficiente de requerimientos y un inapropiado diseño de especificaciones. Algunas herramientas CASE como Excelerator, IEW y IEF tienen cierto rigor y especificaciones que obligan a seguir apropiadamente una metodología. A través de un ambiente riguroso CASE, estos productos ayudan a desarrolladores a mejorar la veracidad y completez de especificaciones y diseño requeridos para el desarrollo de sistemas. Compañías como Computer Task Group, Sony, Rolls-Royce y Nykredit han reportado índices de productividad/calidad altos usando herramientas como IEF.

A pesar de todo el beneficio potencial ofrecido por estrategias CASE, hay un gran número de problemas asociados con la tecnología CASE. Para empezar, si una compañía decide adoptar la tecnología CASE deberá realizar una inversión grande. El precio de las herramientas CASE varía (desde \$8,000 usd. por Excelerator y hasta \$300,000 por TELON) dependiendo de la funcionalidad y características proporcionados. Dependiendo del tamaño de la compañía, la compra de una herramienta sofisticada como TELON o múltiples copias del popular Excelerator pueden representar una importante inversión en capital. Aún si la inversión inicial hecha en herramientas CASE pudiera ser considerada "menor", el costo asociado a la capacitación necesaria para desarrollar un equipo experimentado en el uso y el soporte de esta herramientas puede ser aún mayor.

Y aún mas allá, el periodo de retorno de la inversión en herramientas CASE puede ser demasiado largo (años, incluso). Muchos de los beneficios de las herramientas CASE, en términos de costo, son la facilidad para mantenimiento y modificación de los sistemas, redundando en bajos costos.

Las herramientas CASE, al igual que cualquier otra herramienta, generalmente requiere que el equipo de desarrollo cambie sus hábitos de trabajo a fin de aprovechar todo el potencial ofrecido por la herramienta. Por ejemplo, la actitud del desarrollador en cuanto al énfasis que da al desarrollo de código deberá cambiar poniendo mayor atención al proceso de análisis de tal forma que los esfuerzos en el desarrollo de software son orientados a actividades front-end, esto es, en la planeación y análisis de oportunidades de negocios, mas que en el proceso de codificación.

Pero el problema mayor alrededor de la adopción de la tecnología o metodología CASE es la existencia actual de los sistemas legados o heredados (*legacy systems*). Aunque CASE ofrece una ventaja enorme en el desarrollo de nuevos sistemas, éste debe ser capaz de convivir con los sistemas, metodologías y notaciones actuales. Las herramientas de **Reingeniería e Ingeniería en Reversa** se han enfocado a resolver este dilema en el ambiente CASE. Desafortunadamente es aun muy pronto para determinar si estas herramientas serán capaces de cumplir con los retos presentados de mantener estos legacy systems.

Otro factor que afecta la adopción y selección de herramientas CASE es la falta de estándares. Las herramientas se encuentran en una etapa primitiva de crecimiento técnico y el movimiento para el establecimiento de estándares internacionales se vislumbra lejano. Sin dichos estándares, no existe manera de prevenir que los proveedores de las herramientas CASE formen alianzas a fin de crear herramientas que satisfagan un nicho de mercado específico. Este tipo de soluciones para mercados específicos impiden el progreso de las herramientas CASE en lo que respecta a integración (compartir datos, portabilidad de operación) puesto que para estos productos no se les requiere que operen bajo ciertas reglas establecidas.

CASE, ya sea como herramienta o como metodología, no es la panacea para los problemas en el desarrollo de software. El uso de CASE proporciona una manera de

resolver muchos problemas pero no libra al desarrollador de su responsabilidad de conocer y entender el negocio al detalle. Finalmente, el diseño y el análisis de sistemas son ampliamente dependientes de las habilidades de los diseñadores y desarrolladores involucrados.

Un estudio presentado por Case Associates Inc. muestra que entre el 50 y 70 por ciento del tiempo de los ingenieros en software está dedicado en hacer cambios en sistemas de misión crítica. Entonces, la herramienta que proporcione el mayor potencial en las organizaciones no es la que ofrezca el mejor desarrollo sino la que facilite el mantenimiento y crecimiento de los sistemas.

Para la correcta selección de una herramienta que asista en el desarrollo de sistemas, se pueden considerar los siguientes elementos:

- | | |
|---------------------------|--|
| Mantenimiento de Software | Se refiere a la capacidad que ofrece la herramienta para modificar y ampliar el software en el futuro. |
| Ingeniería en Reversa | La capacidad de extraer conocimiento del código fuente y las bases de datos actuales para construir un puente entre el ambiente actual y el ambiente CASE. |

Plantillas	Existen dos aproximaciones tradicionales en el desarrollo de sistemas de información: hacerlos o comprarlos. Recientemente apareció una tercera alternativa híbrida: el uso de plantillas CASE. Las plantillas son modelos de sistemas de información que se pueden adquirir y ser modificados o adaptados con la ayuda de herramientas CASE. Esta estrategia de desarrollo es caracterizada por el uso de modelos abstractos en todas las etapas en lugar del uso de especificaciones en texto "plano".
Validación	Se refiere a la posibilidad de utilizar herramientas CASE en el proceso de validación del software, reduciendo los pasos manuales en la verificación, validación y pruebas. Existen varias métricas para realizar la validación de sistemas (function points, object points y variaciones de estas métricas, denominadas short-forms). Estas herramientas están orientadas en la automatización del tedioso proceso y muchas veces fuente de errores, donde supuestamente es uno de los pasos finales del control de calidad, de la verificación de la funcionalidad.
Modelado	Esta técnica es utilizada para representar físicamente el modelo resultante de un proceso de diseño. A través de esta técnica se pretende afinar las consideraciones de diseño a través del

impacto del mundo real donde se desempeñará el sistema. Este mundo real bien puede ser un sistema de tiempo real, un modelo de negocios en ambientes financieros, procesos de manufactura, etc. Los objetivos del modelado es determinar si el diseño propuesto satisficera el requerimiento reales: tiempo de respuesta, funcionalidad, flexibilidad, portabilidad, etc.

Generalmente el proceso de modelado parte de la descomposición del sistema en subsistemas menores a fin de que a través de este proceso de granularidad sea más fácil y controlable el proceso de medición del sistema. Así los esfuerzos se centran primero en el proceso de verificación de pequeños módulos y posteriormente en la integración total.

Estándares

Este "grado" de estandarización puede ser medido en la capacidad misma que tiene la herramienta para integrarse o compartir información con otra del mismo género. Así bien, dentro de las herramientas CASE, el estándar mas maduro es el CDIF (Case Data Interchange Format). Similarmente, existen otras definiciones que pretenden definir un estándar para la integración de métodos, técnicas y herramientas para el desarrollo y mantenimiento de sistemas. Dentro del proceso de selección de una herramienta es conveniente observar el grado de estandarización, definida por el grado de

integración con el resto de las herramientas.

Facilidad de Uso

Otro de los factores a considerar en el proceso de selección es la facilidad de uso, configuración y mantenimiento que ofrece la herramienta. Mucho depende del soporte existente en el mercado el cual facilita el uso acompañado de la portabilidad y disponibilidad en diferentes plataformas, ofreciendo al desarrollador interfaces amigables, portables, bajos consumos de recursos, bien documentadas y presencia fuerte en el mercado que garantice la evolución de la herramienta incorporando nuevas técnicas acorde a las tendencias de los sistemas de información.

Costo

Un aspecto importante en la adopción de una tecnología es el factor costo/beneficio, el cual se refiere a evaluar la rapidez del retorno de la inversión hecha con respecto al tiempo. La inversión, como mencionábamos antes, no solo aplica al costo mismo de la herramienta sino también a los costos derivados de la capacitación y soporte para lograr un óptimo dominio de la misma.

Visual Basic.

Introducción.

Una breve definición que muestra las características de la herramienta en palabras de Microsoft: **"Microsoft Visual Basic version 4.0 is the programming system for Windows that grows with your needs and experience. Create everything from simple programs to advanced, enterprise-wide client/server applications, and take advantage of the latest three-tier capabilities"** (Microsoft Visual Basic, versión 4.0, es el sistema de programación para Windows que crece junto con sus necesidades y experiencia. Crea desde programas simples a avanzados, aplicaciones Cliente/Servidor empresariales y toma ventaja la tecnología de tres capas).

La visión de Visual Basic es la de proporcionar a los desarrolladores una herramienta para la creación de sistemas para el sistema operativo Windows en forma rápida y eficiente. Esta herramienta soporta todas las plataformas de Windows -Windows 95 y Windows NT-, cubriendo desde desarrolladores individuales a grupales.

La base de este sistema de programación es OLE, el modelo de objetos de Microsoft. Junto con OLE, los controles VBX constituyen la colección mas grande de objetos de componentes pre-construidos que pueden ser adquiridos, usados y reusados en cualquier programa.

Ediciones de Visual Basic.

Con la edición profesional de Visual Basic (*Professional Edition*) es posible crear aplicaciones Cliente/Servidor robustas que soportan el acceso local o remoto de bases de datos. Para equipos de desarrollo trabajando sobre la misma aplicación existe la edición empresarial (*Enterprise Edition*) la cual permite a los equipos de desarrollo trabajar sobre el mismo código a través del sistema de control Microsoft Visual

SourceSafe. Adicionalmente, esta versión permite el acceso directo, en esquema Cliente/Servidor a servidores de bases de datos remotos a través de el control *Remote Data*. Lo que distingue a la Enterprise Edition de otras herramientas Cliente/Servidor es que, además de ser la primera herramienta dentro de la clase de sistemas de Segunda Generación, ofrece a los desarrolladores un ambiente para la construcción de sistemas escalables, mantenibles y altamente reutilizables dentro del esquema de RAD (Rapid Application Development).

Por otro lado, la edición estándar de Visual Basic (Standard Edition), es una manera fácil y barata para iniciarse a programar y obtener la experiencia en el desarrollo de aplicaciones para Windows.

Existen más de 2 millones de copias de Visual Basic en manos de desarrolladores creando aplicaciones poderosas para millones de personas cada día.

Características:

Visual Basic, Enterprise Edition: La herramienta para grandes equipos desarrollando grandes aplicaciones.

- Se obtienen aplicaciones escalables y fáciles de mantener en forma rápida.
- Permite el desarrollo y depuración de una aplicación totalmente distribuida en una sola estación de trabajo para posteriormente ser liberada en una red de servidores en un ambiente distribuido.
- Permite el particionamiento de los objetos OLE que contienen las reglas de negocios.
- Proporciona una gran capacidad de reutilización de componentes de negocios que son accesados a través de la red desde una gran variedad de herramientas de desarrollo.

- Utiliza el control Remote Data para acceder en forma rápida a bases de datos que soportan ODBC como Microsoft SQL Server y ORACLE.
- Soporta grandes equipos de desarrolladores directamente a través del sistema de control de versiones integrado a Visual Basic: Microsoft Visual SourceSafe™.
- Implementa el marco de soluciones Microsoft (Microsoft Solution Framework) para servicios distribuidos.
- Distribuye aplicaciones en forma efectiva a través de los más populares protocolos de red.
- Diseñado para ser compatible con futuras generaciones de la arquitectura del sistema operativo Windows.

Visual Basic, Professional Edition: *State-of-the-art tool. State-of-the-art solutions.*

- Toma las ventajas que un buen programador necesita para programar rápidamente para Windows 95, Windows NT y Windows 3.1, tales como crear componentes OLE reutilizables y construir aplicaciones Cliente/Servidor.
- Crea aplicaciones Windows para plataformas de 16 y 32 bits y auxilia en la transición de 16 a 32 bits.
- Apoya en la construcción de soluciones ad hoc tales como aplicaciones Cliente/Servidor, Sistemas Ejecutivos de Información, soluciones de workflow y workgroup, utilerías para productividad en red y de oficina, aplicaciones de manufactura, financieras, etc.
- Permite la creación de librerías con objetos OLE que pueden ser accedidos desde cualquier controlador OLE, tal como Excel.
- Permite el desarrollo a través de la base de datos incorporada a la herramienta la cual proporciona acceso a las bases de datos mas populares.
- A través del ambiente abierto de desarrollo permite a los desarrolladores agregar fácilmente nuevas características al mismo ambiente de desarrollo.

incluyendo herramientas CASE, sistemas para el control de versiones o componentes creados por el desarrollador.

- Permite la creación e incorporación de *wizards* a través del *Add-In Manager*.
- Cuenta con una interface directa a las soluciones de grupo más populares como Microsoft Exchange y Lotus Notes.
- Proporciona a los usuarios todo el poder de Windows 95 a través de controles basados en este sistema operativo.
- Desarrolla aplicaciones que se ven y trabajan como las aplicaciones de uso diario, usando la capacidad de integración de OLE.

Visual Basic, Standard Edition: La manera rápida de construir aplicaciones para Windows 95, es Windows 95.

- Permite que las necesidades de sistemas sean rápidamente trasladadas a soluciones reales a través del ambiente fácil-de-usar-y-aprender (*easy-to-learn-and-use*), lenguaje de programación flexible y capacidad de acceso a datos. Todo esto tomando ventaja de las capacidades y características de Windows 95.
- Contiene todo lo necesario para RAD (*rapid application development*) a través del ensamble de componentes preconstruidos OLE.
- Incluye un engine de base de datos y controles para su acceso para la fácil construcción de programas con acceso a datos.
- Desarrolla aplicaciones que se ven y trabajan como las aplicaciones de uso diario, usando la capacidad de integración de OLE.

En resumen, para cualquier tipo de desarrollador existe una herramienta Visual Basic.

La siguiente tabla resume las características principales y en las versiones en las que están disponibles:

Característica	Standard Edition	Professional Edition	Enterprise Edition
Creación de aplicaciones de 32-bits	X	X	X
Ambiente de desarrollo Visual para creación rápida de aplicaciones <i>drag-and-drop</i>	X	X	X
Fácil acceso a ayuda en línea	X	X	X
Migración automática de Visual Basic 3.0 a Visual Basic 4.0 de 32 bits.	X	X	X
Creación de aplicaciones de 16 y 32 bits		X	X
Ambiente abierto de desarrollo integrado con tercero para modelado, control de versiones e integración de <i>wizards</i> .		X	X
Herramienta para la creación de Setup para instalación y desinstalación.		X	X
Soporte de VBX, controles OLE, servidores OLE y DLLs.		X	X
Compilación condicional para soporte de múltiples plataformas usando el mismo código fuente.		X	X
Encapsulamiento de strings, bitmaps e iconos como Windows Resource Files		X	X
Controles adicionales incluyendo interfaces a controles 3-D, controles Multimedia.		X	X
11 controles OLE basados en Windows 95 para aplicaciones en Windows 95		X	X
Replicación programada de la base de datos del <i>engine</i> integrado a la herramienta.		X	X
Acceso a la base de datos integrada a través de programación.		X	X
Total soporte de ODBC incluyendo los drivers para Microsoft SQL Server y ORACLE		X	X
Data Manager para creación y manipulación de bases de datos, tablas e índices.		X	X
Creación de componentes OLE.		X	X
Creación de componentes OLE DLL.		X	X
Soporte automático de versiones de componentes OLE.		X	X
Compilador de Help para creación de archivos tipo Help de windows.		X	X
Reporteador Crystal Reports incluyendo integración mejorada con bases de datos y afinación.		X	X
Creación de aplicaciones distribuidas de tres capas.			X
Tecnología <i>Remote Automation</i> para liberación de aplicaciones distribuidas.			X
Ambiente de desarrollo totalmente distribuido y de depuración en una sola máquina.			X
<i>Component Manager</i> para manejo y liberación de componentes OLE.			X
<i>Pool Manager</i> para optimizar rendimiento en aplicaciones distribuidas.			X
Control optimizado para acceso a SQL Server y ORACLE.			X
Control integrado <i>Visual SourceSafe</i> para soporte de diferencias visuales, en componentes compartidos e historia del proyecto.			X
Objetos de datos remotos para acceso a bases de datos remotas.			X
Herramienta para <i>Profiling</i> de código.			X
Data Explorer para acceso optimizado de datos.			X
Seguridad abierta tipo RPC para el control de acceso a componentes distribuidos.			X
Trabajo distribuido sobre los más populares protocolos de red.			X

Requerimientos.

Enterprise Edition:

- Personal computer (PC) con procesador 486/25 o mayor.
- Microsoft Windows 95, ó
- Windows NT Workstation versión 3.51 o mayor, o
- Windows versión 3.1 (requiere MS-DOS version 5.0 o mayor),
- 8 MB de memoria (recomendado 12 MB o más) si se usa Windows 95 ó Windows 3.1;
- 16 MB si se usa Windows NT Workstation.
- Espacio en disco disponible:
 - Instalación mínima: 20 MB
 - Instalación completa: 80 MB
- Drive de CD-ROM.
- Monitor VGA o mayor.
- Mouse Microsoft o compatible pointing device.

Professional Edition:

- PC con procesador 386DX/25 o mayor (se recomienda 486 o mayor)
- Microsoft Windows 95, ó
- Windows NT Workstation 3.51 o mayor, o
- Windows 3.1 (requiere MS-DOS versión 5.0 o mayor)
- 6 MB de memoria disponible (se recomiendan 8 MB o más) si se usa Windows 95 ó Windows 3.1;
- 16 MB si se usa Windows NT Workstation
- Espacio en disco disponible:
 - Instalación mínima: 20 MB
 - Instalación completa: 70 MB
- Drive de CD-ROM (disponible también en discos de 3.5" alta densidad)
- Monitor VGA o mayor.

- Mouse Microsoft o compatible pointing device.

Standard Edition:

- PC con procesador 386DX/25 o mayor (se recomienda 486 o mayor)
- Microsoft Windows 95, ó
- Windows NT Workstation 3.51 o mayor, o
- Windows 3.1 (require MS-DOS versión 5.0 o mayor)
- 6 MB de memoria disponible (se recomiendan 8 MB o más) si se usa Windows 95 ó Windows 3.1;
- 16 MB si se usa Windows NT Workstation
- Espacio en disco disponible:
 - Instalación mínima: 20 MB
 - Instalación completa: 60 MB
- Drive de CD-ROM (disponible también en discos de 3.5" alta densidad)
- Monitor VGA o mayor.
- Mouse Microsoft o compatible pointing device.

Notese que algunas características de la Enterprise y Professional edition, incluyendo el control Remote Data, el componente OLE para la creación de DLL, controles Windows 95, y componentes para la creación de aplicaciones distribuidas, están sólo disponibles en las versiones de 32 bits y requieren Windows 95 o Windows NT Workstation 3.51.

Un análisis más detallado de las herramientas se realizará en el punto 2.4 y 2.5 a fin de observar detalladamente tanto el front-end, la base de datos y empatarlos con los requerimientos específicos de la empresa.

1.6 CONTABILIDAD.

1.6.1. CONCEPTOS BÁSICOS CONTABLES.

1.6.1.1 Introducción.

La sociedad de nuestros días es una expresión de toda una corriente de comunidades con características particulares, siendo algunas de ellas tan específicas que han constituido etapas fundamentales en el curso de la historia.

Todo ese devenir histórico, hasta hoy en día, así como experimenta condiciones de diferencias, también presenta características y necesidades comunes, como lo es, la necesidad de información financiera y la participación imprescindible de la contabilidad como medio para satisfacer directamente tales necesidades.

Se ha confirmado que cada sociedad ha tenido, tiene y tendrá necesidad de contar con información financiera. Pero la sociedad en sí, como un todo, se encuentra constituida por instituciones, gobiernos, empresas, organizaciones profesionales y demás que a mayores recursos materiales y humanos que poseen para la consecución de fines determinados, van requiriendo de amplia información financiera para tomar decisiones que les permitan alcanzar sus metas. De esta manera todo conjunto de recursos materiales y humanos para la obtención de fines determinados se denominará **ENTIDAD**.

Considerando de esta manera que los recursos son entonces los bienes de que disponen las entidades para lograr sus fines.

Entendido el concepto de entidad como el conjunto de recursos materiales y humanos para la obtención de fines determinados, se hace lógico el concebir la existencia de entidades en todo el curso de la historia.

Es indiscutible que todas ellas experimentaron y viven, las de nuestros días, un sinnúmero de necesidades de todos tipos como lo son entre ellas las económicas.

Tiempo atrás, por las características propias de la sociedad , la necesidad de controlar los recursos de las entidades no tenía las atenciones que hoy se exige. En la actualidad, al vivir en una comunidad con características múltiples y complejas impulsa al ser humano a ser más analítico e investigar con amplitud para tomar decisiones que con mayor o menor repercusión afectará el curso de sus actividades. Por las mismas razones, los recursos de las entidades de nuestros días guardan una importancia de tal magnitud que crea en ellas la necesidad imprescindible de contar con información que exprese la forma en que se ha obtenido y a su vez, aplicado, los recursos, proceso al cual se conoce con el nombre de **finanzas**.

Parece ser, que los matices de la sociedad actual dejan vislumbrar un futuro en el cual la necesidad de controlar los recursos y obtener información financiera será de exigencia mayor.

El avance tecnológico no es que refleje un estancamiento, ni mucho menos, es en si la evolución de la sociedad quien brinda retos a cada instante para medir la capacidad creativa del mismo ser humano quien resulta ser el creador de necesidades y satisfactor directo de las mismas.

El ser humano en todas sus actividades desarrolla medidas de control. Si sucede así en esta entidad, es por ende palpable el observar que las entidades en general implantan sistemas de control de recursos.

Así observamos que el individuo como individuo establece tiempos para alimentación, trabajo y descanso como medida de control para guardar un ritmo de vida saludable. También es común entre nosotros como individuos, el controlar nuestro sueldo para

llevar a cabo alguna compra futura o bien distribuirlo entre los gastos personales y pago de préstamos.

Existen entidades comerciales, industriales, agrícolas, ganaderas, pesqueras, etcétera, que implantan también ciertos sistemas para controlar sus recursos y como resultado de ellos expresan tener utilidades en tanto que pagan sus deudas y conservan un remanente.

Esta información empírica, fruto de un control informal, proporciona información poco confiable para descansar sobre ella, la responsabilidad de las decisiones a las que se puede llegar.

Hoy en día los problemas son múltiples y las decisiones deben tomarse con mayor rapidez y superior índice de eficiencia. Por tanto se requiere de una adecuada información financiera.

La contabilidad, como polo opuesto a la información empírica, es el medio que, por sus métodos y técnicas, permite controlar y visualizar, a través de estados financieros, información clara, veraz y oportuna de todos los recursos de la entidad y solo así poder tomar decisiones, conscientes de sus consecuencias.

1.6.1.2 Antecedentes históricos.

Desde siempre el individuo se ha esforzado por tener información. Los procedimientos para ello, han evolucionado en función de las características y necesidades del medio.

Las primeras etapas de la contabilidad como medio para controlar y proporcionar información financiera, se aprecia en el siglo XV, cuando en Italia, el monje fray Luca Paciolo crea libros para registrar la obtención y aplicación de recursos consecuentes a las operaciones realizadas por las entidades, definiendo a su vez reglas para su manejo. Así, estos libros satisfacen las necesidades de un período de la historia.

Una expresión más del proceso evolutivo de la contabilidad, se tiene a fines del siglo XVIII, cuando Edmond Le Granje, en Francia, implanta el libro mayor tabular, cuya importancia es superior, por contar con las características básicas de los registros tabulares posteriores.

En la actualidad, la tecnología está a las ordenes de la contabilidad para implantar, reajustar o sofisticar sistemas completos, que proporcionen información financiera con la claridad, veracidad y oportunidad deseada, lo cual permiten descansar en ella, la confianza para tomar decisiones de diferente indole.

1.6.1.3 Objetivo de la contabilidad.

El objetivo de la contabilidad se puede resumir, y entendiendo por entidad, el conjunto de recursos materiales y humanos para la obtención de estudios determinados, que la contabilidad es el medio que con métodos y técnicas controla los recursos de las entidades y, en consecuencia, satisfacer su cada vez más creciente necesidad de contar con información financiera.

1.6.1.4 Los estados financieros.

La contabilidad por una parte controla la multiplicidad de recursos de las entidades. Pero, como dicho control no tiene sentido por si solo, la contabilidad proporciona información resultante de él, a través de documentos denominados técnicamente **ESTADOS FINANCIEROS.**

Existen diversos estados financieros, en función al tipo de información que proporcionan, sobre los cuales comentaremos más adelante.

1.6.1.4.1 Objetivo.

Todas las entidades por una u otra situación ven modificados sus recursos. Si compran mercancías pagándolas con efectivo, aumentan ciertos recursos, que son las propias mercancías, pero a su vez disminuye otro recurso, el efectivo. Estas situaciones que afectan los recursos se denominan **operaciones**.

1.6.1.4.2 Elementos esenciales del balance general.

Todas las entidades, cualquiera que sea su actividad o giro: hospitales, centros deportivos, fábricas, bancos, sindicatos, profesionales, tiendas comerciales, etc., cuentan con recursos y efectúan operaciones que en una o en otra forma los afectan.

Estos recursos como: edificio, máquinas de escribir, camionetas para repartos de mercancías, aparadores, mercancías, dinero en efectivo, etc. y todos los demás propios de cada entidad, tienen diversos orígenes, esto es, están a disposición de las entidades pero no necesariamente se encuentran pagados en sus totalidad, quedando, en algunos casos, una cantidad pendiente de pagar, esto es, una **deuda u obligación**.

Las entidades realizan así, una intensa variedad de operaciones pero todas ellas relacionadas directa o indirectamente con los recursos.

Por lo tanto, si la contabilidad es el medio para satisfacer necesidades de control e información, presentada esta, a través de estados financieros, servirán para controlar las operaciones realizadas por las entidades e informar a través de un estado financiero denominado **balance general**. Los recursos con los que cuenta, obligaciones contraídas y la parte de los recursos que han sido adquiridos, valga la redundancia, con recursos propios. Toda esta información referida a una fecha determinada.

Por tanto, el balance general no es un estado financiero exclusivo de entidades de gran magnitud como pudieran ser: Petroleos Mexicanos, BANAMEX, la UNAM, etc., sino de toda la entidad, sin importar su magnitud, actividad, o régimen legal.

De esta manera, de cualquier entidad considerando a esta como el conjunto de recursos materiales y humanos para la consecución de fines determinados se podrá formular un balance general.

El balance general es un estado financiero que informa de los recursos, obligaciones y recursos propios o patrimonio correspondiente a una entidad y a una fecha determinada.

Con esto se llega así a obtener una definición elemental del balance general que se formalizará en el siguiente punto.

1.6.1.4.3 Denominación técnica de los elementos anteriores.

En efecto, los elementos del balance general son los recursos, obligaciones y patrimonio pero así como todas las ramas del conocimiento humano poseen una terminología propia, la contaduría pública también dispone de ella, por tanto, estos elementos del Balance General técnicamente se denominan:

Recursos	=	Activo
Obligaciones	=	Pasivo
Patrimonio	=	Capital

1.6.1.4.4 Formas de presentación del Balance General.

El balance general es un estado financiero cuyo objetivo es satisfacer necesidades de información de las entidades. Por lo anterior, el balance general puede presentarse en la forma más conveniente, observando en todos los casos, las siguientes consideraciones:

- **Primera.**

La ecuación del balance general siempre debe cumplirse:

$$\text{Activo} = \text{Pasivo} + \text{Capital}$$

$$\text{Pasivo} = \text{Activo} - \text{Capital}$$

$$\text{Capital} = \text{Activo} - \text{Pasivo}$$

- **Segunda.**

La información proyectada debe cumplir el principio de contabilidad generalmente aceptado denominado, principios de revelación suficiente, lo cual es factible, elaborándose el balance general en forma clara, veraz y oportuna. Por tanto, se debe formular este estado financiero en la forma y con la terminología más comprensible para los interesados, con cifras reales y con una presentación que no sea motivo para dejar de proporcionar el balance general en la fecha solicitada.

Por lo precedente, no existe alguna forma especial o única para presentar el Balance General, pero la predominante en nuestro medio es la forma de cuenta.

$$\begin{array}{cc} \text{A} & \text{P} \\ & + \\ & \text{C} \end{array}$$

En segundo término, la forma de reporte:

$$\begin{array}{r} A \\ - P \\ \hline C \end{array}$$

Para llegar a un completo balance general es necesario clasificar los conceptos del activo, pasivo y capital. No interesa si se trata de una entidad comercial o de beneficencia; si es comerciante o agricultor; comerciante de alimentos o prendas de vestir; el pequeño comerciante o la entidad monopolista. Al elaborar el balance general de cualquier entidad los conceptos del activo deben agruparse:

- a) *Activo circulante.*
- b) *Activo no circulante.*

Para esta clasificación se tomará en consideración la disponibilidad de los recursos, entendiéndose por disponibilidad la rapidez en que pueden convertir los recursos en efectivo.

1.6.1.5 Activo circulante.

El activo de cualquier entidad se presenta por medio de conceptos que deben ser accesibles para comprender el recurso o activo que está representado. Ejemplos: bancos, denota el dinero depositado en instituciones bancarias; identificados con conceptos representativos de los diferentes recursos o activos, como se tiene en el balance general, el cual tomaremos como modelo para la ejemplificación de este y próximos puntos, se clasificarán bajo el rubro de **CIRCULANTE**: aquellos recursos o activos que, aparte del efectivo, sean disponibles en menos de un año, esto es, que sean factibles de convertirse en efectivo en el lapso indicado, por tratarse de recursos que normalmente se cobran, se venden o se consumen en tal periodo.

1.6.1.6 Activo no circulante

Si para clasificar el activo se ha tomado en consideración el grado de disponibilidad de los recursos, en el grupo de los activos no circulantes se incorporarán todos los demás recursos no clasificados dentro del activo circulante. En consecuencia, al activo no circulante, corresponden aquellos recursos convertibles en efectivo a más de un año, o que normalmente se disponga de ellos para auxiliar en la ejecución de los objetivos o actividades propias de la entidad, independientemente de ser posible o no su conversión a efectivo en menos del lapso citado.

1.6.1.7 Inversiones permanentes

Cuando la entidad invierte su efectivo en valores que le producirán un interés, un dividendo o cualquier otro producto se controlarán dentro del activo no circulante con el rubro de investigaciones permanentes. La condición para controlarse a este grupo de activo consiste en que los valores como bonos financieros, obligaciones, certificados de depósito, etc, sean convertibles en efectivo sólo hasta después de un año de la fecha del balance general que los muestre. Estos valores son vendidos al público por instituciones financieras, quienes normalmente pagan los intereses ganados en forma mensual o trimestral, pero el dinero invertido en tales valores sólo es reintegrado hasta después de un año o más de la fecha de adquisición.

Como segunda condición se tiene, que la inversión sea realizada con intención de conservarla por un periodo mayor de un año. Como en este caso, puede citarse la compra de acciones de alguna entidad, con la finalidad de mantener con tiempo indefinido esta inversión, independientemente de tratarse de un documento que en principio puede ser convertible en dinero, en menos de un año.

1.6.1.8 Inmuebles y equipo

Este concepto comprende los inmuebles y equipos que lógicamente son factibles de convertirse en efectivo en un periodo menor a un año, pero ante la mentalidad de ser adquiridos para conservarse por tiempo indefinido, por estar destinados a cooperar en la realización de los objetivos de la entidad, quedan clasificados dentro del activo no circulante.

Las entidades de nuestros días son múltiples y diferentes contando por consecuencia con diferentes inmuebles y equipos como:

Terreno, edificio, "construcciones"; equipos de oficina, escritorios, calculadoras, archiveros, libreros, etc; mobiliario, mostradores, estantes, etc; equipo de transporte, automóviles, camiones para transporte de personal, camionetas para reparto, motocicletas, etc.

1.6.1.9 Intangibles

Existen otros activos que como en el caso de ciertas inversiones permanentes e inmuebles y equipo, a pesar de ser posible su disponibilidad para convertirse en efectivo, en un periodo menor a un año a partir de la fecha del balance general, se clasifican en el activo no circulante, por existir normalmente la intención de conservarlos por largo tiempo. (Fig. 1.6.1.A. Ejemplos de Intangibles).

Patentes	Empleado en entidades con actividades de fabricación.
Marcas	La cantidad pagada para tener derecho a explotar el nombre de algún artículo, aplicándose a otro similar o diferente.
Crédito Mercantil	Cantidad pagada en la compra de una entidad por el supuesto prestigio o fama que pudiese tener. Este prestigio o fama permite al nuevo propietario conservar a la clientela existente, y evitar la difícil situación de iniciar toda una labor de publicidad para dar a conocer al público la existencia de una nueva entidad así como los artículos que ofrece en venta.

Fig. 1.6.1.A. Ejemplos de Intangibles.

1.6.1.10 Cargos diferidos

Los activos diferidos conocidos como cargos diferidos son aquellos recursos por aprovecharse durante varios años.

Se constituyen dentro del activo no circulante en virtud de tener una difícil disponibilidad, esto es, su convertibilidad en efectivo es mínima. (Fig. 1.6.1.B. Ejemplos de Cargos Diferidos).

Gastos de Organización	Importe de los pagos realizados para la organización inicial de la entidad. Es el caso de los honorarios pagados al licenciado en contaduría por sus servicios en el establecimiento del sistema implantado para controlar el activo, pasivo y capital y permitir la elaboración de estados financieros.
Gastos de Instalación	Valor de los pagos efectuados con la finalidad de adoptar el local o construcción a las necesidades de la entidad, para su adecuado funcionamiento. Como ejemplo puede citarse: pago por instalaciones eléctricas y sanitarias, reparación de pisos, trabajos de pintura, etcétera.

Fig. 1.6.1.B. Ejemplos de Cargos Diferidos.

1.6.1.11 Clasificación del pasivo

Con la finalidad de presentar en forma más técnica el balance general, se ha procedido a clasificar al activo con base a su grado de disponibilidad en dos grupos: activo circulante y activo no circulante.

Considerandose esta disponibilidad como la conversión del activo en efectivo, en un plazo no mayor a un año contando a partir de la fecha del balance general en el cual se presenten.

De esta manera, al ser el activo motivo de clasificación, el pasivo no puede ser excepción y, para alcanzar una elaboración más técnica del balance general, el pasivo se divide en dos grupos en función a su grado de exigibilidad, esto es, al tiempo en que deberán ser pagadas las obligaciones. Para los efectos de la clasificación del pasivo como medida de exigibilidad se tomará un año, pudiendo variar. Por lo anterior, el grado de exigibilidad permite clasificar al pasivo en dos grupos:

- a) A corto plazo.
- b) A largo plazo.

- **A corto plazo**

Al formularse un balance general en el grupo del pasivo a corto plazo se incluirán todos aquellos conceptos con la obligación de ser pagados dentro de un año, contando este tiempo a partir de la fecha a la cual se refiera el balance general por elaborarse. Ante esta observación si deseo elaborar un balance general al 30 de agosto de 1997 en función a los conceptos del estado financiero, tengo que identificar cuales de los conceptos expresados en el pasivo debo pagar dentro del año siguiente a la fecha del balance, esto es, las obligaciones por liquidar el 30 de agosto de 1997 al 30 de agosto de 1998.

- **A largo plazo**

Si el pasivo a corto plazo comprende obligaciones por pagarse a menos de un año, el pasivo a largo plazo se integrará por todas aquellas obligaciones con el compromiso de liquidarse en más de un año, tomando siempre como base la fecha del balance general.

1.6.1.12 Clasificación del capital

Analizado el pasivo y activo, solo resta para concluir el clasificar el capital.

El capital en el balance general de cualquier entidad es igual a la diferencia de activo menos pasivo como ha sido explicado al comentar la igualdad o la ecuación del balance general. El capital al presentar la parte del activo que ha sido adquirida o aportada con recursos propios de la entidad se clasifica en dos grupos:

- a) Capital inicial.
- b) Utilidades o pérdidas acumuladas.

- **Capital inicial**

Normalmente toda entidad para iniciar sus operaciones solo cuenta con efectivo propio. También se dan casos en que, aparte de contar con efectivo se aportan otros activos en especies como: mercancías, escritorios, máquinas de escribir, etc, pero también propios. Sea una u otra forma si se desea elaborar un balance general de la entidad en el primer momento de iniciar sus actividades, su activo estará representado por una cantidad equivalente al efectivo y en su caso, al valor de los recursos aportados en

especie. No habrá pasivo al no existir obligaciones y el capital será igual al activo, en tanto que los activos han sido adquiridos en su totalidad con recursos propios.

- **Utilidades o pérdidas acumuladas**

Todas las entidades comerciales tienen como objetivo principal el lucro esto es, obtener utilidades a través de la compra y venta de artículos.

Es posible asegurar que todos hemos efectuado por lo menos en una ocasión una operación de vender a una persona determinado artículo para recibir dinero en una cantidad superior al valor de dicho objeto y de lograrse, es cuando expresamos ¡gané!

Así como estas operaciones son realizadas por las entidades comerciales, con la diferencia que nosotros las hemos llevado a cabo en forma esporádica, en tanto que dichas entidades, las realizan constantemente y en forma natural con la total intención de tener utilidades y, naturalmente, evitar pérdidas.

Finalmente, todo tipo de utilidades se presentan en el balance general dentro del capital, aumentándolo; en tanto de existir pérdidas, también se presentan en el capital, pero disminuyéndolo.

1.6.1.13 Interpretación del balance general.

En puntos anteriores al clasificar el activo, pasivo y capital se ha obtenido un balance general técnicamente elaborado. Las observaciones señaladas para tal clasificación son procedentes para cualquier balance general que se desee formular.

El activo, pasivo y capital presentado en el balance general, constituye la situación financiera de la entidad, motivo por el cual este estado financiero es denominado estado de situación financiera.

- **Principio de la entidad**

La información financiera es canalizada en obediencia a los principios de contabilidad generalmente aceptados. De esta forma es útil para ser interpretada y posteriormente tomar decisiones.

Por tanto, el principio de la entidad es aquel que permite delimitar e identificar a la entidad, en virtud de concluir su balance general en su propia situación financiera, sin presentar por ende, conceptos de activo, pasivo y capital no correspondiente a ella.

1.6.1.14 Elementos esenciales del Estado de Resultados.

Toda entidad mercantil como las cadenas de tiendas comerciales, zapaterías, tiendas de abarrotes, mueblerías, almacenes de prendas de vestir, tlapalerías, gasolineras, librerías, farmacias, distribuidoras de automóviles, etcétera, tienen como objetivo fundamental el obtener utilidades por medio de la compra de artículos a los fabricantes y su venta al público.

El estado de resultados es un estado financiero que informa de los conceptos que aumentan y disminuyen al capital, para determinar finalmente la pérdida o utilidad resultante de las operaciones realizadas durante un periodo determinado.

Así como los recursos, obligaciones y patrimonio son denominados técnicamente activo, pasivo y capital; en el estado de resultados, los conceptos o elementos que representan utilidades y por tanto aumentos de capital, son denominados técnicamente *ingresos* y los elementos que disminuyen el capital por representar pérdidas se denominan *egresos*.

La diversidad de actividades dentro de las entidades mercantiles, dan origen a ingresos y egresos, siendo éstos, en algunas ocasiones, múltiples, lo cual daría lugar a un Estado de Resultados muy extenso, al tener la necesidad de detallar todos los conceptos de ingreso; así como los correspondientes a egresos.

El optar por elaborar un estado de resultados largo, si bien es cierto que muestra un análisis de ingresos y egresos, también invita a distraer la atención de quienes van a tomar decisiones. Normalmente los estados financieros, por ende, tienden a ser concretos para informar de manera resumida, con posibilidad de recurrir a estados financieros secundarios para conocer el análisis de algún elemento de ellos. De esta forma, las entidades mercantiles dedicadas a la compra y venta de mercancías agruparán sus ingresos y egresos en conceptos accesibles.

El elemento básico para elaborar un estado de resultados los constituye el principio de contabilidad generalmente aceptado, denominado principio de revelación suficiente. Este principio orienta a elaborar los estados financieros, en este caso, el Estado de Resultados en forma clara, veraz y oportuna. Ante esta consideración no existe una forma única para presentar este estado financiero, pero la más aceptada y aplicada es la denominada "Forma de Reporte".

1.6.1.15 Clasificación de Ingresos.

Considerando el ingreso como el elemento del estado de resultados que representa utilidad y, por ende, de capital, toda entidad cuenta con ingresos que en función a su actividad o giro se clasifican en:

- a) Ordinarios.
- b) Extraordinarios.

Toda entidad mercantil posee como objetivo el lucro, esto es, realizar operaciones que produzcan utilidades, de esta forma cada entidad al tener su actividad o giro específico efectuará operaciones concretas para obtener de ellas utilidades siendo tales precisamente sus ingresos ordinarios.

Al hablar de ingresos extraordinarios es tratar de utilidades extraordinarias. Considerando que extra es una preposición que significa fuera de, es de comprender por deducción que ingreso extraordinario será aquel ingreso fuera de los ordinarios. De esta manera, toda operación realizada por la entidad al generar utilidad, sin ser por su actividad propia, representa un ingreso extraordinario.

1.6.1.16 Clasificación de Egresos.

El segundo elemento de los estados de resultados son los egresos. Este elemento representa las pérdidas en consecuencia, disminuciones de capital. Este tipo de operaciones son múltiples, motivo por el cual no todas las entidades poseen utilidades. El hecho de tener ventas no implica existencia real de utilidades, dado que estas se ven disminuidas por los egresos que en función a su actividad o giro, pueden clasificarse en:

- a) Ordinarios.
- b) Extraordinarios.

Los ingresos ordinarios son las pérdidas en que se incurre para crear los ingresos ordinarios y por otro lado, los egresos extraordinarios son los ocasionados por conceptos diferentes a la actividad propia de la entidad, por tanto todo egreso o pérdida que no tienda a generar ingresos ordinarios será egreso extraordinario.

1.6.1.17 Interpretación del Estado de Resultados.

Toda entidad, cualquiera que sea su magnitud, tiene la necesidad de contar con control e información financiera.

La contabilidad, debe hacerse hincapié, no es en ningún momento la simple actividad de registrar las operaciones realizadas por una entidad. La contabilidad genera información presentada en los estados financieros. Con el Balance General muestra razonablemente la situación financiera de una entidad a una fecha determinada, manifestando en consecuencia las utilidades o pérdidas. El estado de resultados tiene plena relación con el Balance General, en virtud de informar sobre el origen de los resultados logrados por una entidad, en función a las operaciones realizadas en un periodo determinado. Esto es, informar de los ingresos y egresos que han originado las utilidades o pérdidas presentadas en el Balance General.

El Estado de Resultados, por consecuencia, al igual que el Balance General, constituyen estados financieros potencialmente útiles, en tanto no se tomen decisiones como fruto de la información proporcionada. Si el Balance General por muy técnica que fuese su elaboración, si no se analiza y si no se toman decisiones en función a su interpretación no es útil. El Estado de Resultados tampoco es una excepción, no es importante tener como objetivo contar con Estados de Resultados periódicos, elaborarse con la mayor técnica y automatizados en perfecta forma, el objetivo debe ser, analizar la información en él expresada, para interpretarla y tomar decisiones de cualquier naturaleza. Un Estado de Resultados es útil cuando su información se analiza, se interpreta y se toman decisiones.

El Estado de Resultados puede analizarse e interpretarse a través de una observación superficial que permita descubrir aspectos relevantes y tomar decisiones inmediatas. Pero resulta más completo cuando se interrelacionan los conceptos que lo integran para

lo cual una simple observación resulta insuficiente, haciéndose necesario el uso de las matemáticas:

Utilidad del Ejercicio

Ventas

Así pues, el dividir la utilidad del ejercicio entre las ventas, se obtiene como cociente la representación de la relación entre cada peso vendido una ganancia determinada lo cual sirve para confirmar si están obteniéndose las utilidades previamente esperadas, para en caso contrario investigar la causa fundamental que genera el tener mayor o menor utilidad de la establecida como objetivo.

Por otro lado, el relacionar utilidad bruta sobre ventas, permite conocer la utilidad bruta por cada peso vendido.

Utilidad bruta

Ventas

Otro aspecto también muy importante es la periodicidad de elaboración e interpretación de el Estado de Resultados para contemplar cualquier desviación negativa de la entidad y tomar decisiones que después de haber sido analizado e interpretado dicho estado financiero, sean las adecuadas para evitar las pérdidas o alcanzar las utilidades fijadas. Por tal circunstancia se divide la vida de la entidad en periodos iguales, normalmente de un año, para conocer después de un año de operaciones las utilidades o pérdidas experimentadas, compararlas con las correspondientes a las de años anteriores y, de esta forma, disponer de mayores elementos de juicio para interpretar las variaciones suscitadas.

1.6.2. ÁREAS DE AUTOMATIZACIÓN EN EL PROCESO CONTABLE.

La auditoría tiene como finalidad la de verificar la eficiencia del control establecido, la debida presentación de los estados financieros y la veracidad de las cifras en ellos expresadas, y tendrá necesidad de comprobar la existencia de las operaciones llevadas a cabo por una entidad. De esta manera, todas las operaciones deben hacerse constar en documentos que manifiesten una serie de datos como fecha de la operación, naturaleza, importe, etc. encaminados a demostrar en todo momento la veracidad de las operaciones. A dichos documentos se les conoce como comprobante. Estos documentos forman parte del control establecido, permitiendo verificar en todo momento las operaciones, por parte de la auditoría interna, externa o cualquier otra persona según el organigrama de cada entidad. Además facilitan la tarea de capturar o recopilar los datos de las operaciones realizadas por la entidad, cubriéndose así la segunda etapa del proceso de control estructurado como se muestra en el siguiente diagrama (Fig. 1.6.2.A. Diagrama del proceso de control estructurado):

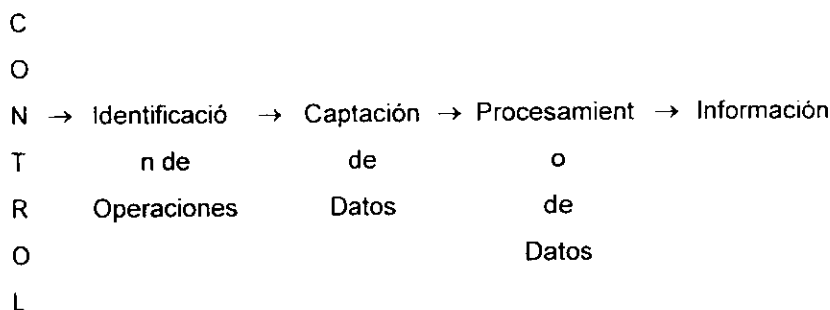


Fig. 1.6.2.A. Diagrama del proceso de control estructurado.

El control, en su fase de captación de datos a través de los comprobantes, exigirá que estos reúnan varios requisitos. Algunos de ellos deberá contener información exigida por entidades regulatorias externas como Secretaría de Hacienda y Crédito Público (registro federal de causantes, fecha, cédula, etc.)

Como etapa inicial del proceso de control deben identificarse las operaciones realizadas por la entidad y como segunda fase captar los datos de tales operaciones disponiendo de los comprobantes correspondientes.

Identificadas las operaciones y captados sus datos por medio de los comprobantes se tendrá una situación favorable para iniciar la tercera etapa del proceso de control, la cual comprende, el procesamiento o registro.

Para contar con información útil y confiable es también necesario afectar el activo, pasivo, capital, ingresos y egresos, exclusivamente por las cifras asentadas en los comprobantes. Esto permite cumplir con el principio del valor histórico original el cual invita a procesar o registra los datos de las operaciones realizadas por la entidad a su valor real.

El procesamiento de datos, conocido también como registro de datos, consiste en controlar los aumentos y disminuciones de cada uno de los conceptos del activo, pasivo, capital, ingresos y egresos como consecuencia de las operaciones realizadas por la entidad.

Para facilitar el proceso de contabilización se introduce el concepto de esquemas de mayor el cual tiene como finalidad registrar en un diagrama todas las operaciones realizadas y dependiendo del concepto del que se trate, se registrarán en el "Debe" (lado izquierdo del diagrama) o en el "Haber" (lado derecho).

Resumiéndose se tiene:

Esquema de mayor de cualquier concepto de activo	
Debe	Haber
Aumenta	Disminuye

Y de forma inversa:

Esquema de mayor de cualquier concepto de pasivo y capital	
Debe	Haber
Disminuye	Aumenta

Al procesar o registrar cualquier tipo de operación realizada por alguna entidad, sin interesar su naturaleza, la misma cantidad colocada en el debe se anotará en el haber, lo cual constituye la **Teoría de la Partida Doble**.

Así es posible clasificar cada uno de los conceptos que se presentan en cualquier Balance General y Estado de Resultados con cuentas por lo que los esquemas de mayor controlan los aumentos y disminuciones de la cuenta que está representando. A estos movimientos realizados en el "debe" de cualquier esquema de mayor se denomina *cargo* y su contraparte, todo registro en el haber se conoce como *abono* y la suma de cargos de cualquier cuenta recibe el nombre de *movimiento deudor* y la suma de abonos de cualquier cuenta se denomina *movimiento acreedor*. Según la naturaleza de las cuentas, el movimiento deudor y acreedor puede representar aumento o disminución: en virtud de ello, la diferencia entre los movimientos (deudor menos acreedor) se denomina *saldo*.

Por lo tanto, en los estados financieros siempre se presenta el saldo de las cuentas.

Así pues, estas sencillas reglas son las que determinan el "cómo" realizar cualquier automatización de los procesos contables, básicos para la derivación del resto de la información necesaria para la oportuna toma de decisiones.

CAPITULO 2
PLANTEAMIENTO DEL PROBLEMA
Y PROPUESTA DE SOLUCIÓN

2. PLANTEAMIENTO DEL PROBLEMA Y PROPUESTA DE SOLUCIÓN.

2.1. CASA POSTPRODUCTORA DE CINE Y VÍDEO.

La producción de vídeo constituye la actividad de poner a disposición de los clientes, el equipo propio de filmación, ya sea para exteriores o para interiores. Esto es, las cámaras de vídeo, las luces, la transportación, los generadores eléctricos portátiles, los foros de filmación, micrófonos, cables, etc. El resultado final es uno o varios rollos de cinta de cine, o cartuchos de vídeo de cualquier formato listos a ser postproducidos.

La postproducción consiste en retocar, corregir, mejorar, añadir efectos y sonidos especiales, a la cinta o cartuchos previamente filmados, todo esto en salas debidamente equipadas con máquinas especiales y operadores capacitados para realizar todo aquello que el cliente desea sea realizado con su material.

Dependiendo el equipo portátil de filmación que tenga una compañía para ser alquilados, y de las salas con las máquinas especiales de edición y manejo de imágenes con que cuente, serán los servicios que esta compañía pueda ofrecer al público, convirtiéndose así en una compañía productora, postproductora o ambas, de vídeo.

Nuestro trabajo se enfocará a cualquiera de estos tres tipos de empresas.

En la fig. 2.1.A detallamos las áreas que conforman el organigrama de la empresa, esto con el fin de ubicar y marcar los límites hasta dónde el sistema a analizar, diseñar y desarrollar, podría cubrir.

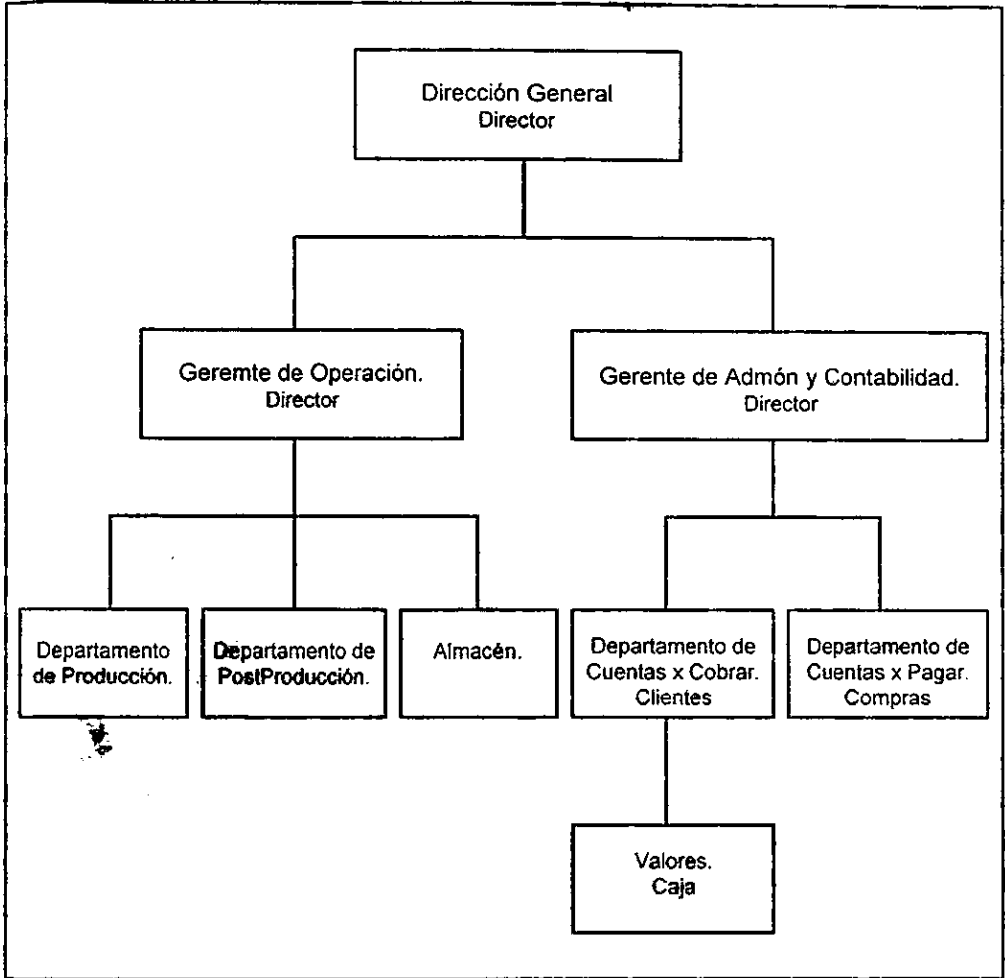


Fig. 2.1.A. Organigrama de la empresa.

Dirección General. Es el área donde se fijan los lineamientos y políticas básicos y esenciales de la empresa a un nivel genérico. Así mismo, se retroalimenta de la información que recibe de las áreas que directamente dependen de ella. Requiere para realizar sus funciones de manera más eficiente y real, de la información, a nivel gerencial, oportuna, clara, precisa, compacta y actualizada.

Gerencia de operaciones (ventas). Es el área quien regula las políticas y programas de ventas, asigna los precios de venta, fija los servicios y productos que la empresa ofrece a sus clientes, investiga nuevos productos, equipos y máquinas de video, realiza la proyección de la venta y entrega y fomenta la facturación como su principal función. Todo esto lo realiza a través de los departamentos que dependen de ella, reportándole sus resultados.

Departamento de producción. Es el área encargada de realizar el alquiler del equipo de producción a los clientes, hace la cotización del equipo, la reservación, la cancelación de órdenes y la generación de la orden de trabajo. Una orden de trabajo es un documento donde se indica la relación de servicios, en este caso de producción, solicitados por el usuario, donde se especifica las fechas, la cantidad, el tiempo de utilización y observaciones adicionales.

Departamento de postproducción. Es el área encargada de realizar el alquiler de las salas de postproducción a los clientes, Estas salas cuentan con máquinas y servicios inherentes que definirán los costos involucrados al reservar dicha sala. Este realiza también la cotización y la reservación previas a un alquiler definitivo. A su vez, lleva el control de los operadores de los equipos en las salas para efecto de otorgar comisiones. Con la información que en esta área se genera, se da respuesta a las necesidades del usuario para programar su solicitud de alguna sala y operador en especial.

Almacén. Aquí se lleva el control de las entradas y salidas de materiales que se ponen a disposición de los clientes para su venta. Mediante órdenes de entrada y de salida de materiales, se lleva el control de existencias y se programan pedidos futuros, según también, de la demanda que las áreas de producción y de postproducción le indiquen.

En el organigrama se muestra la dependencia que tiene el departamento de almacén con el departamento de postproducción, esto es porque directamente el almacén no puede facturar, esta función es propia de postproducción, quien a su vez recibe las órdenes de trabajo de producción para ser integradas a una factura.

Gerencia de Administración y Contabilidad. En esta área se regulan los lineamientos contables que toda empresa debe seguir, es la responsable que las obligaciones fiscales sean cumplidas claramente. Sus principales funciones son:

- Generación y conciliación de documentos (como pagos de impuestos y declaraciones periódicas) para el fisco.
- Aplicación de normas fiscales y legales.
- Generar y controlar la nómina.
- Controlar las compras y gastos de la empresa (control de Cuentas por Pagar).
- Controlar el flujo de efectivo.
- Controlar las Cuentas por Cobrar (facturación normal y cartera vencida).
- Control y conciliación bancaria.
- Control de la caja chica (gastos menores no comprobables con una factura).
- Aplicación contable de todos los movimientos que afecten a la contabilidad, esto es, generación de pólizas que respalden las diversas ventas, compras y gastos que en el ejercicio diario de la empresa se tienen.
- Generación y control de pólizas modelo, esto es, machotes de operaciones constantes y repetitivas que cada área se encuentra en su proceso normal de trabajo.
- Controlar los cierres en los meses y años contables para finalizar ejercicios contables.
- Generación de reportes de resultados, balances, auxiliares y de estados financieros para retroalimentar otras áreas y a la Dirección General.

Cuentas por Cobrar (Clientes). Es el área donde se da seguimiento al comportamiento crediticio de los clientes, y una vez detectada la cartera vencida, proceder a su cobro utilizando métodos que dependerán del nivel de atraso de la deuda. Entre las principales funciones que esta área tiene son:

- Estudia y da seguimiento a la cartera vencida.
- Califica a los clientes para especificarle sus parámetros crediticios como límite y meses de crédito.
- Controla, actualiza y depura el catálogo de clientes.
- Lleva a cabo el cobro de los pagos vencidos en las fechas adecuadas, según lo acordado con los clientes.
- Envía los estados de cuenta mensualmente a los clientes.
- Realiza la cancelación de las facturas que lo ameriten.
- Genera y entrega los reportes de la cobranza junto con un pronóstico de la cobranza a una fecha determinada.
- Utiliza y da seguimiento a los procesos jurídicos para realizar el cobro de clientes morosos que así lo requieran.

Cuentas por pagar (Compras). Es el área encargada de controlar las compras fijas, como pago de renta, de teléfonos, de luz, vigilancia, estacionamiento; y variables como compra de nuevo equipo, viajes, publicidad, exposiciones; todos éstos necesarios para el buen funcionamiento de la empresa. También entrega la proyección de gastos programados básica para presionar al departamento de Cuentas por Cobrar y tener flujo necesario y a tiempo.

Valores (Caja). Es donde se reciben y se registran, todos los pagos de los clientes. Estos pagos pueden ser: anticipados (los que no pagan ninguna factura presente sino posterior), parcial y total, y para una o varias facturas del mismo cliente. La información de valores pasa directamente a Cuentas por Cobrar para que éste tenga conocimiento de los pagos.

2.1.2. DESCRIPCIÓN DEL PROCESO ACTUAL.

Los procesos actuales a los que este trabajo se enfocarán, se ven condensados en dos ciclos principales y en un proceso de conciliación:

- El ciclo de la venta.
- El ciclo del cobro.
- Conciliación de la información.

2.1.2.1. El Ciclo de la venta.

En la fig. 2.1.2.1.A se muestra completamente el esquema de este ciclo.

La fase A nos muestra el proceso desde que el cliente solicita un servicio a la empresa y ésta le atiende, entregando una orden de trabajo. El orden de los procesos es como sigue:

1. El cliente solicita un servicio de producción, de postproducción de compra de materiales o bien, una combinación de las tres anteriores.
2. Al cliente se le elabora una cotización de los servicios productos solicitados, entregándole el estado de la disponibilidad de aquellos y de los precios aproximados de costo.
3. El cliente puede reservar equipo y salas en los días y horarios disponibles.
4. El cliente utiliza los servicios solicitados pudiendo existir modificaciones a lo originalmente reservado, agregándosele servicios, tiempos y productos o no utilizando alguno solicitado.
5. En caso de que el almacén no cuente con las existencias requeridas, o bien, que el límite mínimo de un producto en especial se haya rebasado, se solicita al

departamento de Compras el suministro de dicho producto, avalado por una orden de entrada de material al almacén.

6. Una vez terminado el servicio con el usuario, cada área llena en un mismo formato cada uno de los servicios realizados, el tiempo y el costo y el operador que atendió dichos servicios, en este documento, también se le anexan los materiales que el cliente haya necesitado para terminar su labor.

En este punto, se tiene entonces un documento, la orden de trabajo, con los servicios de producción, postproducción y materiales que el cliente realmente utilizó.

La fase B nos muestra la integración de las funciones del departamento de Operación con el de Contabilidad, donde un auxiliar transcribe la orden de trabajo y la lista de materiales en una factura, una copia será entregada al cliente y la otra servirá para informar a los departamentos de Cuentas por Cobrar, Valores y Contabilidad. Estos procesos se resumen en:

1. Se transcribe la orden de trabajo y la lista de materiales en una factura con varias copias, una de éstas se entrega al cliente.
2. Una copia de la factura es entregada al departamento de Cuentas por Cobrar, quien se pone en contacto con el cliente, y si es nuevo en la empresa, le solicita información (días de revisión, de pago, referencias y la relación de personas que pueden solicitar servicios), para asignarle en su caso, parámetros de crédito; con estos datos, actualiza los saldos para llevar un control actualizado de su cartera de clientes.
3. Este mismo departamento revisa diariamente las facturas que deben ser enviadas a revisar con el cliente, y sobretodo, aquellas que deberán ser cobradas. Para las facturas que ya han pasado su límite de meses de crédito, se procede a elaborar cartas de recordatorio, llamadas telefónicas o visitas personales y así dar seguimiento a la cobranza.

4. En caso de llegar a procesos jurídicos en el cobro de una cuenta atrasada, entonces se entabla un juicio de demanda de cobro y se da seguimiento al trámite y desarrollo del caso.
5. Por otro lado, una copia de la factura se entrega al departamento de Contabilidad para que sea generada a póliza correspondiente. Actualmente tienen esta información en hojas electrónicas (Excell), y es de aquí de donde se obtiene la información para generar los reportes contables diarios, mensuales y anuales. De la misma forma, el almacén entrega copias de sus órdenes de entradas de materiales para que de igual manera, se capturen las pólizas contables necesarias.
6. Una copia de la factura también es entregada al departamento de Valores para que éste tenga una relación de los posibles pagos que puede recibir.

2.1.2.2. El ciclo de cobro.

En la fig. 2.1.2.2.A se muestra el diagrama del proceso actual de este ciclo. El ciclo de cobro abarca la recepción del pago, su control, descargo en la cuenta del cliente y su reflejo contable, estos puntos son:

1. El cliente hace llegar a la empresa un pago. El puede ser en efectivo, en cheque, en tarjeta de crédito, traspaso bancario electrónico y por nota de crédito (documento que la misma compañía emite para respaldar algún descuento o bonificación que se haya realizado previamente con el cliente). El pago puede ser anticipado, pago total o parcial de una factura o pagar más de una factura.
2. Valores recibe el pago, registra en un recibo, los datos de quien paga y de qué paga, los montos y los medios de pago. Entrega una copia al cliente o a su representante, otra la conserva para posteriormente hacer sus reportes de corte de caja y envía otras copias al departamento de Cuentas por Cobrar y Contabilidad.
3. Diariamente, Valores hace un documento llamado Cifras de Control donde indica todos los pagos recibidos en el día, detallando el cliente, las facturas pagadas y los

montos. Este documento es revisado por Cuentas por Cobrar y cotejadas con los documentos que avalan dichos pagos.

4. El departamento de Cuentas por Cobrar, junto con los recibos y las Cifras de Control, actualiza sus archivos de cada cliente para tener el nuevo saldo pendiente o a favor de los clientes que ese día realizaron algún tipo de pago.
5. Algunas veces, es necesario cancelar facturas, o ajustar los pagos que se hayan hecho, el departamento de Cuentas por Cobrar se encarga de estas funciones, informando a su vez, a Contabilidad para que ésta haga también lo necesario y no exista diferencia en la información que maneja cada uno.
6. El departamento de Contabilidad captura nuevamente en sus archivos de Excell las pólizas generadas de la recepción de los pagos así como de las cancelaciones y ajustes, que se le hayan reportado.

2.1.2.3. La conciliación de la información.

En la fig. 2.1.2.3.A se observa el esquema del proceso de este punto. Mensualmente y ocasionalmente otros días, según sea requerido por parte de la Dirección General, cada departamento debe entregar informes del estado actual de cada una de sus actividades en cantidad y montos, es decir, el departamento de Operación debe entregar un reporte con la especificación de lo cotizado, reservado y principalmente de lo facturado, el departamento de Cuentas por Cobrar un informe con la información de la cartera vencida, la recuperación de la cartera en períodos específicos, el pronóstico de la cobranza, la relación de la cartera conflictiva y comparativos con otros meses, Contabilidad debe entregar la información de lo aplicado contablemente en estos y otros rubros de la operación diaria de la empresa.

En este punto, todos los reportes deben coincidir en montos y cantidades, cambiando únicamente el enfoque con que se genere cada informe.

Para esto, los responsables de cada área trabajan juntos para conciliar sus cifras y entregar entre ellos un sólo documento general pero completo y consistente. De esta información Contabilidad utilizará para llevar el registro fiscal y en su momento, realizar las declaraciones y pagos correspondientes.

Con estos tres puntos, englobamos el proceso actual que se lleva a cabo en la empresa cuyos problemas, generados a causa de estos mismos procesos y de no contar con un sistema integral, confiable y único, se detallarán en las secciones siguientes.

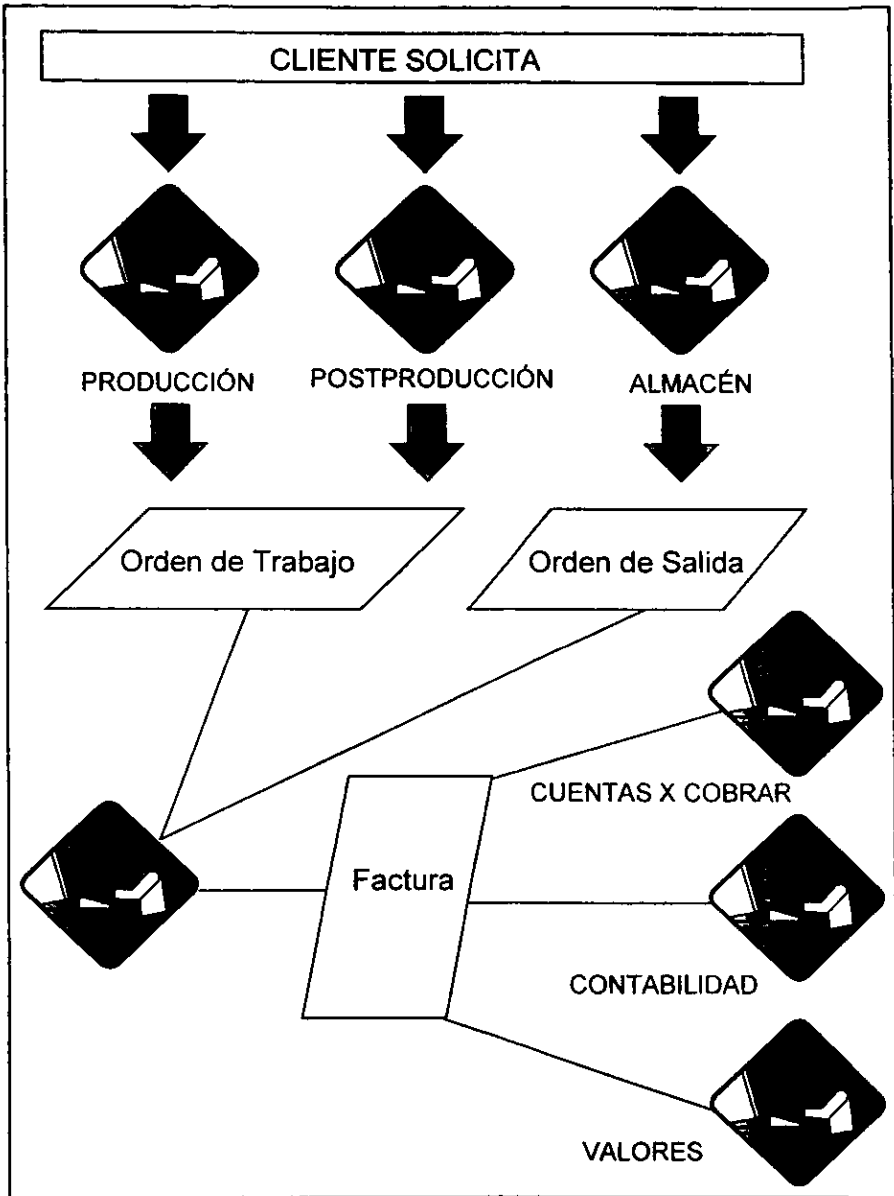


Fig. 2.1.2.1.A El Ciclo de Venta.

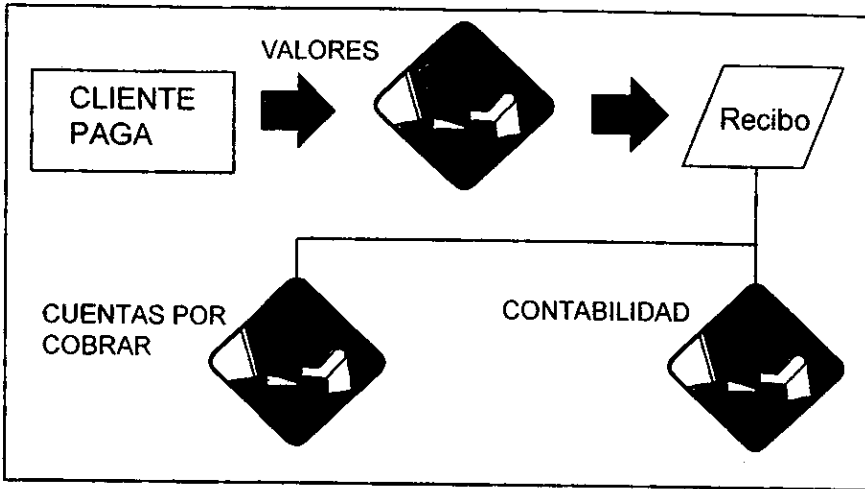


Fig. 2.1.2.2.A. El Ciclo de cobro.

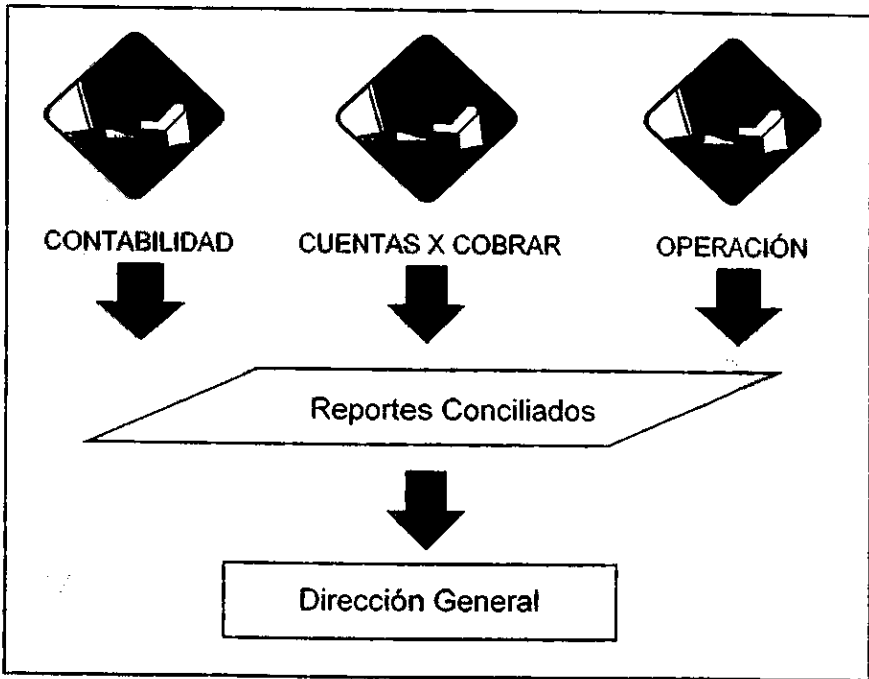


Fig. 2.1.2.3.A. La conciliación de la información.

2.2. PROBLEMÁTICA ACTUAL.

En este punto, se analizará la problemática actual de la organización a fin de tomar esta información y, junto con los requerimientos específicos planteados por el usuario, desarrollar un modelo de solución.

Es importante entender claramente cuales son las áreas que influyen en la posición financiera de la empresa a fin de ofrecer una solución de automatización, buscando con esto, optimizar el proceso que actualmente se realiza en diferentes áreas en forma manual. La importancia del entendimiento radica en que no siempre una solución de automatización significa la solución total de la problemática. Muchas veces, la automatización debe ir acompañada de nuevas definiciones de políticas de operación, de cambios en la forma de pensar de las personas involucradas de la empresa y, por supuesto, los costos incurridos con la implantación de la nueva solución.

Dentro de este análisis es importante tomar en cuenta la arquitectura actual de cómputo de la empresa y sus expectativas de inversión y los posibles ahorros que puede presentar la nueva solución. Pero una parte son los costos (ahorro en la operación, inversión), otra es la ventaja competitiva dada a través de la oportunidad y confiabilidad de la información.

2.2.1. ÁREAS DE INTERÉS.

De acuerdo a lo descrito en el punto 2.1, donde se describe el organigrama de la institución, así como el proceso actual de operación, existen áreas donde no es necesario el control contable, o más bien, dichas áreas no generan ninguna afectación contable que deba ser registrada dentro del sistema a desarrollar.

Así pues, el proceso de determinar cuales son las áreas en las que deba registrarse la contabilidad, es simple: las de mayor impacto y flujo de operación.

La empresa cuenta inicialmente con procesos de control de las áreas:

- Reservasiones.
- Almacén.

Estos procesos, como se discutió anteriormente, actúan independientes uno del otro y los resultados contables de ambos es a través de documentos en papel (pólizas) que deben ser ingresadas al proceso contable en forma manual.

Ante esta perspectiva, las áreas que se seleccionaron para la automatización del proceso contable y control de información para el mejoramiento de la planeación financiera, son:

- Facturación.
- Contabilidad.
- Cuentas por Cobrar.
- Valores.

Ubicando estas áreas dentro del organigrama, vemos la relación existente con las áreas previamente en operación y es lógico pensar en el desarrollo de un componente unificador a través del proceso contable. (ver Fig. 2.2.1.A).

CASA POSTPRODUCTORA DE CINE Y VIDEO
SITUACION ACTUAL

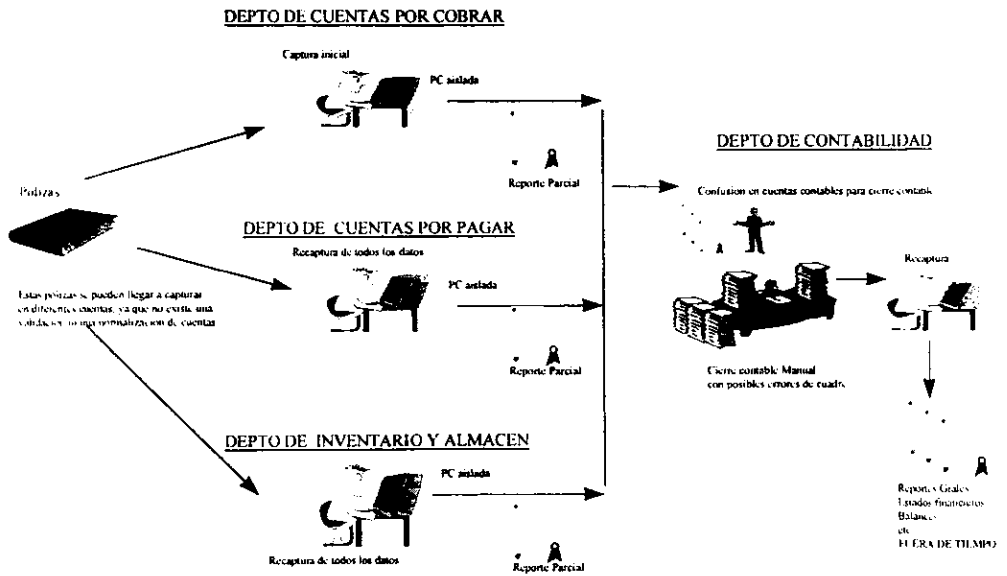


Fig. 2.2.1.A. Organigrama de la empresa y Áreas de interés.

2.2.2. CONSECUENCIAS DEL PROBLEMA.

Como se mencionó en el punto anterior, la comprensión del problema nos lleva a planteamientos claros y sencillos de solución. En este apartado veremos cuales son las consecuencias de la falta de procesos de administración en el aspecto contable en la operación de la empresa.

A través de la esquematización del problema, es fácil determinar muchas de las consecuencias de la falta de un control contable adecuado (ver Fig. 2.2.2.A).

CASA POSTPRODUCTORA DE CINE Y VIDEO
SITUACION PROPUESTA

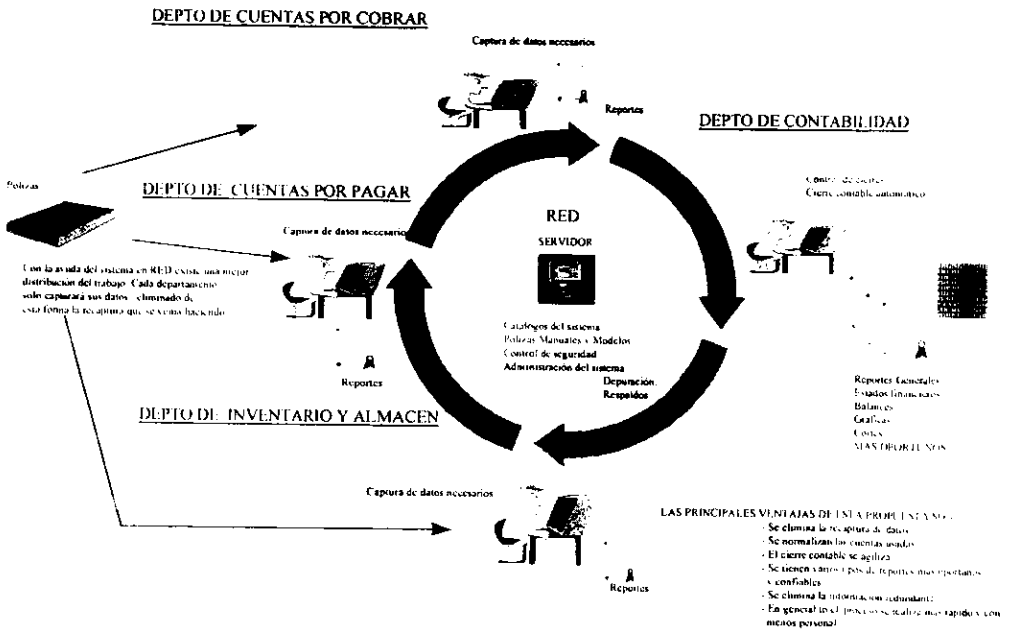


Fig. 2.2.2.A. Esquema de áreas no integradas en su operación.

A continuación, mencionamos los principales problemas sobre los cuales se enfrentaba la empresa en su operación:

- Primero, es necesario indicar que las necesidades de la empresa no son cubiertas por paquetes comerciales (SOI, COI) por presentar su operación en el manejo de cuentas sólo hasta el tercer nivel.
- No se cuenta con una contabilidad confiable, en el sentido de que cada área generaba sus propias pólizas, con sus propias reglas y en sus propios tiempos.
- La atención al cliente no era la adecuada: el área de producción y postproducción definía sus propias políticas de crédito y descuentos, así como la asignación de materiales que debiera estar en concordancia con el manejo de almacén.
- Una vez que se realiza una reservación de sala, también se genera una orden de material, el cual podía incrementarse durante el periodo de trabajo.
- Muchas veces, el material no era cobrado por no existir una liga entre almacén y facturación.
- Podía darse el caso de que postproducción cancelara la orden de trabajo y almacén emitiera el material.
- Muchas veces no concordaban las emisiones de almacén contra las ventas registradas por postproducción y por tanto en la contabilidad.
- El control de inventario era inadecuado: sobrepoblado en algunos casos y faltantes en otros.
- No existía posibilidad de planear stocks en almacén.

- El área de cobranzas no contaba con la información necesaria, derivada por reservaciones y almacén, para ejercer adecuadamente las obligaciones de las facturas pendientes de cobro.
- En el caso de administración de clientes, no se contaba con información suficiente para determinar cuando el cliente tiene sus fechas de recepción de facturas, entrega de contrarecibos, de pago, etc. Esto provocaba que la cobranza se extendiera más de lo convenido en las órdenes de trabajo.
- El manejo de la operación es a través de crédito y si no se cuenta con un historial de cada cliente en cuanto a como se ha desempeñado en trabajos anteriores, era difícil establecer en cada orden el monto de crédito permitido o bien negarlo.
- No se contaba con la posibilidad de emitir estados de cuenta por cliente a fin de informarle los movimientos realizados, el crédito disponible, pagos de facturas pendientes, cartas de llamado de atención, de felicitación por su desempeño, etc.
- Tampoco era posible informar al consejo de socios la información necesaria para establecer las políticas de administración. Tal información se refiere a conocer el estado de la cartera vencida, la calificación de cartera, los volúmenes de ventas, cobranza real contra facturado, el estado financiero de la empresa, ocupación de activos y servicios de operadores, etc.
- Tampoco era posible realizar una planeación adecuada en el área de valores para el control de los gastos fijos de la empresa: caja chica, renta, luz, teléfono, etc.
- No existía la posibilidad de planear las actividades de los operadores de equipo, por lo que muchas veces el trabajo era más cargado en ciertos días y otros no había actividad.

Como se mencionó en el primer capítulo, la contabilidad es el medio que, por sus métodos y técnicas, permite controlar y visualizar, a través de estados financieros, información clara, veráz y oportuna de todos los recursos de la entidad y solo así poder tomar decisiones, consientes de sus consecuencias.

Es por esto que, dados todos los problemas mencionados anteriormente, no era posible tomar decisiones en base a el estado que guardaba la empresa a fin de buscar un mejor posicionamiento del mercado sobre el cual se maneja la empresa.

2.3 REQUERIMIENTOS DEL USUARIO

2.3.1 Requerimientos generales

Actualmente en esta compañía de cine y vídeo están funcionando unos sistema para realizar reservaciones de los servicios que se prestan y para llevar el control del almacén, además de utilizar un paquete para realizar su contabilidad. El usuario desea integrar estos sistemas de reservaciones y almacén con el nuevo sistema de contabilidad con el propósito de cubrir las siguientes necesidades generales:

- Eliminación de redundancia de datos
- La información sea única.
- La información este en un punto central y que pueda ser accesible desde cualquier área con su respectiva seguridad.
- La información este a tiempo.
- Elimine la recaptura de datos entre las diferentes áreas involucradas.
- El sistema sea amigable
- Sea fácil de usar
- Contemple pantallas de captura gráfica tipo Windows para cada uno de los casos que se manejan en las áreas involucradas.
- Que realice validación a nivel campo para evitar errores de captura.
- Contemple una seguridad para todas áreas involucradas, existiendo niveles y facultades para efectuar operaciones de alto riesgo.
- Emisión de gráficas para la toma de decisiones.
- Contabilidad automática al realizar movimientos desde cualquier área.
- Se adecue a la plataforma tecnológica que esta siendo utilizada por los sistemas de reservaciones y almacén:
 - Ambiente gráfico Windows de Microsoft
 - Datos almacenados sobre la base de datos Access de Microsoft.

- El sistema debe ser flexible para poder modificar ordenes de trabajo y facturas:
 - Manejo de diferentes descuentos sobre precios establecidos.
 - Poder realizar una sola factura o varias facturas e intercambiar su contenido según convenga al cliente.
 - Esta flexibilidad debe ser manejada con su respectiva seguridad para que no cualquier persona pueda realizar estas modificaciones.

Esta flexibilidad es muy importante en este tipo de empresas ya que muchas veces se llegan a acuerdos personales con el cliente y el sistema debe poder soportarlos (el sistema no debe ser un obstáculo para no poder realizar una venta).

- El sistema contemple al menos a tres áreas donde se realizan actualmente funciones de contabilidad:
 - Contabilidad
 - Cuentas por cobrar
 - Valores.
- Aumento de la productividad al cubrirse los puntos anteriores.

2.3.2 Requerimientos particulares

El usuario esta contemplando los siguientes requerimientos específicos para cada una de las áreas involucradas:

Almacén

- Cuando se registre una entrada de material se debe realizar los asientos contables de manera automática para evitar la recaptura.
- Cuando se registre una salida (venta), esta tiene que estar respaldada por una factura. Con esta factura se libera la orden de materiales para reflejarlo de manera automática en el inventario para evitar recaptura.

Contabilidad

- Control de cuentas, es decir realizar una unificación y estandarización de ellas
- Manejo de cinco niveles de cuentas.
- Manejar tres tipos de pólizas
 - Póliza automática: se genera por la entrada de mercancía al almacén o por facturación.
 - Póliza manual : se genera por entrada de documentos y ajustes.
 - Póliza modelo.: son pólizas estándares existentes, o bien se pueden realizar una nueva póliza de acuerdo a las necesidades de una área.
- Control de cierres contables mensuales y anuales
- Control de usuarios
- Emisión de los reportes de balances, auxiliares y estados financieros.

Cuentas por cobrar

- Control sobre cartera vencida
- Historia sobre comportamiento de clientes.
- Crédito a clientes según su comportamiento.
- Lista negra de clientes o morosos.
- Pronósticos sobre cobranza
- Calendario de revisiones.
- Aumentar al catalogo de clientes (catálogo existente en el sistema de reservaciones) la siguiente información:

- Días de reservación
- Días de pago
- Limites de crédito
- Meses de crédito
- Saldos

- Emisión de los siguientes reportes:
 - Catalogo completo de clientes
 - Saldos vencidos y antigüedad de saldos
 - Revisión de facturas
 - Estados de cuentas de clientes

Valores

- Control de pagos totales, parciales o anticipados
- Todo lo introducido se refleje de manera contable automáticamente.

- Poder realizar diferentes tipos de ajustes:
 - Cancelación de facturas
 - Cheques devueltos
 - Pagos no aplicados

- Emisión de los reportes de cifras de control y recibos de pagos.

2.3.3 Recopilación y clasificación de información

Como se menciona en el inciso 2.3.2 actualmente están funcionando los sistemas de reservaciones y almacén. Además de llevar su contabilidad de manera aislada con un paquete de contabilidad.

Estos sistemas tienen las siguientes características:

Reservaciones

Contiene cuatro módulos:

- **Catálogos:** en el se manejan clientes, salas, servicios, maquinas y operadores entre otros. Se emiten reportes de todos estos catálogos
- **Asignaciones:** el cual administra la asignación de maquinas a servicios y servicios a salas
- **Reservaciones:** tiene las funciones de producción, copiado y postproducción. Este modulo se apoya en las características de una interfaz gráfica para presentar de una manera visual el estado (reservado o libre) de todos los recursos.
- **Ordenes de trabajo** que realiza cotizaciones, reservaciones y generación de ordenes.

En la fig. 2.3.3.A se presenta una orden de trabajo.

En la fig. 2.3.3.B se presenta un reporte de clientes emitido por el sistema.

En la fig. 2.3.3.C se presenta el diagrama entidad relación.

Almacén

- **Administra un inventario con el principio de PEPS (primeras entradas primeras salidas)**

- Manejo de stock en todos los productos
- Maneja dos módulos uno para entradas de mercancía y otro para salidas de mercancía.

En la fig.2.3.3.D se presenta el diagrama entidad relación de este sistema.

Contabilidad

La información contable que se pudo recopilar fue el de algunos reportes :

- Cuentas (ver fig. 2.3.3.E)
- Facturas (ver fig. 2.3.3.F)
- Antigüedad de saldos (ver fig. 2.3.3.G).

2.3.4 Plan de trabajo

Un plan de trabajo tiene el objetivo de dividir, organizar y distribuir tareas específicas de un determinado proyecto con el fin de estimar tiempos y recursos que se utilizarán en su desarrollo.

Este plan de trabajo se realizará en base a la utilización de tres recursos (bgc: Blanca Gil Castellanos, ega: Edgar Garcia Adame y mcc: Mario Carrasco Carrasco).

Como las etapas de análisis y diseño son las más importantes en el desarrollo de cualquier sistema, los tres recursos trabajaran en ellas. El tiempo utilizado en el análisis será de 288 horas hombre (hh) , es decir 12 días hábiles trabajando los tres recursos juntos. El diseño se llevara 336 hh repartido entre los tres recursos durante 14 días hábiles.

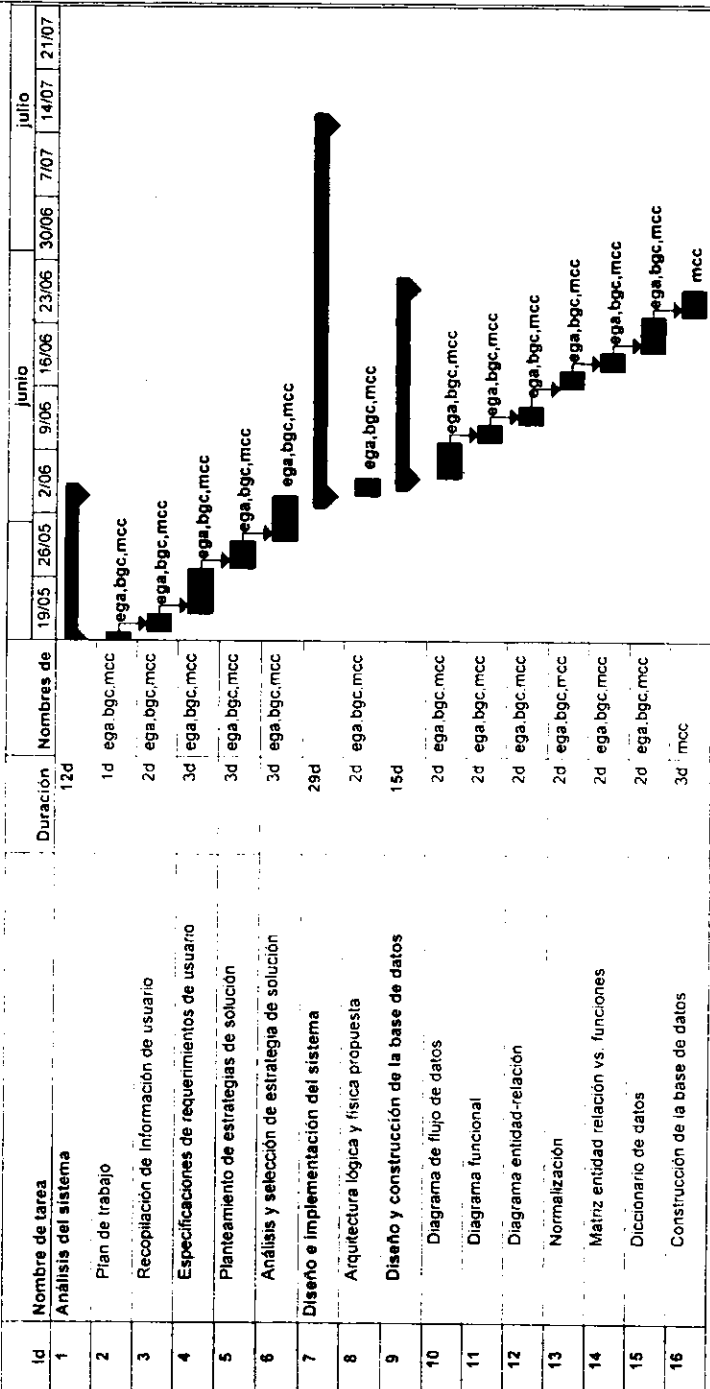
La etapa de construcción se puede dividir en varios módulos: desarrollo del front end, desarrollo de módulos de administración y construcción de la base de datos . Todos estos módulos se pueden realizar en paralelo junto con la etapa de documentación del sistema. El tiempo requerido para esta etapa será aproximadamente de 280 hh durante 12 días hábiles.

Finalmente la etapa de pruebas del sistema serán realizadas nuevamente por los tres recursos utilizándose 210 hh durante 9 días.

Haciendo sumas de los tiempos, el sistema requerirá de 1,114 hh. Estas horas estarán repartidas entre los tres recursos para terminarse aproximadamente en dos meses y medio.

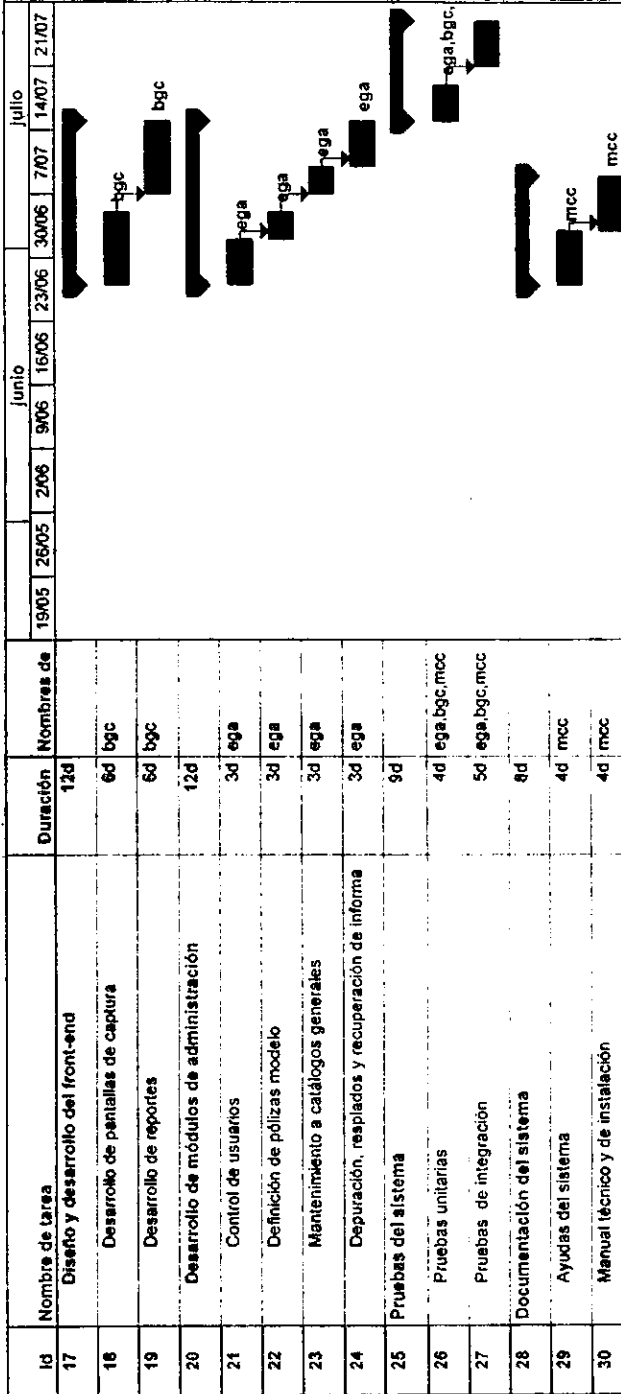
En la fig. 2.3.4.A se muestra el plan de trabajo de una manera esquemática para su mejor apreciación.

**FIG 2.3.4.A PLAN DE TRABAJO
SISTEMA INTEGRAL DE CONTABILIDAD
DE UNA CASA POSTPRODUCTORA DE CINE Y VIDEO**



bgc : Blanca Gil Castellanos.	Tarea		Tarea resumida	
ega : Edgar Garcia Adame	Progreso		Hito resumido	
mcc : Mario Carrasco Carrasc	Hito		Progreso resumido	
	Resumen			

**FIG 2.3.4.A PLAN DE TRABAJO
SISTEMA INTEGRAL DE CONTABILIDAD
DE UNA CASA POSTPRODUCTORA DE CINE Y VIDEO**



bgc : Blanca Gil Castellanos.		Tarea resumida	
ega : Edgar Garcia Adame.		Hito resumido	
mcc: Mario Carrasco Carrasc		Progreso resumido	
		Resumen	

Orden: 2
 Fecha orden: 26 de 97
 Fecha emisión: 13 de 97

CUENTE	Vega Charra Roque	PROYECTO	
RFC	XXX 77442 HHH	TIPO TRABAJO	Postproduccion
DIRECCION	Miguel Laurent 678	SOLICITADA POR	
COLOANIA	Plenitud	ORDEN CLIENTE	
DELEGACION	Ivairo Obregon	DESCRIPCION	
TELEFONO	5 55 21 33	AUTORIZADA POR	
RESERVACION	6	OPERADOR	

Salas

Fecha	Sala	Servicio	Reservado		Utilizado		1/2 Hrs	Precio	Total
			Hr Ini	Hr Fin	Hr Ini	Hr Fin			
26-06-97	Sala Rank	Cine-Video Sin Corrector D1	15:30	16:30	15:30	16:30	2	\$120.00	\$240.00

Maquinas:

Maq. D1.1	15:30	16:30	15:30	16:30
Maq. D1.2	15:30	16:30	15:30	16:30

Fecha	Sala	Servicio	Reservado		Utilizado		1/2 Hrs	Precio	Total
			Hr Ini	Hr Fin	Hr Ini	Hr Fin			
26-06-97	Sala Copiado	Servicio Multifomato	16:30	19:00	16:30	19:00	5	\$90.00	\$450.00

Maquinas:

Maq. VHS.1	16:30	19:00	16:30	19:00
Maq. VHS.4	16:30	19:00	16:30	19:00

Fecha	Sala	Servicio	Reservado		Utilizado		1/2 Hrs	Precio	Total
			Hr Ini	Hr Fin	Hr Ini	Hr Fin			
26-06-97	Sala Henry	Sala digital con 3 VTRs D3	18:00	20:00	18:00	20:00	4	\$200.00	\$800.00

Maquinas:

Maq. D3.1	18:00	20:00	18:00	20:00
Maq. D3.3	18:00	20:00	18:00	20:00

Recibimos y aceptamos a nuestra entera satisfacción los servicios y materiales aquí descritos, estando de acuerdo con los términos y condiciones de ambos lados de esta orden de trabajo. Asimismo, debemos y pagaremos incondicionalmente por ese pagaré a la orden de Telemática producciones, S.A. de C.V. la cantidad de _____ por los servicios antes descritos.

Nombre: _____
 Firma: _____

FIG. 2.3.3.A ORDEN DE TRABAJO

**SISTEMA DE RESERVACIONES
CASAS POSTPRODUCTORAS DE VIDEO**

Catálogo de Clientes

Fecha: 19/05/97
Hoja: 1

Clave	Razon Social	Dirección	Ciudad	Teléfono	C.P.	Entidad
1	Aguilar Anaya Felipe	Lagartos Num 127	Coloma	5 07 64 58	06500	Chihuahua
2	Berajas Amarguilla María	Tehuacanpoc 251	Casa Blanca	5 42 3796	03456	Chihuahua
3	Carnes Seleutas Rocky	CCB 345678 N	Leon Vacaio 10	5 82 31 45	04400	Chihuahua
4	Chavez Martinez Marcelino	DDD 456789 O	Monja Alegre 45-A	6 45 12 57	09089	Chihuahua
5	Diamica Almenica SA de CV	EEE 567890 Q	Hornero 530-34	5 09 50 23	03100	Chihuahua
6	El Faro de la Colonia	FFF 678901 R	Central 12 Loc 11	6 03 51 55	06140	Chihuahua
7	Flores Chavez Arturo	GGG 789012 S	Lomas Verdes 382	5 11 53 66	01090	Chihuahua
8	Garduño Ruiz Duke	HHH 890123 T	Galati 33 Int 34	6 35 90 11	02400	Chihuahua
9	Huerta Andrade Angel	JJJ 901234 UU	Paya Homce 31-3	2 51 29 22	03410	Chihuahua
10	Ibarra Millán Rodio	KKK 012345 V	Boque de Duraznos 34	6 44 21 33	02410	Chihuahua
11	Jurado Meza Antonia	LLL 102936 W	Navegantes y Fundadores 123	2 51 29 44	02400	Chihuahua
12	La Cadena de Oro	MMM 203947	Av. Las Palmas 756	2 58 54 55	14050	Chihuahua
13	Mancera Bartera Eugenia	NNN 364756 Y	Sra. Ana No. 86	6 08 94 66	01010	Chihuahua
14	Napoles Campa Raul	OOO 764954 Z	Penahuyan 2281	6 18 70 77	14310	Chihuahua
15	Ocampo Carmen Jesús	PPP 659403 A	Rancho del Arco Edif 98	6 28 23 88	06784	Chihuahua
16	Pedraza Mendoza Rogelio	QQQ 135791 B	Indipendientes Sur 1234	6 20 35 99	01060	Chihuahua
17	Quintana Conteras Armando	RRR 246802 C	Amaneceran 365-9	5 87 34 77	02300	Chihuahua
18	Reyes Teliez Agustín	SSS 192837 D	Monterrey 566	6 44 68 00	14210	Chihuahua
19	Super Alaska	TTT 029384 E	Pudilco 45	5 12 15 45	04326	Chihuahua
20	Torres Campos Hilario	UUU 185633 F	C. Deportiva Mexica	6 26 86 98	01029	Chihuahua
21	Uribea Chavez Marco	VVV 222899 G	Miguel Jacentis No. 356	2 99 91 23	11800	Chihuahua
22	Vega Chimal Roque	XXX 777444 H	Miguel Lavrent 678	5 55 21 33	06780	Chihuahua
23	ETV PROO	nnnnnnnnnn	Puebla 3134b			
24	Televisión Azteca	www.wwwww				
25	Asstas y ATT	ddddddddd				
26	Doriana Video					
27	Canal 7 de Televisión					
28	FERNNANDO VILLEGAS BAHENA	Urupac, Mich	Capulitico	480 50 497 53		Morelos
29	ALTEC ELECTRONICA CHIHUAHUA	Núm 34	AMPLIACION	14 292 475	31020	Chihuahua
30	Asociación Oaxaqueña de Televisión	PLAN DE AYALA Y RIVERA	CHAPULTEPEC	47544 63646		Oaxaca
		CRESPLO LOCAL G	PARQUE INDUSTRIAL EL SAUCITO			
		JUVENTUD SIN	Centro Cultural Oaxaca			
		Av. Tecnológico				

FIG. 2.3.3.B REPORTE DE CLIENTES

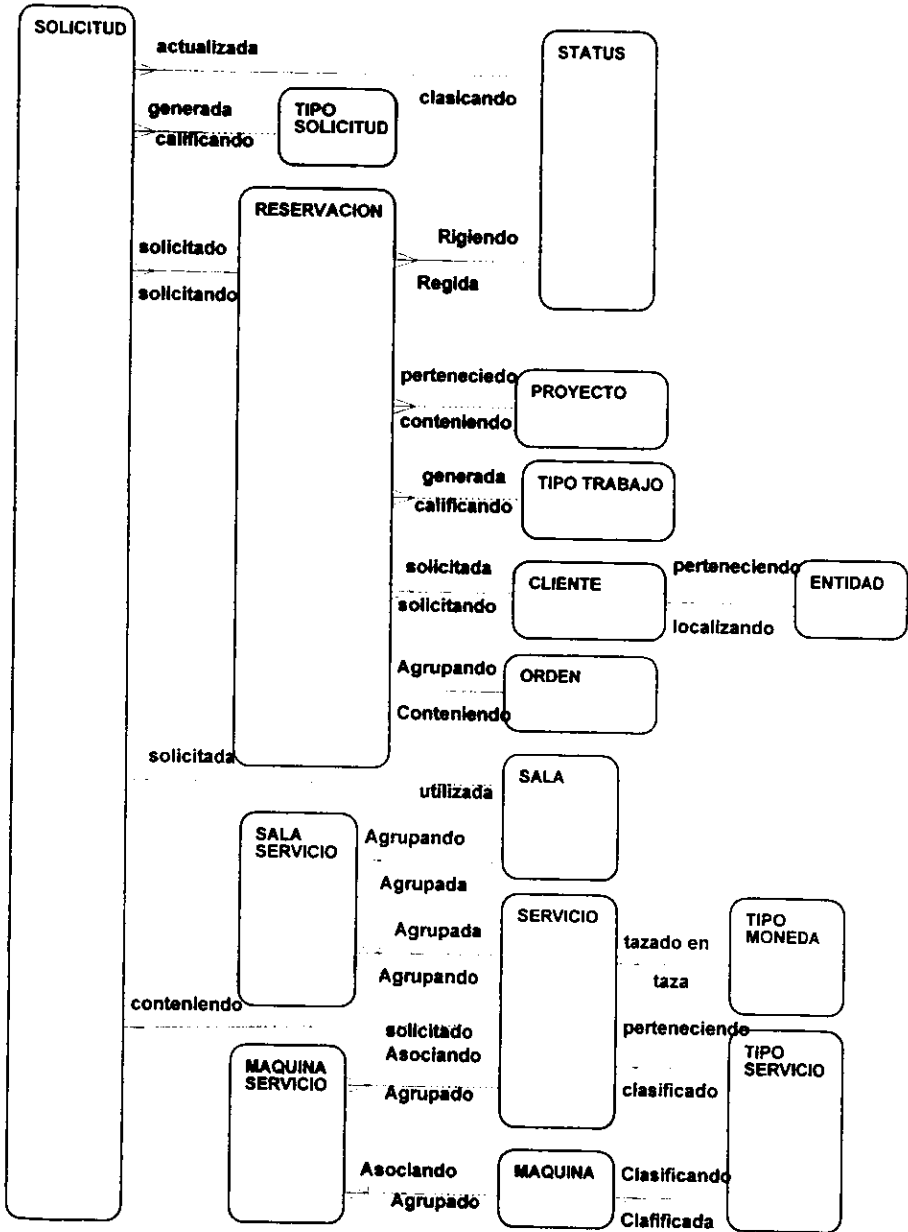


FIG.2.3.3.C DIAGRAMA ENTIDAD RELACIÓN MÓDULO DE RESERVACIONES

08	04	09	DIRECCION GENERAL	D
08	04	10	ADMINISTRACION	D
08	04	11	VENTAS	D
08	05		MANO DE OBRA DIRECTA	A
08	05	02	MASTER	D
08	05	03	RANK	D
08	05	04	C.M.X.	D
08	05	05	PAINT BOX	D
08	05	06	PRODUCCION	D
08	05	07	MATERIALES	D
08	05	08	OPERACIONES	D
08	05	09	DIRECCION GENERAL	D
08	05	10	ADMINISTRACION	D
08	05	11	VENTAS	D
09			INGRESOS	A
09	01		INGRESOS PROPIOS	A
09	01	02	MASTER	D
09	01	03	RANK	D
09	01	04	C.M.X.	D
09	01	05	PAINT BOX	D
09	01	06	PRODUCCION	D
09	01	07	MATERIALES	D
09	01	08	C.M.X. 2	D
09	01	09	DIRECCION GENERAL	D
09	01	10	ADMINISTRACION	D
09	01	11	VENTAS	D
09	02		OTROS INGRESOS	A
09	02	02	MASTER	D

FIG. 2.3.3.E ALGUNAS CUENTAS

Compañía S.A. DE C.V.

Camino a San Pablo No. 255
Del. Xoxhimilco, Mexico, D.F.
(915) 7650519 - Fax 915) 7251110

FACTURA

Nº DE FACTURA: 1001
FECHA: 22 de noviembre de 1997

Recibi de ::
TELEVISA, S.A. DE C.V.

Dirección de entrega:
Camino a San Pablo No. 255

REPRESENTANTE	Nº DE PEDIDO	FECHA DE ENVIO	MODO DE ENVIO	MODO DE PAGO
1	12	23-11-97		CHQ

CANTIDAD	DESCRIPCIÓN	PRECIO/ UNIDAD	SUMA
5	"EL ROBO IMPOSIBLE" HORAS DE CALIFICACION		532.50
4	HORAS DE TRANSFER		2,430.00
1	COPIA DE D3 A %		750.00
4	"FALLASTES CORAZON" HORAS DE CALIFICACION		710.00
4	HORAS DE TRANSFER		2,430.00
2	COPIA DE D3 A %		1,000.00
SUBTOTAL			0
OTROS			0
ENTREGA			0
SUMA TOTAL			7,852.50

Cheque nominal de Banca Serfin. S.A. No 9762 Total 7,852.50

Si tiene alguna pregunta acerca de esta factura, póngase en contacto con: Blanca Gil C., Nº Tel.6023501

¡LE AGRADECEMOS SU PEDIDO!

FIG. 2.3.3.F EJEMPLO DE UNA FACTURA

Clientes		Desglose de saldos vencidos en días.				Clasificación
Documento	Fecha Venc	Al corriente	1-30	31-60	61-90	91 o mas
1 LABORATORIOS BOEHRING/COLOMBIA						
6506	29-Ago-96					438.93
		439	0.00	0.00	0.00	0.00
						438.93
8 ASOCIACION PROGRAMA COMPARTIDOS IAP						
6225	15-Jul-96					0.00
6860	5-Nov-96					15,157.00
		22,736.72	0.00	7,578.91	0.00	0.00
						15,157.00
26 ALUCINEMA, S.A DE C.V.						
2036	28-Mar-94					33,956.00
2080	30-Abr-94					3,883.00
2467	30-Jun-94					10,219.00
2474	30-Jun-94					3,080.00
2476	30-Jun-94					396.92
		51,534.92	0.00	0.00	0.00	0.00
						51,534.92
35 ALBERTO LASTRA Y ASOCIADOS, S.A. DE C.V.						
6900	8-Nov-96		1481.20			0.00
6902	8-Nov-96		115.00			0.00
6903	8-Nov-96		218.50			0.00
		1,814.70	0.00	1,814.70	0.00	0.00
						0.00
51 CHINAMPA PRODUCCIONES, S.A. DE C.V.						
2845	17-Oct-95					629.28
		629.28	0.00	0.00	0.00	0.00
						629.28

FIG. 2.3.3.G REPORTE DE ANTIGÜEDAD DE SALDO.

2.4. ESTRATEGIAS DE SOLUCIÓN.

Después de haber marcado las diferentes necesidades funcionales que el cliente tiene en la actualidad, podemos entonces marcar cada uno de los aspectos técnicos en software y en hardware que la solución final provea.

Los requerimientos se enlistan a continuación:

- Manejar sólo una base de datos compartida y actualizada.
- Poder tener la información en cada uno de los puntos de la organización actualizada y rápidamente.
- El sistema debe ser amigable y fácil de utilizar.
- Debe contar con una estructura de seguridad.
- Puede ser utilizado en cualquier PC o terminal de la empresa y en cualquier momento.
- No debe ser superior en precio, que el uso que el sistema tendrá en operación.
- Debe aprovechar al máximo el equipo que actualmente se encuentra instalado.

De lo anterior, se proponen cuatro opciones de solución, cada una resuelve los puntos citados. Los diagramas se muestran en las figuras de la 2.1.A a la E, y muestran los siguientes arreglos:

1. Arreglo Multiproceso: Una máquina central con una BD común, una red y terminales tontas que acceden los programas y la información de la máquina central.
2. El mismo arreglo pero sin utilizar una BD expofeso, sino un conjunto de archivos y una serie de programas de tercera generación.
3. Módulos individuales del sistema en cada PC, con un proceso manual o semiautomático para asegurar el uso de la misma información centralizada en otra PC.

4. Un arreglo Cliente/Servidor, donde el servidor puede estar dedicado o puede ser otra computadora asignada a un módulo del sistema. Las PC realizarán la parte del Front-End.

En los subtemas siguientes, se detallarán las ventajas y desventajas de cada uno y la elección finalmente de la solución a realizar.

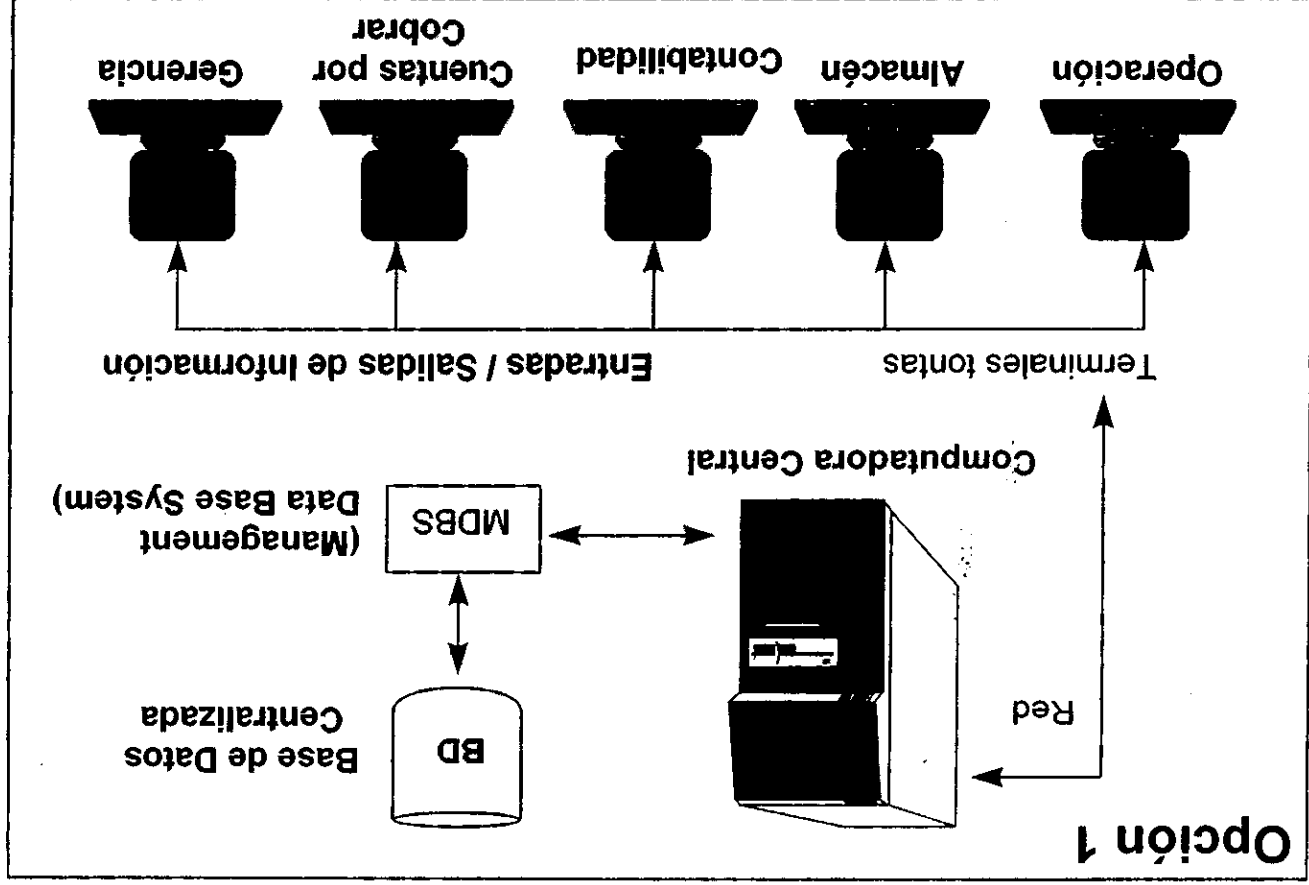


Fig. 2.1.A. Arreglo Multiproceso.

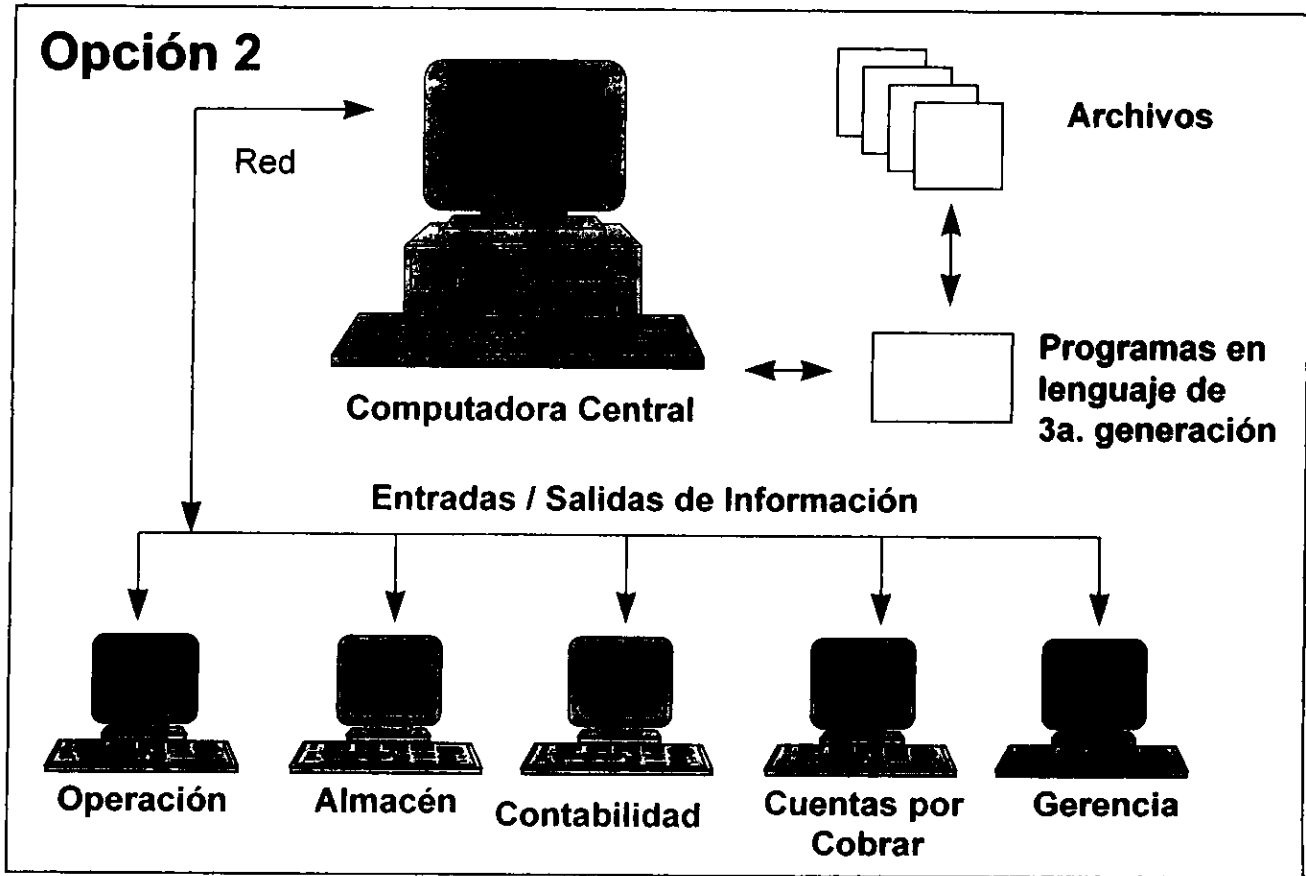


Fig. 2.1.B. Arreglo Multiproceso usando archivos.

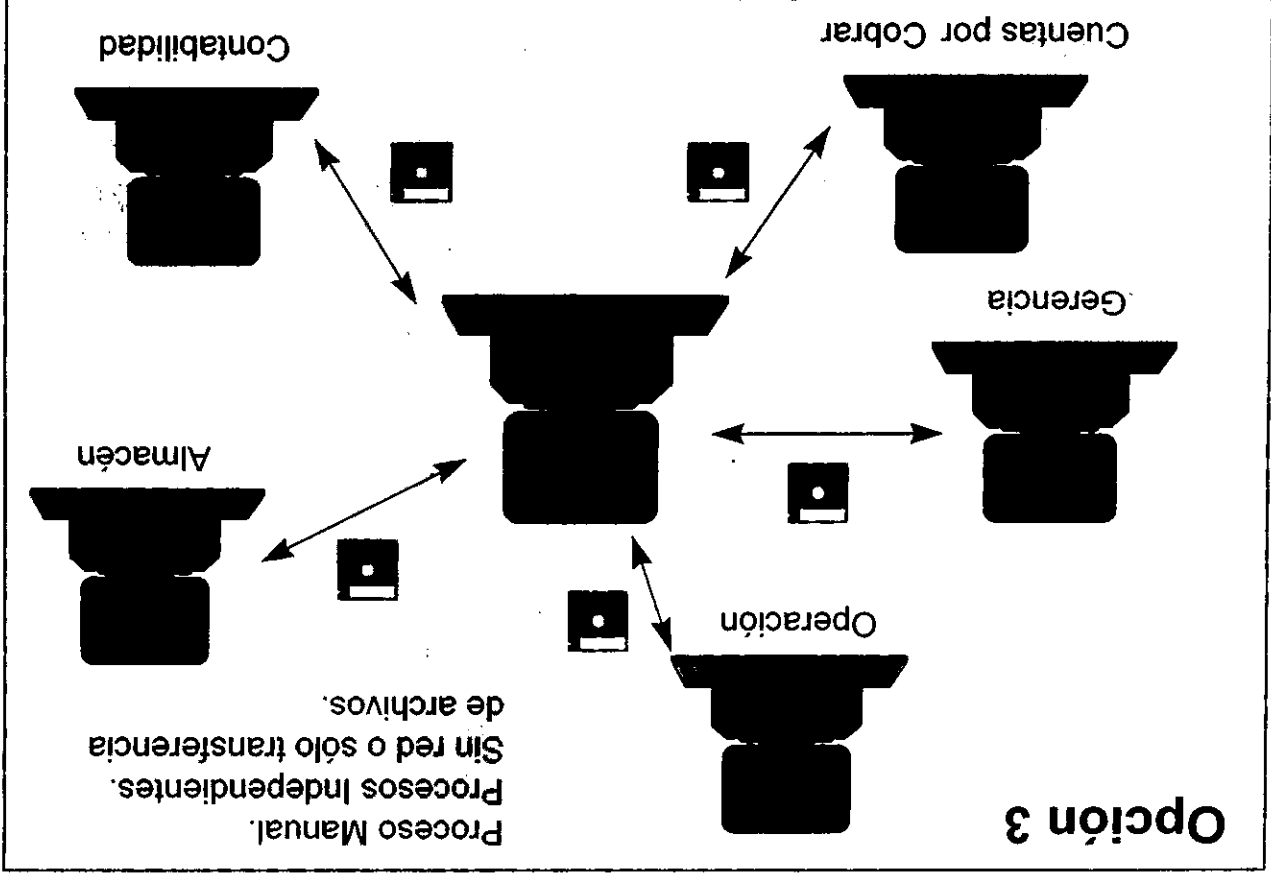


Fig. 2.1.C. Módulos Individuales.

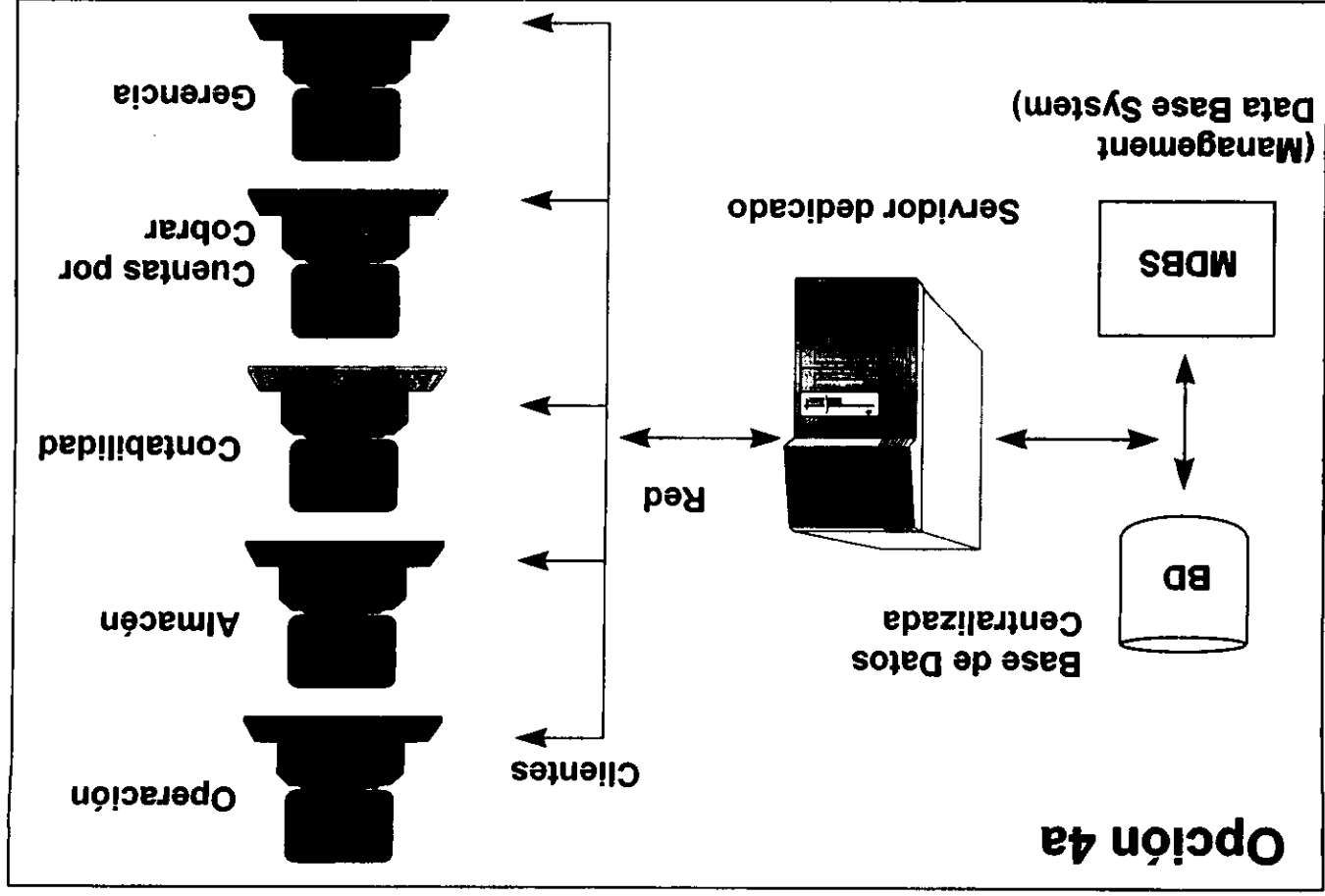


Fig. 2.1.D. Arreglo Cliente/Servidor con servidor dedicado.

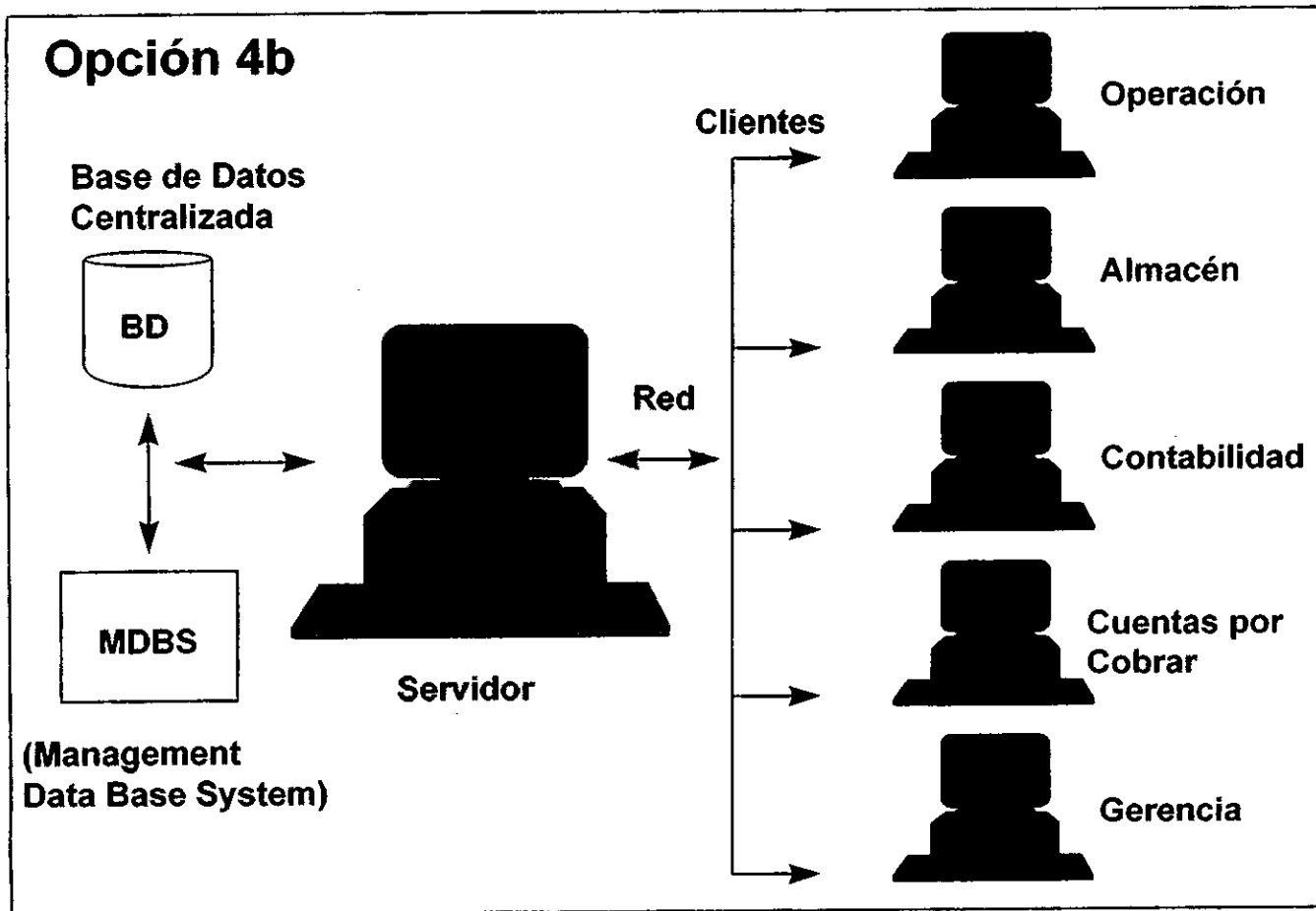


Fig. 2.1.E. Arreglo Cliente/Servidor con PC de servidor.

2.5. OPCIONES DE SOLUCIÓN.

En esta época sobresalen los sistemas para el manejo de base de datos (DBMS), los archiveros de la edad electrónica. Si se busca entre el papeleo de cualquier organización en el mundo, seguramente se encontrará una gran carga de documentos solicitando el manejo de una base de datos: la lista de clientes para la sala de alquiler de videos, el inventario de CD's en una tienda de música o los registros de personal de una empresa. De igual manera sobresalen los paquetes de computación que proporcionan las herramientas necesarias para el desarrollo de sistemas más amigables al usuario mediante un ambiente gráfico.

En este punto, se analizan las alternativas de arquitectura propuestas en el punto anterior a fin de evaluar cada una de las ventajas y desventajas.

Opción 1. Arreglo Multiusuario.

Desventajas	Ventajas
<ul style="list-style-type: none">• Orientado exclusivamente (ó mayormente) a procesamiento batch (en lotes).	<ul style="list-style-type: none">• Facilidad de administración: los componentes y periféricos están físicamente juntos.
<ul style="list-style-type: none">• Alto costo de mantenimiento.	<ul style="list-style-type: none">• La lógica del negocio, interface de usuario, lógica de impresión están en un mismo programa: es un sólo punto geográfico de mantenimiento.
<ul style="list-style-type: none">• Dificultad para compartir información con otros sistemas, incluso del mismo tipo y dentro del mismo equipo.	
<ul style="list-style-type: none">• Alta duplicidad de datos debido al tipo de archivos utilizados para el almacenamiento de información.	
<ul style="list-style-type: none">• Esquemas débiles de seguridad en términos transaccionales.	
<ul style="list-style-type: none">• Interface de usuario poco amigable: modo carácter.	
<ul style="list-style-type: none">• Esquemas propietarios: pocos proveedores.	
<ul style="list-style-type: none">• Pocas aplicaciones en el mercado para estas arquitecturas.	
<ul style="list-style-type: none">• Para empresas grandes.	

Opción 2. Red de PC's Compartiendo Archivos.

Desventajas	Ventajas
• Duplicidad de información.	• Ofrece una interface de usuario amigable: estándar con respecto a otras aplicaciones de oficina (procesador de palabras, hojas de cálculo, presentadores, etc).
• Cada equipo debe administrarse como sistema independiente.	• Capacidad de compartir recursos de impresión.
• Representa un costo alto de mantenimiento y administración.	
• Altamente demandante en la sincronización de información.	
• Poca capacidad de compartir información con equipos de diferentes arquitecturas (DOS/Windows/Mac/Next, etc.)	
• Involucra el costo de una red: concentradores, ruteadores, servidor de red, etc.	

Opción 3. PC's separadas con transferencia de información manual.

Desventajas	Ventajas
• Duplicidad de información.	• Ofrece una interface de usuario amigable: estándar con respecto a otras aplicaciones de oficina (procesador de palabras, hojas de cálculo, presentadores, etc).
• Cada equipo debe administrarse como sistema independiente.	• No existe un costo de instalación y mantenimiento de red.
• Representa un costo alto de mantenimiento y administración.	
• Altamente demandante en la sincronización de información.	
• Alto grado de error humano.	
• No se cuenta con la información integrada y a tiempo.	
• No hay posibilidad de compartir recursos: almacenamiento, lectura, impresión, etc.	
• Posibilidad de manejar diferentes versiones de herramientas, provocando dificultad en el proceso de compartir información.	

Opción 4. Cliente/Servidor.

Desventajas	Ventajas
<ul style="list-style-type: none"> • Múltiples proveedores: múltiples puntos de solución. 	<ul style="list-style-type: none"> • Ofrece una interface de usuario amigable: estándar con respecto a otras aplicaciones de oficina (procesador de palabras, hojas de cálculo, presentadores, etc).
<ul style="list-style-type: none"> • Múltiples puntos de administración. 	<ul style="list-style-type: none"> • Capacidad de mezclar componentes de cualquier tipo: software y hardware.
	<ul style="list-style-type: none"> • Alta distribución geográfica.
	<ul style="list-style-type: none"> • Ofrece la infraestructura para aplicaciones grupales: correo electrónico, procesadores de palabras, agendas electrónicas, etc.
	<ul style="list-style-type: none"> • Al usar un manejador de datos compartido se logra la sincronía inmediata de datos.
	<ul style="list-style-type: none"> • La información puede ser accesada desde cualquier punto.
	<ul style="list-style-type: none"> • Hay una gran diversidad de soluciones para estas arquitecturas.

OPCIÓN ELEGIDA.

Una vez presentados los puntos más relevantes como ventajas y desventajas de cada solución, es indiscutible que la más apropiada es la arquitectura Cliente/Servidor y sobre la que adicionalmente se agregan los siguientes puntos que refuerzan esta decisión:

1. Es una tecnología abierta y actualmente en uso en la empresa.
2. La empresa cuenta previamente con las herramientas que soportan esta arquitectura: manejador Access, Visual Basic, servidor de red, estaciones de trabajo Windows 95.
3. Es la opción más barata para la empresa.
4. Es una solución fácilmente expandible.
5. Es posible implantarla en otras oficinas regionales de la empresa, sin cambio alguno.
6. Existen aplicaciones previamente desarrolladas en esta arquitectura.

2.6 DIAGRAMA DE FUNCIONALIDAD

2.6.1 Arquitectura lógica propuesta.

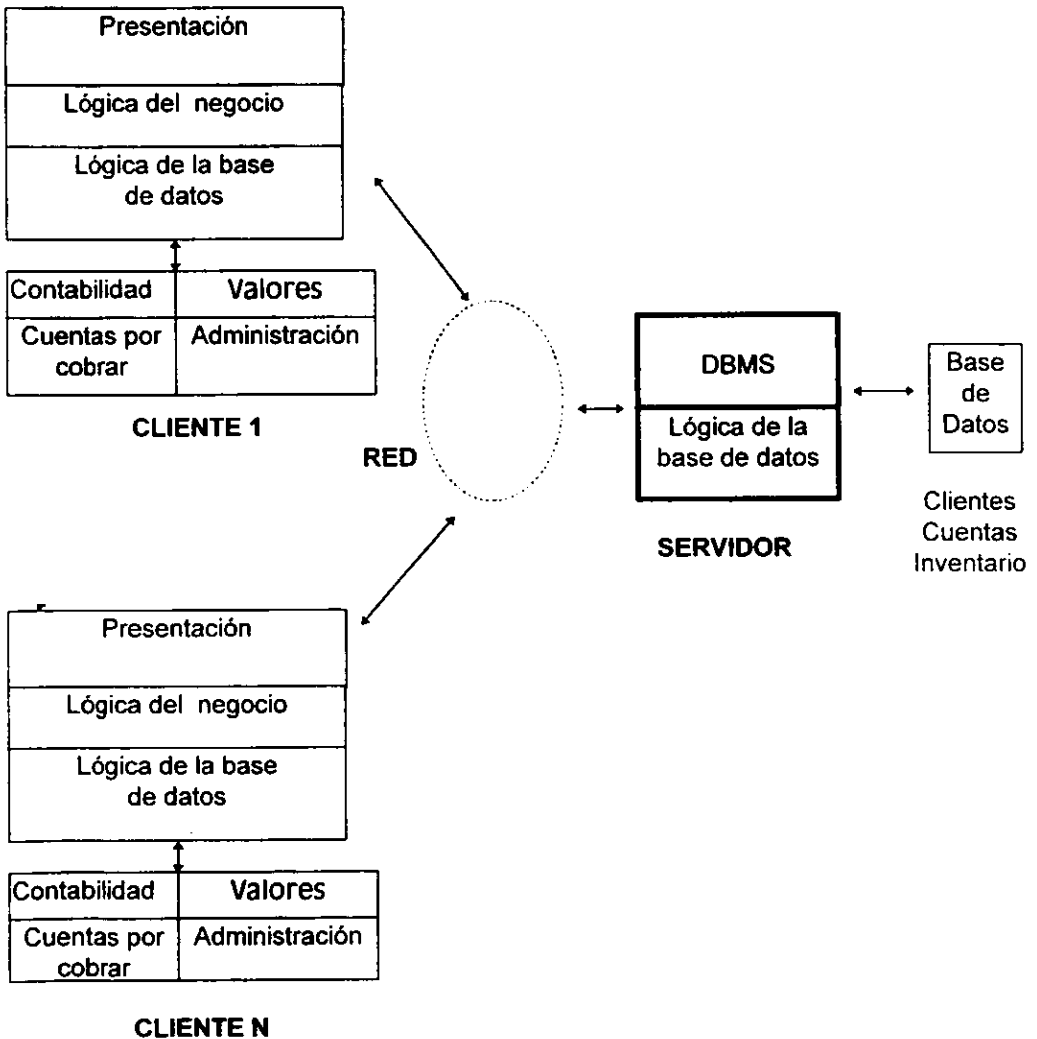


FIG. 2.6.1.A ARQUITECTURA LÓGICA PROPUESTA.

Como se ve en la fig. 2.6.1.A se trata de distribuir lo mejor posible el proceso entre el cliente y el servidor. Sin embargo el cliente es quien tiene la mayoría de las funciones, entre estas funciones tenemos:

- Presentación
- Validación de entrada de información
- Lógica del negocio.
- Parte de la lógica de la base de datos.

El servidor únicamente mantendrá al administrador de la base de datos (DBMS) así como una parte de la lógica de la base de datos.

2.6.2 Arquitectura física propuesta.

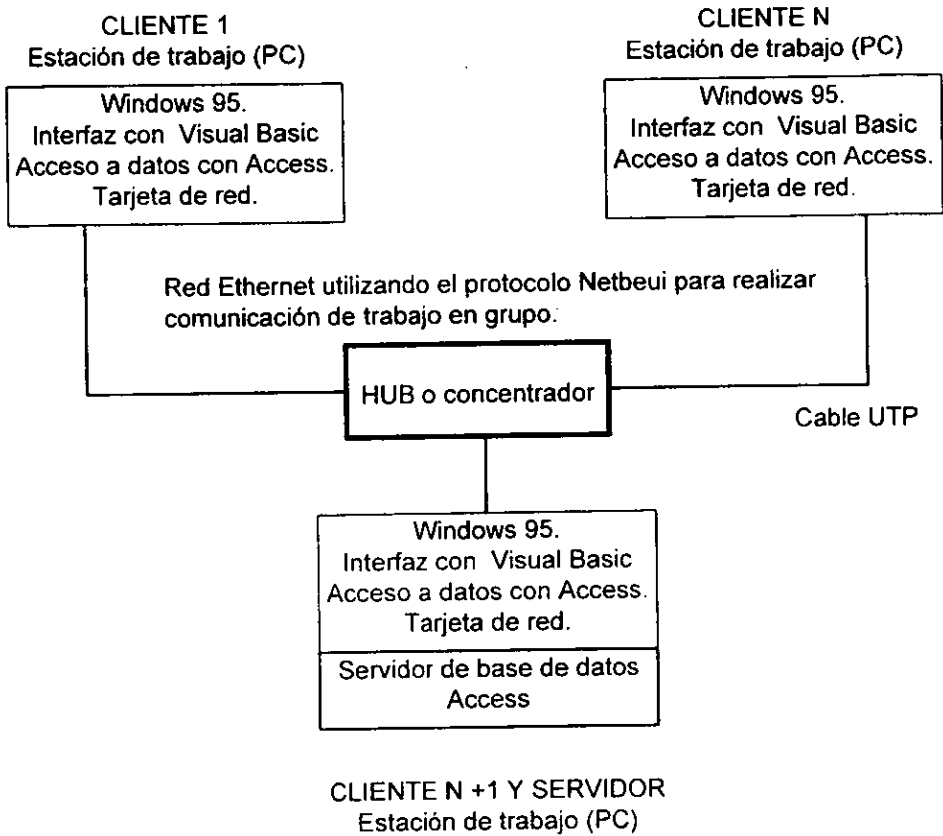


FIG 2.6.2.A ARQUITECTURA FÍSICA PROPUESTA.

En virtud de que el sistema no maneja grandes cantidades de información, ni realiza muchos accesos a discos. Estamos proponiendo que una estación de trabajo pueda realizar tanto las funciones cliente (interfaz gráfica) como las de un servidor (administrador de la base de datos).

El protocolo que se usará para comunicar las estaciones será el que maneja Windows 95 como estándar, es decir Windows para trabajo en grupo.

2.6.3 Requerimientos mínimos del sistema.

Requerimientos mínimos para el proceso de desarrollo.

Se necesitan de tres estaciones de trabajo conectadas en red, ya que son tres los recursos que se emplearán en el desarrollo del sistema.

Cada estación de trabajo necesita de tres componentes de software principalmente para poder establecer ambiente de desarrollo:

- Sistema Operativo Windows 95.
- Manejador de base de datos Access de Microsoft.
- Visual Basic de Microsoft.

Estos componentes necesitan los siguientes requerimientos :

- PC con procesador 486DX/25 o mayor (se recomienda pentium)
- 4 MB de memoria disponible (se recomiendan 8 MB).
- Monitor VGA o mayor.
- Mouse Microsoft o compatible pointing device.

Además de un espacio en disco para cada uno.

- Windows 95 requiere mínimo 110 MB
- Access requiere 13 MB de espacio disponible en disco
- La versión Professional de Visual Basic requiera de 20 MB para la instalación mínima y 70 MB para la versión completa.

En cuanto al equipo en forma general se necesitará el siguiente :

- Tres estaciones de trabajo con las siguientes características:
 - PC con procesador 486DX/25 o mayor (se recomienda pentium)
 - 4 MB de memoria disponible (se recomiendan 8 MB).
 - Monitor VGA o mayor.
 - Mouse Microsoft o compatible pointing device.
 - 250 MB de espacio disponible en disco.
 - Tarjeta de red Ethernet a 10 Mb/s
- Concentrador o Hub para conectar las estaciones en red.
- Cables tipo UTP
- Impresora para obtener listados.

Requerimientos mínimos para la operación

El número de estaciones mínimo que se requiere para que funcione el sistema de manera adecuada es de una por área (las áreas son contabilidad, cuentas por cobrar, Valores), es decir tres estaciones. Los requerimientos para estas estaciones son menores a las de desarrollo, ya que estas no requerirán de ambiente de desarrollo.

Para que las estaciones pueden operar adecuadamente se requiere el siguiente software :

- Windows 95 que requiere 110 MB
- Manejador de base de datos Access de Microsoft. que requiere 13 MB
- El sistema integral de contabilidad que requiere 10 MB
- Los datos que requiere 15 MB, únicamente para la estación que opere como servidor de la base de datos.

En cuanto al equipo en forma general se necesitará el siguiente :

- Tres estaciones de trabajo con las siguientes características:
 - PC con procesador 386DX/25 o mayor (se recomienda 486 o mayor)
 - 4 MB de memoria disponible (se recomiendan 8 MB).
 - Monitor VGA o mayor.
 - Mouse Microsoft o compatible pointing device.
 - 150 MB de espacio disponible en disco.
 - Tarjeta de red Ethernet a 10 Mb/s
- Concentrador o Hub para conectar las estaciones en red.
- Cableado estructurado con cable UTP.
- Tres Impresora para obtener reportes en cada área

**CAPITULO 3
DISEÑO E IMPLEMENTACIÓN DEL
SISTEMA**

3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3.1 ESPECIFICACIONES

Algunas de las especificaciones del sistema son básicamente los requerimientos de usuario que nos hizo saber desde un inicio, aunque mas aterrizados y formales.

Para poder llegar a las especificaciones del sistema se analizaron la factibilidad de varios puntos involucrados:

- Situación actual de la empresa
- Requerimientos iniciales del usuario
- Arquitectura física del sistema.
- Tecnología de software, hardware y redes actual.
- Requerimientos mínimos de hardware y software para la operación del sistema

En este análisis se involucro en varias etapas al usuario para que nos proporcionará un mejor detalle de sus requerimientos.

Analizados los puntos anteriores se llego a un acuerdo con el usuario para definir los alcances de las especificaciones de una primera versión.

Especificaciones generales de equipo

- Arquitectura física será cliente servidor de acuerdo a la propuesta que se mostró en el capítulo 2, fig 2.6.2.A
- **HARDWARE**
 - Dos estaciones de Trabajo PC 486 a 66 Mhz mínimo con disco duro de 150 MB.

- Una PC 486 a 66 Mhz mínimo con disco duro de 200 MB para que opere como estación de trabajo y servidor de base de datos..
- Una impresora mínimo para obtener reportes.

- SOFTWARE
 - Sistema operativo Windows 95.
 - Manejador de base de datos Access 95
 - Sistema integral de contabilidad.

- REDES
 - Red Ethernet a 10 MB/seg.
 - Cableado estructural
 - Cable UTP
 - Tarjetas de red Ethernet
 - Concentrador o Hub
 - Protocolo TCP/IP integrado en Windows 95.
 - Servicio de red de Microsoft para compartir archivos, directorios e impresoras integrado en Windows 95.

Especificaciones generales del sistema integral de contabilidad de acuerdo a requerimientos del usuario.

- Arquitectura lógica será cliente servidor de acuerdo a la propuesta que se mostró en el capítulo 2, fig 2.6.1.A
- El sistema contará con dos módulos :
 - Módulo de contabilidad.
 - Módulo de cuentas por cobrar.

- Una sola base de datos compartida entre todos los usuarios y módulos, para satisfacer los siguientes requerimientos:
 - Información única
 - Eliminación de redundancia de datos.
 - Accesible desde cualquier área
 - Información a tiempo.
 - Eliminación de recaptura de datos entre las diferentes áreas involucradas.

- El sistema sea amigable y fácil de usar.
- Que realice validación a nivel campo para evitar errores de captura.
- Contemplar seguridad de acuerdo a :
 - Grupos de usuarios manejando los siguientes niveles:
 - Superusuario
 - Usuario normal de contabilidad
 - Usuario de consulta de contabilidad.
 - Superusuario de crédito
 - Valores

 - Por área de trabajo :
 - Contabilidad
 - Crédito y cobranza
 - Registro de valores
 - Cuentas por cobrar
 - Facturación.

De acuerdo al grupo y al área asignada es lo que podrá realizar un usuario. El Superusuario podrá realizar todo.

Especificaciones del modulo de Contabilidad.

- Debe manejar los siguientes catálogos
- Catálogo de cuentas

Catálogo de Usuarios			
ARC	PATO DONALD	CONTABILIDAD	SUPER USUARIO DE CONTA
COAF	PEPITO GOMEZ	CONTABILIDAD	USUARIO NORMAL DE CONT
COGVC	RATON MIGUELITO	CREDITO Y COBRANZA	SUPER USUARIO DE CREDIT

- Catálogo de clasificación de cuentas

Catálogo de Cuenta Clasificación		
1	1	ACTIVO CIRCULANTE.
2	1	ACTIVO FIJO.
3	1	DEPRECIACIONES.
4	1	DIFERIDO.
5	1	PASIVO A CORTO PLAZO.

- Catálogo de cuentas de mayor

Catálogo de Cuenta de Mayor			
1	1	1	CAJA.
1	2	1	FONDO FIJO.
1	3	1	BANCOS.
1	4	1	INVERSIONES EN VALORES.
1	5	1	CLIENTES.
1	6	1	DEUDORES DIVERSOS.
1	7	1	ANTICIPO A PROVEEDORES.

- Centros de Costos

Catálogo de Centro de Costos				
Cta. General	Cta. Mayor	Centro Costos	SubCuenta	Descripción
1	1	1	2	MATRIZ.
1	2	1	2	MATRIZ.
1	3	1	1	MATRIZ.
1	4	1	1	MATRIZ.
1	5	1	1	MATRIZ.
1	6	1	1	MATRIZ.

- Catálogo de subcuentas

Catálogo de SubCuenta					
Cta. General	Cta. Mayor	Centro Costos	SubCuenta	SubSubCuenta	Descripción
2	3	1	1		EQ.DE TRANSPORTE VALOR Cuenta Detalle
2	3	1	2		REVALUACION DE EQ. DE T Cuenta Detalle
2	4	1	1		EQ.DE COMPUTO A VALOR Cuenta Detalle
2	4	1	2		REVALUACION EQ. DE COM Cuenta Detalle
2	5	1	1		HERRAM.Y ACCES.DE TRAE Cuenta Detalle
2	5	1	2		REVALUACION HERRAM.Y C Cuenta Detalle

- Catálogo de subsubcuentas

Catálogo de SubSubCuenta					
Cta. General	Cta. Mayor	Centro Costos	SubCuenta	SubSubCuenta	Descripción
1	3	1	1	1	CTA.- 400719601-7.
1	4	1	1	1	CTA.- 602659612-0.
1	6	1	1	1	JUAN PEREZ
1	6	1	1	2	EDGAR GARCIA ADAME

- Catálogo de cuentas de mayor

Catálogo de Cuenta de Mayor			
1	1	1	CAJA.
1	2	1	FONDO FIJO.
1	3	1	BANCOS.
1	4	1	INVERSIONES EN VALORES.
1	5	1	CLIENTES.
1	6	1	DEUDORES DIVERSOS.
1	7	1	ANTICIPO A PROVEEDORES.

- Estandarización de cuentas
- Manejo de cinco niveles de cuentas.
- Control de cierres contables mensuales y anuales
- Manejo tres tipos de pólizas
 - Póliza automática. Para hacer contabilidad automática al realizar movimientos desde cualquier área.
 - Póliza manual

• Póliza modelo

Comprobante Póliza Modelo			
Núm.	Fecha	Pl. Día	Detalle
02/01/1997	4	Pl. Día	Factura Serie: A no. 2
02/01/1997	5	Pl. Día	Materiales de la Factura Serie: A no. 2
02/01/1997	7	Pl. Día	Materiales de la Factura Serie: A no. 3

Núm.	Fecha	Pl. Día	Detalle	Debe	Haber			
11	9	1	2	0	0	Detalle de la Factura Serie: A No. 3	\$0.00	\$700.
12	9	1	2	0	0	Detalle de la Factura Serie: A No. 3	\$0.00	\$840.
13	9	1	2	0	0	Detalle de la Factura Serie: A No. 3	\$0.00	\$710.
14	9	1	2	0	0	Detalle de la Factura Serie: A No. 3	\$0.00	\$1,000.
							\$22,959.75	\$22,959.75

• Estados Financieros

Cuenta		Fecha																													
02-Costo de Devoluciones		23/06/97																													
Dijo		Cuenta de Copiado																													
<table border="1"> <thead> <tr> <th>Núm.</th> <th>Fecha</th> <th>Pl. Día</th> <th>Detalle</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>MATRIZ</td> </tr> <tr> <td>1</td> <td>2</td> <td>1</td> <td>0</td> <td>0</td> <td>MATRIZ</td> </tr> <tr> <td>1</td> <td>3</td> <td>1</td> <td>1</td> <td>1</td> <td>CTA- 400719601-7.</td> </tr> <tr> <td>1</td> <td>5</td> <td>1</td> <td>1</td> <td>0</td> <td>DIVERSOS.</td> </tr> </tbody> </table>				Núm.	Fecha	Pl. Día	Detalle	1	1	1	0	0	MATRIZ	1	2	1	0	0	MATRIZ	1	3	1	1	1	CTA- 400719601-7.	1	5	1	1	0	DIVERSOS.
Núm.	Fecha	Pl. Día	Detalle																												
1	1	1	0	0	MATRIZ																										
1	2	1	0	0	MATRIZ																										
1	3	1	1	1	CTA- 400719601-7.																										
1	5	1	1	0	DIVERSOS.																										
<table border="1"> <thead> <tr> <th>Núm.</th> <th>Fecha</th> <th>Pl. Día</th> <th>Detalle</th> <th>Cuenta</th> <th>Fecha</th> <th>Presupuesto</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>1</td> <td>0</td> <td>0</td> <td>Cuenta de Copiado</td> <td>02-Costo de Devoluciones 30/06/97</td> <td>\$0.00</td> </tr> <tr> <td>1</td> <td>4</td> <td>1</td> <td>1</td> <td>1</td> <td>Cuenta de Copiado</td> <td>02-Costo de Devoluciones 30/06/97</td> <td>\$0.00</td> </tr> </tbody> </table>				Núm.	Fecha	Pl. Día	Detalle	Cuenta	Fecha	Presupuesto	1	2	1	0	0	Cuenta de Copiado	02-Costo de Devoluciones 30/06/97	\$0.00	1	4	1	1	1	Cuenta de Copiado	02-Costo de Devoluciones 30/06/97	\$0.00					
Núm.	Fecha	Pl. Día	Detalle	Cuenta	Fecha	Presupuesto																									
1	2	1	0	0	Cuenta de Copiado	02-Costo de Devoluciones 30/06/97	\$0.00																								
1	4	1	1	1	Cuenta de Copiado	02-Costo de Devoluciones 30/06/97	\$0.00																								
\$0.00																															

- Balance general

Activo Fijo						Fecha
1	2	0	0	0	FONDO FIJO.	
1	3	0	0	0	BANCOS.	
1	4	0	0	0	INVERSIONES EN VALORES.	
1	6	0	0	0	DEUDORES DIVERSOS.	
1	1	0	0	0	Activo Fijo	30/06/97
1	5	0	0	0	Activo Fijo	30/06/97

Activo Fijo						Fecha
1	2	0	0	0	FONDO FIJO.	
1	3	0	0	0	BANCOS.	
1	4	0	0	0	INVERSIONES EN VALORES.	
1	5	0	0	0	CUENTAS.	
1	6	0	0	0	DEUDORES DIVERSOS.	
1	1	0	0	0	Activo Fijo	30/06/97
1	5	0	0	0	Activo Fijo	30/06/97

- Emisión de los siguientes reportes
 - Auxiliares por nivel de cuentas
 - Diario general
 - Balanza de comprobación
 - Balanza de comprobación por cuenta de mayor
 - Auxiliares por cuenta
 - Cuentas con saldo
 - Estados financieros
 - Balance general.

Especificaciones del modulo de Cuentas por cobrar

Los catálogos que puede actualizar son:

- Catálogo de clientes

Catálogo de clientes				
RFC	Razón Social	Nombre	Dirección	Colonia
3	PATITO, S.C.	A E	VITO ALESIO ROBLES	
ACR-857664-4A	SOCIEDAD ANONIMA S. A.	Ácrata	Av. Montevideo No. 16 Line	
ADM 740816 94I	DEPORTES PATÁ BENDITA	ADEMEX	2ª CERRADA DE PIRA TE	
SAMA 630129 C	FABRICA LA CALABERA	ANA LILIA	PASEO NUEVO 43-301 PAS	

- Todos sus campos serán:
 - RFC
 - Razón social
 - nombre
 - Dirección
 - Colonia
 - Delegación

- Estado
 - Código postal
 - Teléfono
 - Fax
 - Clasificación
 - Día de pago
 - Limite de crédito
 - Días de crédito
 - Descuento
 - Referencia
 - Día de revisión
 - Saldo anticipado
 - Saldo pendiente
 - Saldo Real
-
- Catálogo de bancos

Catálogo de Bancos			
Nombre	Estado	Referencia	Saldo
Suiza	No lo sé	Quien sabe	56465465
BANAMEX	CONOCIDA	PEPITO GOMEZ	7251541
BANCOMER	LA MISMA	PATO DONALD	6667788

- Historia sobre comportamiento de clientes.
- Crédito a clientes según su comportamiento.
- Control de pagos totales, parciales o anticipados
- Todo lo introducido se refleje de manera contable automáticamente con ayuda de pólizas automáticas y modelo.
- Poder realizar diferentes tipos de ajustes:

- Cancelación de facturas

The screenshot displays a software interface for invoice management. At the top right, the date '23/06/97' is shown. The main header area contains the following information:

- Company:** PATITO, S.C.
- Invoice Number:** 3
- Total Amount:** \$8,201.16
- Client Name:** VITO ALESIO ROBLES

Below this information is a table with columns: 'Gene', 'Num', 'Fecha', 'Subtotal', 'IVA', and 'Total'. The table contains four rows of invoice items:

Gene	Num	Fecha	Subtotal	IVA	Total
A	254	20/02/97	\$827.16	\$124.07	\$951.23
A	427	30/05/97	\$3,750.00	\$562.50	\$4,312.50
A	428	30/05/97	\$3,300.00	\$495.00	\$3,795.00
A	429	30/05/97	\$4,400.00	\$660.00	\$5,060.00

- Cheques devueltos
- Pagos no aplicados
- Manejo de diferentes descuentos sobre precios establecidos.
- Emisión de diferentes reportes
 - Estado de cuenta general
 - Pronostico de cobranza
 - Pronostico de revisión
 - Saldos vencidos
 - Antigüedad de saldos
 - Reporte de facturas

3.2. Diseño y construcción de la base de datos.

A continuación mostramos cada uno de los diagramas y la información detallada de los pasos seguidos en el desarrollo de la BD.

Los diagramas que se presentan son:

- Diagrama de Flujo de Datos (conceptual, nivel 1 y nivel 2).
- Diagrama Funcional (hasta llegar a las funciones elementales).
- Diagrama Entidad-Relación.
- Diagrama de Datos

Los reportes que se presentan son:

- Definición de Entidades y Atributos.
- Esquema físico de la Base de Datos.
- Definición de Funciones.

Las matrices que se presentan son:

- Matriz Entidades vs. Funciones.
- Matriz Atributos vs. Funciones.

Los diagramas están presentados en sus dos respectivos módulos:

- Submódulo de Contabilidad
- Submódulo de Cuentas por Cobrar

3.2.1. Diagrama de Flujo de Datos (DFD).

El DFD se muestra a continuación, el listado siguiente es el de los Almacenadores de datos originados del Diagrama de Flujo de Datos, las funciones se detallan en el Diagrama Jerárquico Funcional.

DATASTORE: D005	AREA
------------------------	-------------

RECEIVES DATA ITEMS FROM

Function F006	Actualizar Áreas de Trabajo	Via Dataflow: Área
---------------	-----------------------------	--------------------

PROVIDES DATA ITEMS TO

Function M007	Controlar Pólizas Modelo	Via Dataflow: Área
---------------	--------------------------	--------------------

DATASTORE: D013	BANCO
------------------------	--------------

RECEIVES DATA ITEMS FROM

Function FC02	Actualizar Bancos	Via Dataflow: Banco
---------------	-------------------	---------------------

PROVIDES DATA ITEMS TO

DATASTORE: D003	CLIENTE
------------------------	----------------

RECEIVES DATA ITEMS FROM

Function MC01	Controlar Catálogos	Via Dataflow: Cliente
---------------	---------------------	-----------------------

Function FC01	Actualizar Clientes	Via Dataflow: Cliente
---------------	---------------------	-----------------------

PROVIDES DATA ITEMS TO

Function M005	Controlar Cierres	Via Dataflow: Cliente
Function F013	Cerrar meses	Via Dataflow: Cliente
Function M001	Controlar Módulo de Contabilidad	Via Dataflow: Cliente
Function MC02	Controlar Pagos	Via Dataflow: Cliente
Function MC03	Controlar Ajustes de Pagos	Via Dataflow: Cliente
Function FC03	Capturar Pagos	Via Dataflow: Cliente
Function RC01	Reportar Cifras de Control	Via Dataflow: Cliente
Function FC04	Asignar Pagos Anticipados	Via Dataflow: Cliente
Function MC04	Controlar Consultas y Reportes	Via Dataflow: Cliente
Function FC06	Capturar Pago no Incluido	Via Dataflow: Cliente
Function MC05	Controlar Pagos mal Aplicados	Via Dataflow: Cliente
Function FC10	Consultar Facturas a Revisión	Via Dataflow: Cliente
Function FC09	Consultar Posible Cobranza	Via Dataflow: Cliente
Function RC03	Reportar Antigüedad de Saldos	Via Dataflow: Cliente
Function RC02	Reportar Estado de Cuenta	Via Dataflow: Cliente

DASTORE: D004

CUENTA

RECEIVES DATA ITEMS FROM

Function F003	Actualizar Centro de Costos	Via Dataflow: Centro costo
Function F001	Actualizar Clasificación de Cuenta	Via Dataflow: Clasificación Cta
Function F002	Actualizar Cuenta de Mayor	Via Dataflow: Cta Mayor
Function M001	Controlar Módulo de Contabilidad	Via Dataflow: Cuenta
Function F004	Actualizar Subcuenta	Via Dataflow: Subcuenta
Function F005	Actualizar Subsubcuenta	Via Dataflow: Subsubcuenta

PROVIDES DATA ITEMS TO

Function F013	Cerrar meses	Via Dataflow: Cuenta
Function M011	Generar balance general.	Via Dataflow: Cuenta
Function M010	Generar estados financieros.	Via Dataflow: Cuenta

DATASTORE: D001	FACTURA
------------------------	----------------

RECEIVES DATA ITEMS FROM**PROVIDES DATA ITEMS TO**

Function M003	Controlar Pólizas	Via Dataflow: Factura
Function F010	Capturar Pólizas Manuales	Via Dataflow: Factura
Function M001	Controlar Módulo de Contabilidad	Via Dataflow: Factura
Function MC00	Controlar Módulo de Cuentas por Cobrar	
		Via Dataflow: Factura
Function MC02	Controlar Pagos	Via Dataflow: Factura
Function MC04	Controlar Consultas y Reportes	Via Dataflow: Factura
Function FC03	Capturar Pagos	Via Dataflow: Factura
Function FC05	Cancelar Factura	Via Dataflow: Factura
Function FC06	Capturar Pago no Incluido	Via Dataflow: Factura
Function MC05	Controlar Pagos mal Aplicados	Via Dataflow: Factura
Function FC10	Consultar Facturas a Revisión	Via Dataflow: Factura
Function FC09	Consultar Posible Cobranza	Via Dataflow: Factura
Function RC03	Reportar Antigüedad de Saldos	Via Dataflow: Factura
Function RC02	Reportar Estado de Cuenta	Via Dataflow: Factura
Function RC04	Reportar Saldos Vencidos	Via Dataflow: Factura

DATASTORE: D009	GPO USUARIO
------------------------	--------------------

RECEIVES DATA ITEMS FROM

Function F014	Actualizar Grupos de Usuarios	Via Dataflow: Gpo Usuario
---------------	-------------------------------	---------------------------

PROVIDES DATA ITEMS TO

Function F015	Actualizar Usuarios	Via Dataflow: Gpo Usuario
---------------	---------------------	---------------------------

DATASTORE: D002

PAGO

RECEIVES DATA ITEMS FROM

Function MC00	Controlar Módulo de Cuentas por Cobrar	Via Dataflow: Pago
Function MC02	Controlar Pagos	Via Dataflow: Pago
Function MC03	Controlar Ajustes de Pagos	Via Dataflow: Pago
Function FC03	Capturar Pagos	Via Dataflow: Pago
Function FC04	Asignar Pagos Anticipados	Via Dataflow: Pago
Function FC06	Capturar Pago no Incluido	Via Dataflow: Pago
Function MC05	Controlar Pagos mal Aplicados	Via Dataflow: Pago

PROVIDES DATA ITEMS TO

Function M003	Controlar Pólizas	Via Dataflow: Pago
Function F010	Capturar Pólizas Manuales	Via Dataflow: Pago
Function M001	Controlar Módulo de Contabilidad	Via Dataflow: Pago
Function RC01	Reportar Cifras de Control	Via Dataflow: Pago
Function MC04	Controlar Consultas y Reportes	Via Dataflow: Pago
Function RC02	Reportar Estado de Cuenta	Via Dataflow: Pago
Function FC11	Consultar Pagos	Via Dataflow: Pago

DATASTORE: D011

PARAMETRO

RECEIVES DATA ITEMS FROM

Function F017	Actualizar parámetros de salas.	Via Dataflow: Parámetro
Function F016	Actualizar parámetros iniciales.	Via Dataflow: Parámetro

PROVIDES DATA ITEMS TO

DATASTORE: D007

POLIZA

RECEIVES DATA ITEMS FROM

Function F010	Capturar Pólizas Manuales	Via Dataflow: Póliza
Function F012	Ajustar Pólizas Manuales	Via Dataflow: Póliza
Function F011	Capturar Pólizas Modelo	Via Dataflow: Póliza
Function M001	Controlar Módulo de Contabilidad	Via Dataflow: Póliza

PROVIDES DATA ITEMS TO

Function R001	Reportar Auxiliares por Cuenta	Via Dataflow: Póliza
Function R002	Reportar Diario General	Via Dataflow: Póliza
Function R003	Reportar Balanza de Comprobación	Via Dataflow: Póliza
Function R004	Reportar Balanza de Comprobación por Cuenta de Mayor	Via Dataflow: Póliza
Function R005	Reportar Auxiliares de todas las Cuentas	Via Dataflow: Póliza
Function R006	Reportar Auxiliares por Nivel de Cuenta	Via Dataflow: Póliza
Function R009	Reportar cuentas con saldo.	Via Dataflow: Póliza

DATASTORE: D006

POLIZA MODELO

RECEIVES DATA ITEMS FROM

Function M007	Controlar Pólizas Modelo	Via Dataflow: Pól Modelo
---------------	--------------------------	--------------------------

PROVIDES DATA ITEMS TO

Function F011	Capturar Pólizas Modelo	Via Dataflow: Pól Modelo
---------------	-------------------------	--------------------------

DATASTORE: D012	SALA
------------------------	-------------

RECEIVES DATA ITEMS FROM

PROVIDES DATA ITEMS TO

Function F017	Actualizar parámetros de salas.	Via Dataflow: Sala
Function M008	Controlar parámetros generales.	Via Dataflow: Sala
Function M001	Controlar Módulo de Contabilidad	Via Dataflow: Sala

DATASTORE: D008	SALDO
------------------------	--------------

RECEIVES DATA ITEMS FROM

Function F013	Cerrar meses	Via Dataflow: Saldo
Function M001	Controlar Módulo de Contabilidad	Via Dataflow: Saldo

PROVIDES DATA ITEMS TO

Function M011	Generar balance general.	Via Dataflow: Saldo
Function M010	Generar estados financieros.	Via Dataflow: Saldo

DATASTORE: D010	USUARIO
------------------------	----------------

RECEIVES DATA ITEMS FROM

Function F015	Actualizar Usuarios	Via Dataflow: Usuario
---------------	---------------------	-----------------------

PROVIDES DATA ITEMS TO

A continuación presentamos los Diagramas de Flujo de Datos y una breve descripción de cada uno de sus elementos. Estos diagramas son el de contexto (nivel 0) el 1 y el 2. El diagrama de contexto nos señala que la función principal del Módulo de Contabilidad es recibir la información de la venta (Factura) y Cuentas x Cobrar (Pagos) y realizar los movimientos contables obligados (Pólizas) afectando los Saldos de Clientes y Cuentas. Para el módulo de Cuentas x Cobrar, es el de recibir la información de la venta (Factura) y realizar el cobro y seguimiento de cada uno de los Pagos. Las funciones principales son detalladas posteriormente en el diagrama funcional, aquí se mencionan como referencia al igual que los almacenadores de datos utilizados.

Controlar Sistema Integral de Casas Postproductoras. Función principal que controla la ejecución de todos los módulos que lo componen.

Controlar Módulo de Contabilidad. El módulo principal que controla la operación de las labores de la Contabilidad.

SALA. Cada uno de los lugares reservados y facturados por y para el cliente.

FACTURA. El documento legal donde indica los totales que el Cliente adeuda.

CLIENTE. Cada una de las personas físicas y morales que solicitan y facturan servicios de la Empresa.

PAGO. Cada uno de los abonos que el Cliente ha realizado para liquidar o anticipar su adeudo.

PÓLIZA. Documento contable donde se indica el encabezado del movimiento contable que se haya realizado.

SALDO. Total de abono y cargos que nuestras cuentas contables y clientes tengan al final del mes fiscal.

CUENTA. Cada uno de los rubros que la empresa utiliza para organizar los tipos de movimientos contables que se desarrollan en el trabajo diario.

Controlar catálogos. Módulo para dar mantenimiento a la información básica de la contabilidad como las cuentas contables (cuentas de mayor, centros de costos, subcuentas, etc.).

Controlar pólizas. Módulo para ingresar, consultar y reportar cada uno de los movimientos contables que se hayan realizado.

Controlar cierres. Módulo para traspasar, del mes contable anterior al actual, los totales de cargos y abonos de las cuentas contables y de los clientes.

Controlar parámetros generales. Módulo para actualizar los parámetros básicos de la contabilidad como porcentaje actual de IVA, cuentas contables especiales, etc.

Controlar usuarios. Módulo donde se actualiza la información de las claves y sus restricciones capaces de acceder los módulos contables.

Controlar reportes. Módulo donde se generan todos los reportes contables y auxiliares como apoyo al desarrollo de las actividades diarias.

Generar Estados Financieros y Balance. Módulo donde se estructuran los Estados Financieros y los Balances basándose en la información que durante lo largo del ejercicio contable se haya generado.

Controlar Módulo de Cuentas por Cobrar.

FACTURA. El documento legal donde indica los totales que el Cliente adeuda.

CLIENTE. Cada una de las personas físicas y morales que solicitan y facturan servicios de la Empresa y que tienen facturas pendientes de liquidar.

PAGO. Cada uno de los abonos que el Cliente ha realizado para liquidar o anticipar su adeudo.

Controlar Catálogos. Módulo para dar mantenimiento a la información básica de Cuentas por Cobrar como los Bancos y los Clientes.

Controlar Pagos. Módulo donde se reciben cada uno de los pagos ordinarios o anticipados del Cliente.

Controlar Ajustes de Pagos. Módulo donde se realizan las operaciones alternas en la cobranza como, cancelar facturas, cancelar pagos o ingresar pagos especiales.

Controlar Consultas y Reportes. Módulo donde se generan todos los reportes de la cobranza y la cartera vencida como apoyo al desarrollo de las actividades diarias.

Diagram Name : DFDCONTA0
 Function Label : MP000
 Title : Sistema de Contabilidad (Nivel 0)

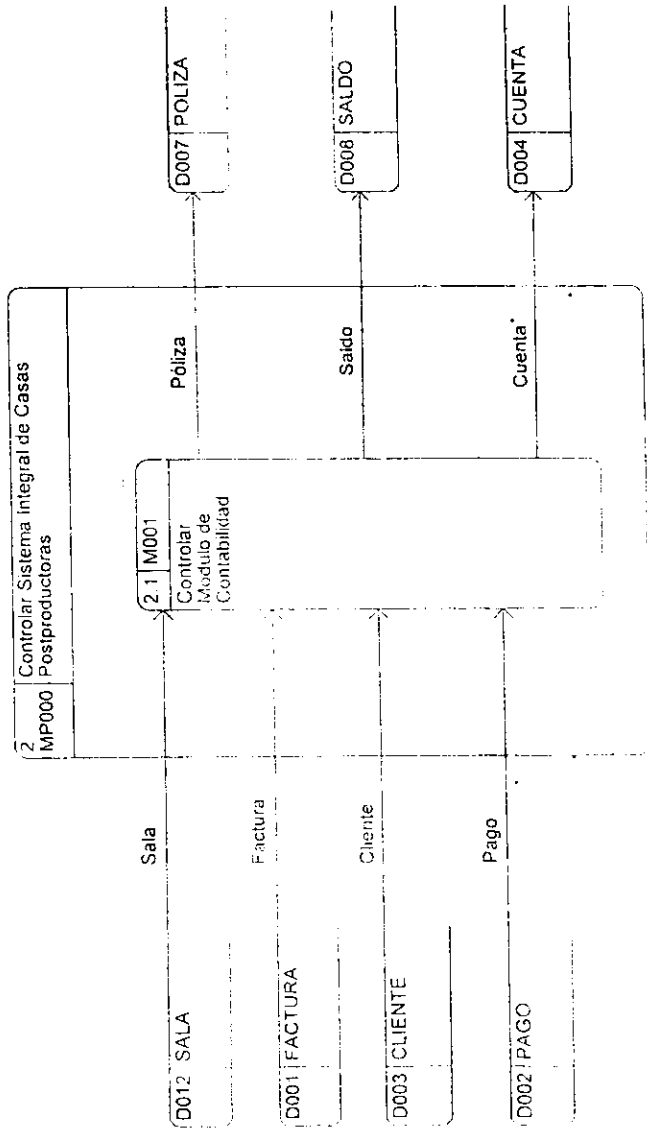


Fig. 3.2.1.A. Diagrama de Flujo de Datos de Contabilidad. (Nivel 0).

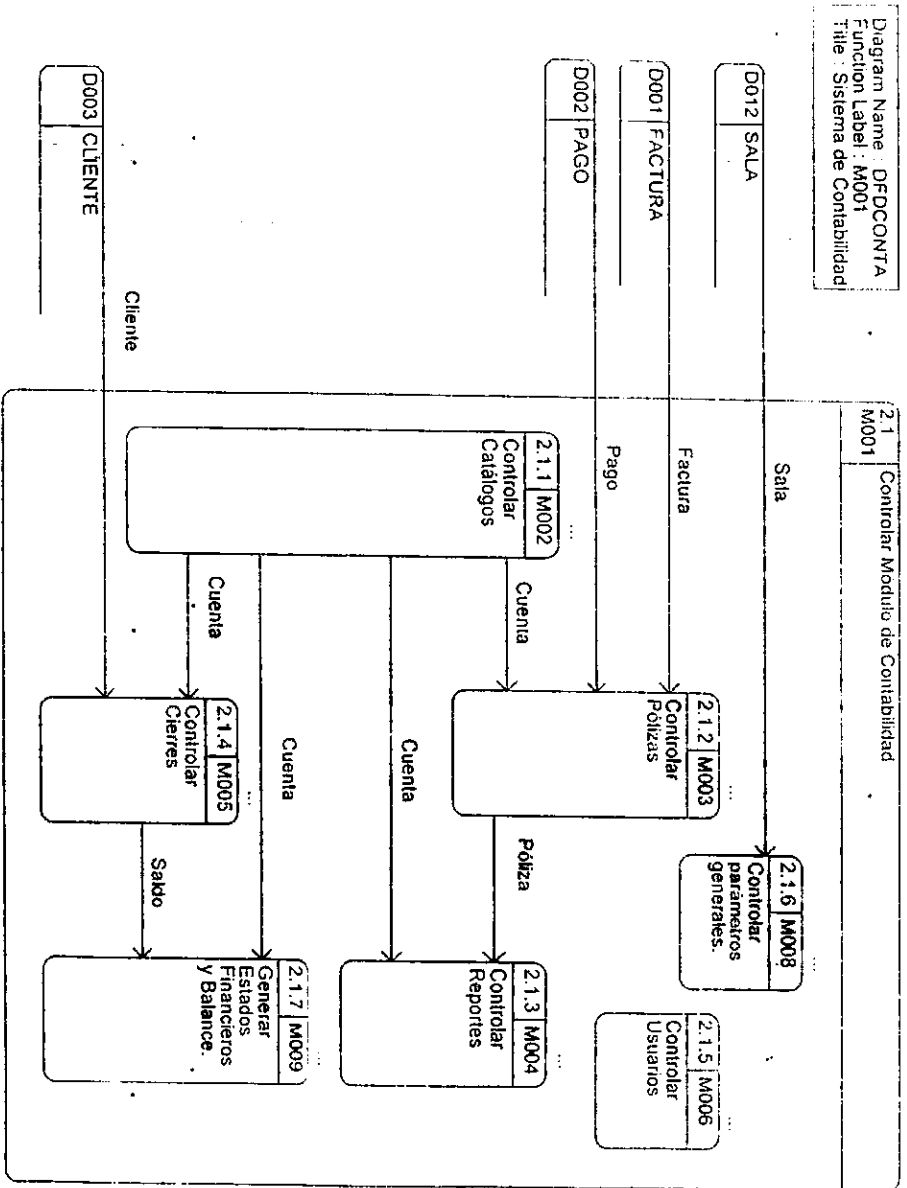


Fig. 3.2.1.B. DFD. Controlador Módulo de Contabilidad.

Diagram Name : DFDCONTA:
 Function Label : M002
 Title : Sistema de Contabilidad (Nivel

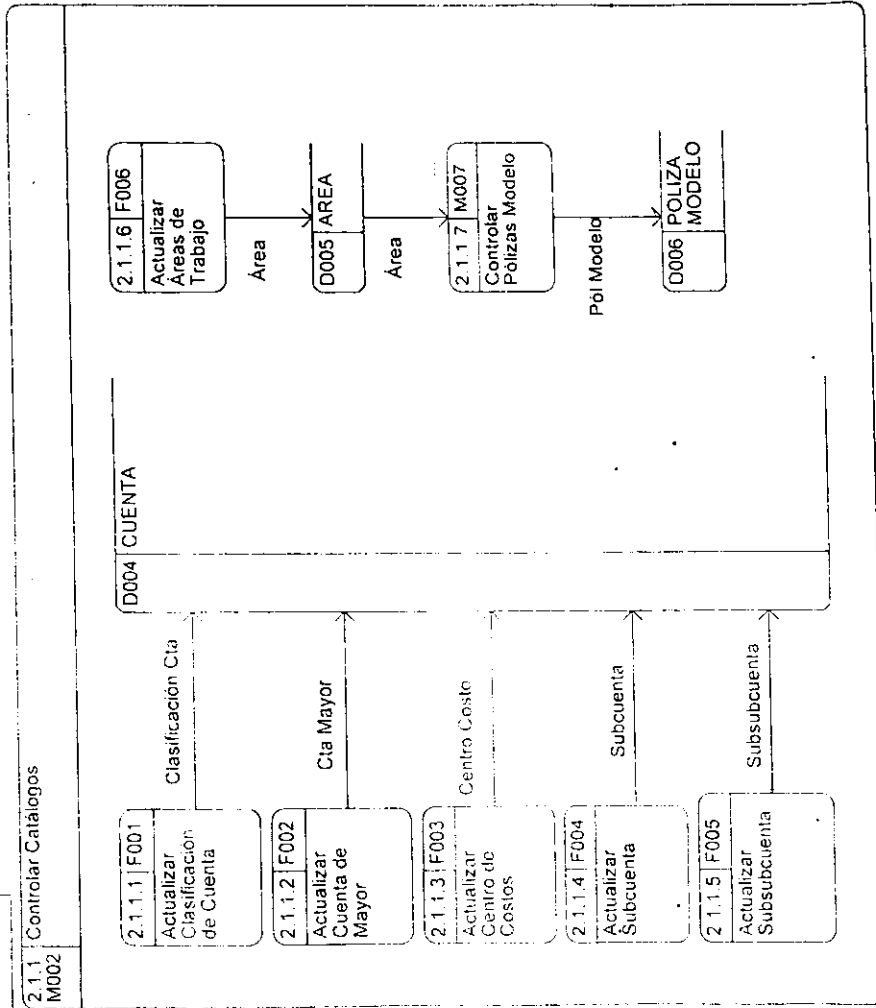


Fig. 3.2.1.C. DFD. Controlar Catálogos de Contabilidad.

Diagram Name : DFDCONTA2
 Function Label : M003
 Title : Sistema de Contabilidad (Nivel 2)

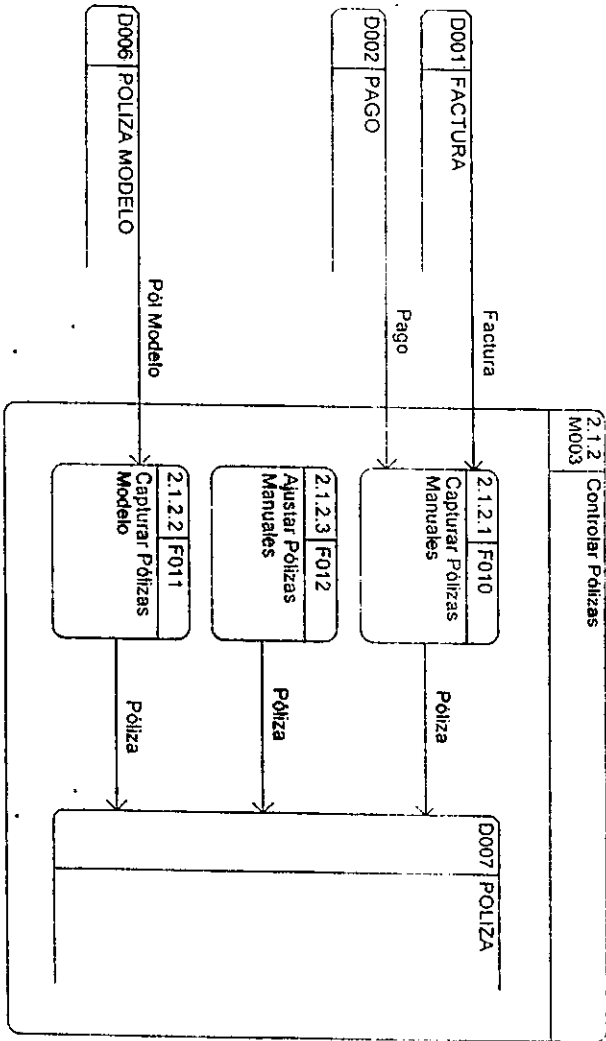


Fig. 3.2.1.D. DFD. Controlar Pólizas.

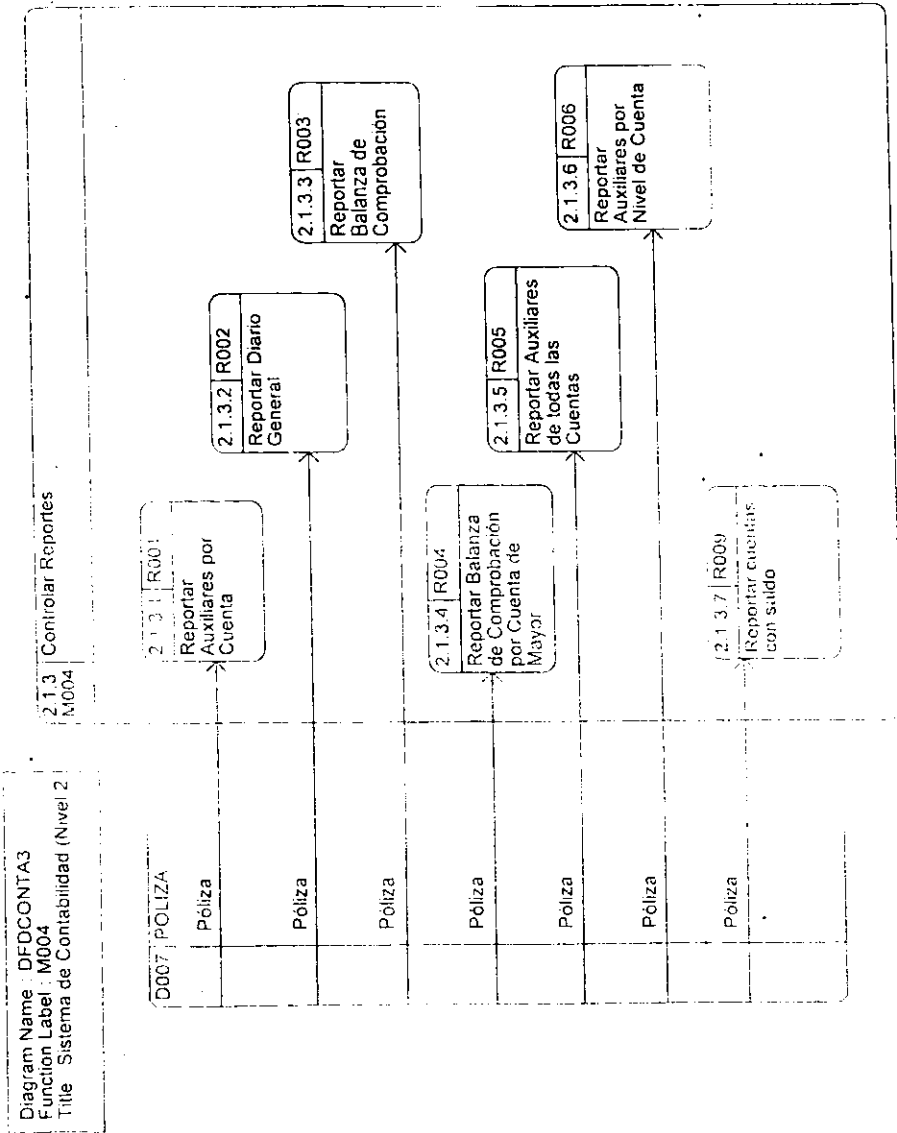


Fig. 3.2.1.E. DFD. Controlar Reportes de Contabilidad.

Diagram Name : DECCONTA4
Function Label : M005
Title : Sistema de Contabilidad (Nivel 2)

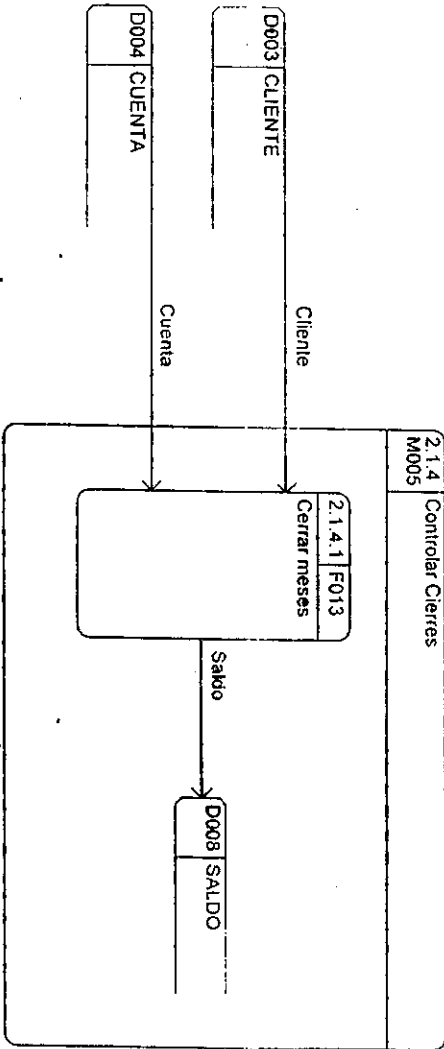


Fig. 3.2.1.F. DFD. Controlar Cierres.

Diagram Name : DFDCONTAS
 Function Label : M006
 Title : Sistema de Contabilidad (Nivel 2)

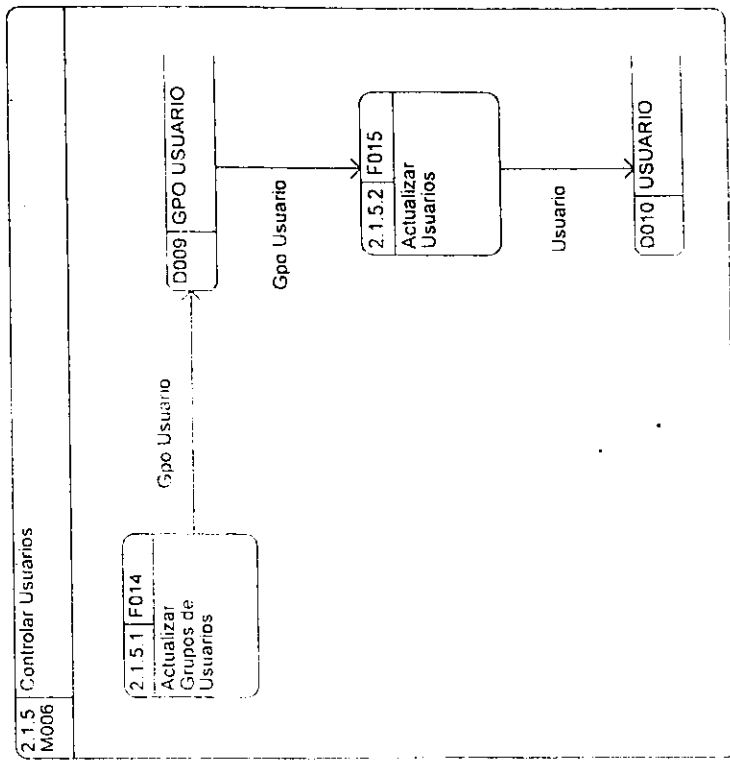


Fig. 3.2.1.G. DFD. Controlar Usuarios.

Diagram Name : DFDCONTA6
Funcion Label : M008
Title : Sistema de Contabilidad (Nivel 2)

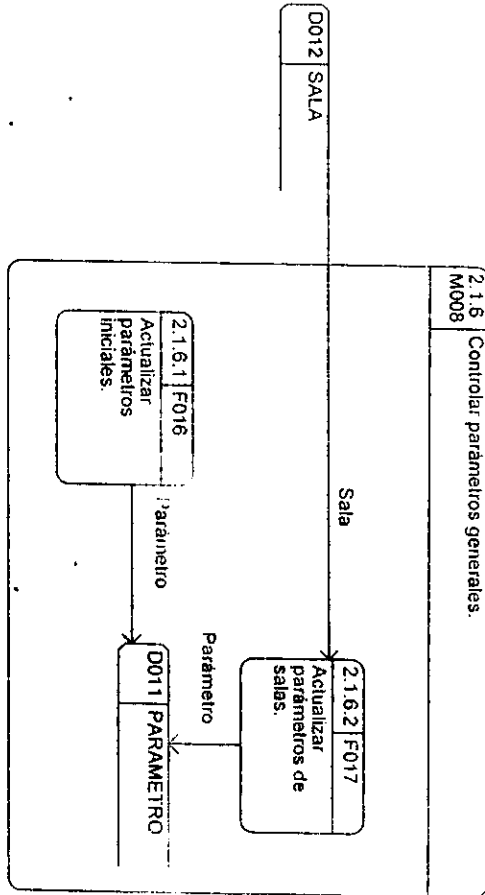


Fig. 3.2.1.H. DFD. Controlar Parámetros Generales.

Diagram Name : DFDCONTAT7
 Function Label : M009
 Title : Sistema de Contabilidad (Nivel 2)

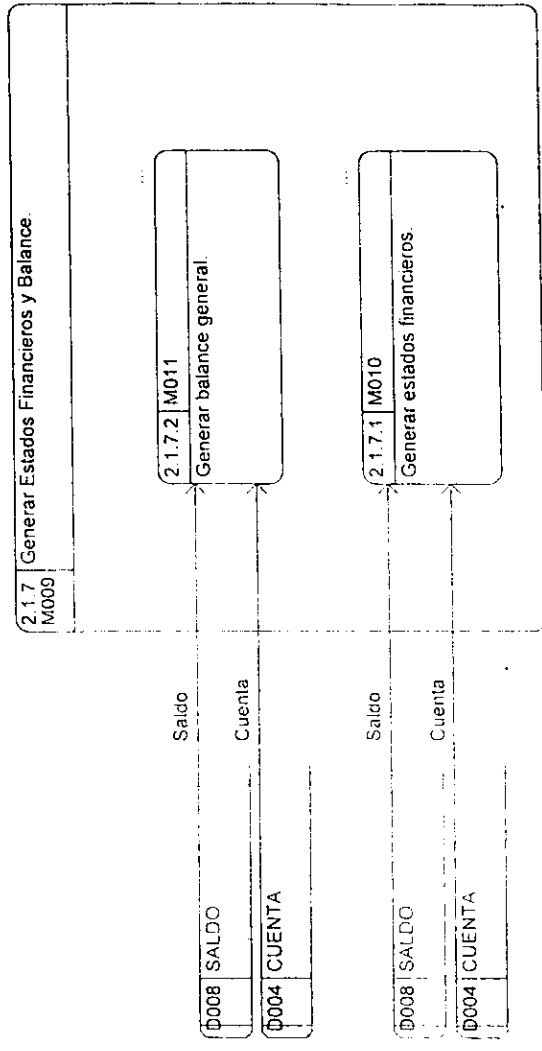


Fig. 3.2.1.I. DFD. Generar Estados Financieros y Balance.

Funcion Label : MP000
Tite Sistema de Cuentas por Cobrar (Nivel 0)

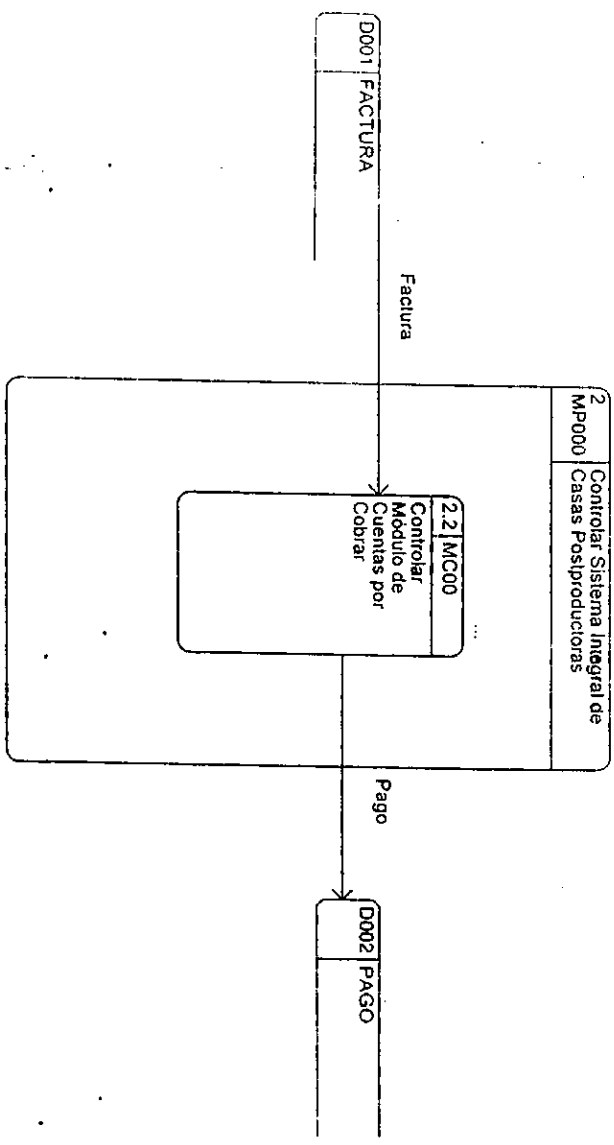


Fig. 3.2.1.J. Diagrama de Flujo de Datos de Cuentas por Cobrar. (Nivel 0).

Function Label: MC00
 Title: Sistema de Cuentas por Cobrar (Nivel 1)

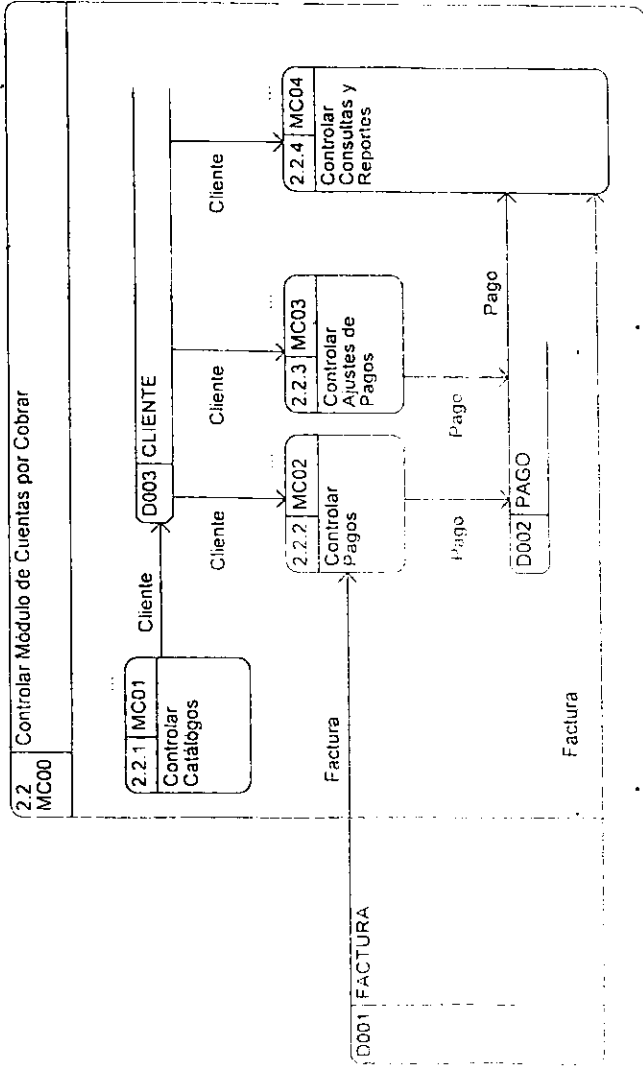


Fig. 3.2.1.K. DFD. Controlar Cuentas por Cobrar.

Función Label: MCC01
7.4. Sistema de Cuentas por Cobrar (Nivel 2)

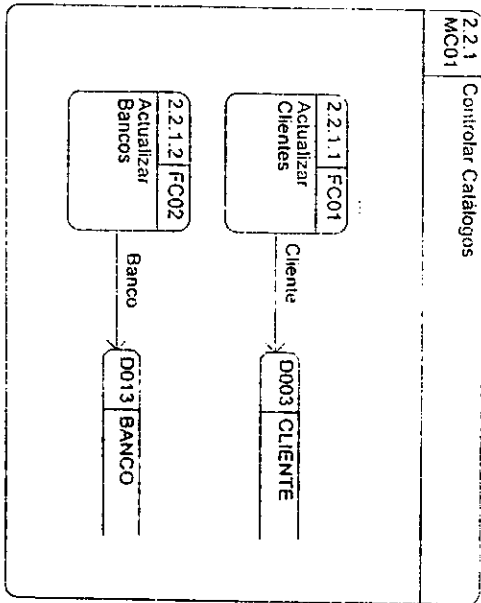


Fig. 3.2.1.L. DFD. Controlar Catálogos de Cuentas x Cobrar.

Function Label: MCO2
 Title: Sistema de Cuentas por Cobrar (Nivel 1)

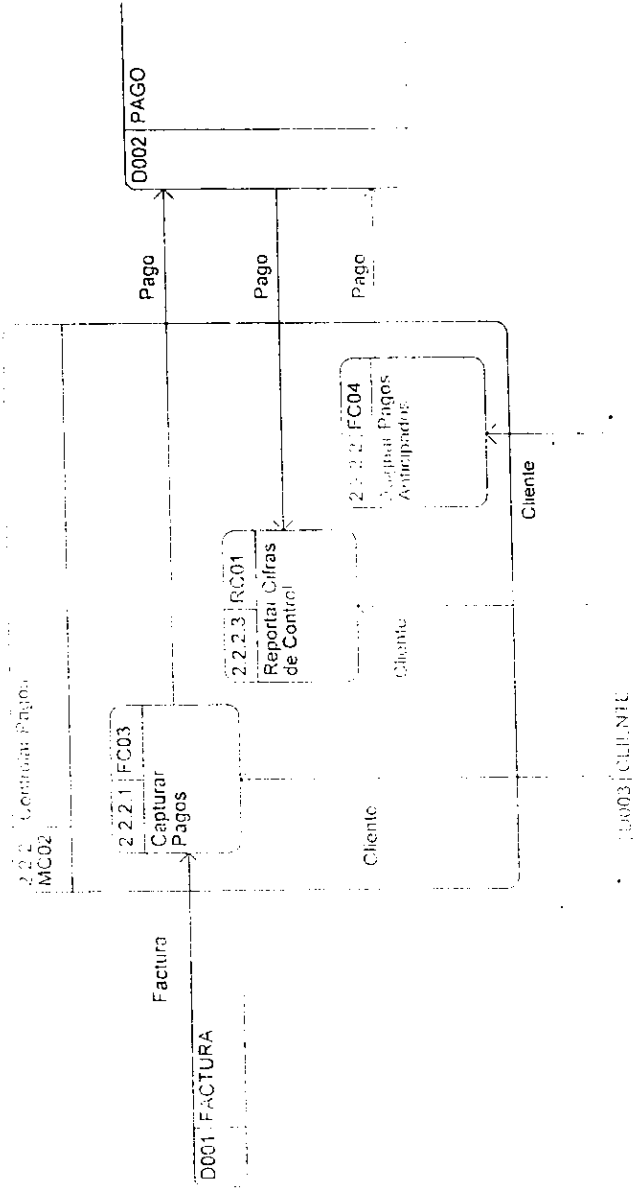


Fig. 3.2.1.M. DFD. Controlar Pagos.

Funcion Label: MC03
Título: Sistema de Cuentas por Cobrar (Nivel 2)

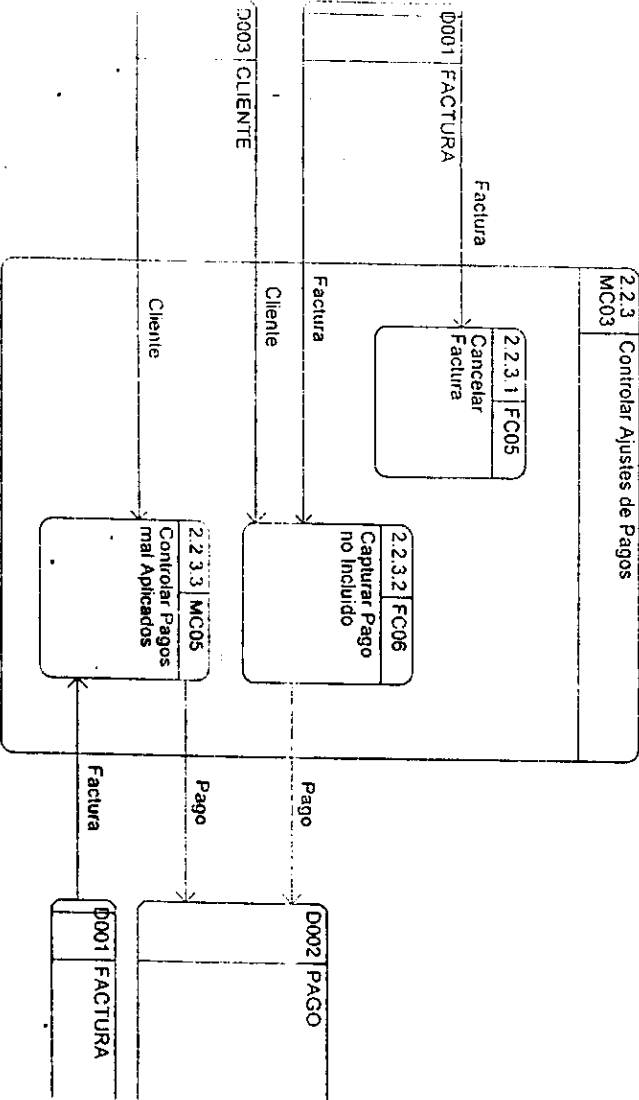


Fig. 3.2.1.O. DFD. Controlar Consultas y Reportes de Cuentas x Cobrar.

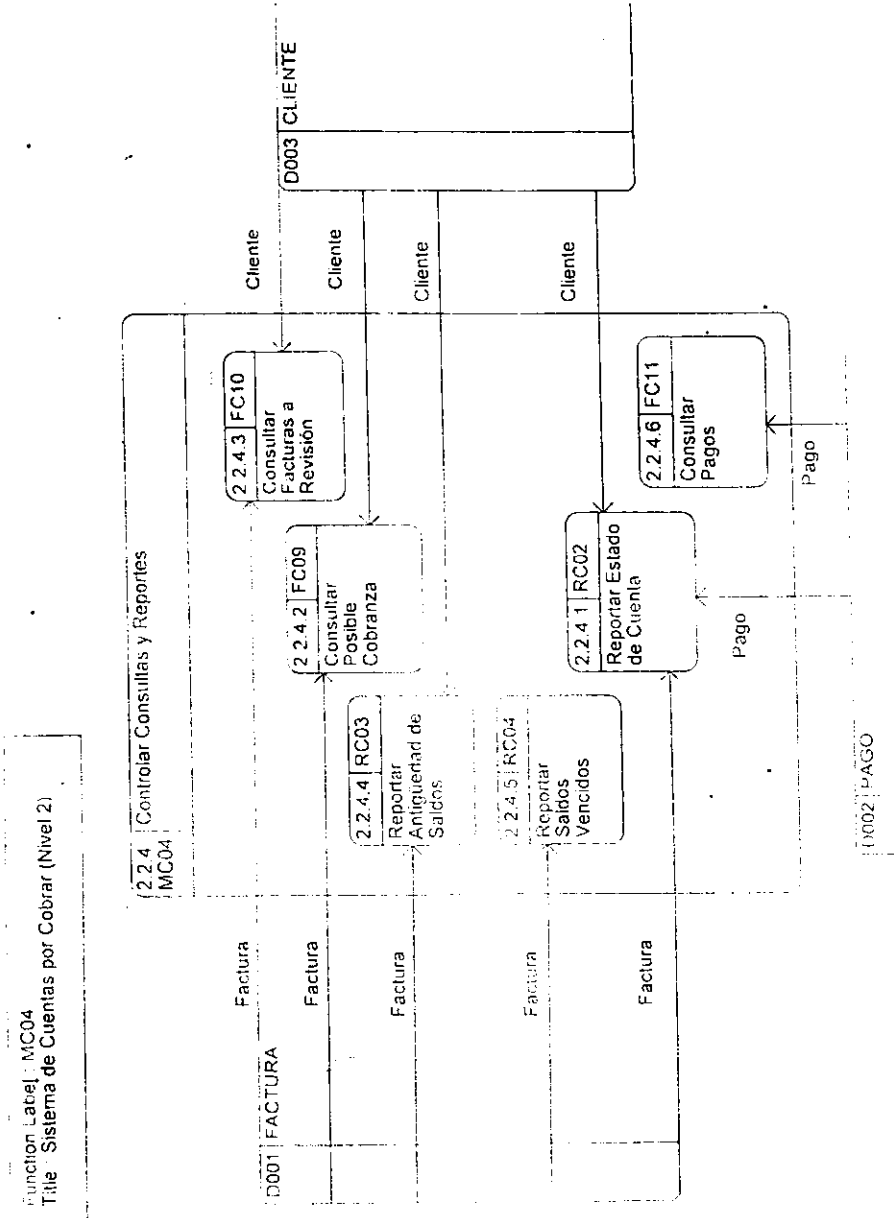


Fig. 3.2.1.N. DFD. Controlar Ajustes de Pagos.

3.2.2. Diagrama Funcional.

A continuacion presentamos la descripcion, detalles y generalidades de las funciones que conforman al sistema. Las convenciones utilizadas para nombrar a cada una de las funciones son:

- El primer caracter es una letra: F para funcion, M para menu y R para reporte.
- Un numero consecutivo por tipo de modulo de tres caracteres.

Se aade tambien las entidades que la funcion utiliza, donde se cataloga cada una de los procesos posibles para cada entidad, que son:

- Cre: permitir o no la creacion o insercion de datos.
- Con: permitir o no la consulta de la informacion.
- Act: permitir o no la actualizacion de los registros.
- Borr: permitir o no el borrado de algun dato.

Toda la informacion de este documento son resultado a su vez, de los reportes entregados por la herramienta de CASE.

Funcion: F001	Elemental ?: S	Frecuencia: 5 por Mes
Antecesor: M002	Subdividida ?: N	Respuesta: Inmediata
Definicion: Actualizar Clasificacion de Cuenta. Permite consultar, dar de alta, modificar, borrar y reportar registros del catalogo de Clasificacion de Cuenta.		

Entidades usadas	Cre	Con	Act	Borr
CUENTA CLASIFICACION	S	S	S	S

Funcion: F002	Elemental ?: S	Frecuencia: 5 por Mes
Antecesor: M002	Subdividida ?: N	Respuesta: Inmediata
Definicion: Actualizar Cuenta de Mayor. Permite consultar, dar de alta, modificar, borrar y reportar registros del catalogo de Cuenta de Mayor.		

Entidades usadas	Cre	Con	Act	Borr
CUENTA MAYOR	S	S	S	S

Funcion: F003	Elemental ?: S	Frecuencia: 5 por Mes
Antecesor: M002	Subdividida ?: N	Respuesta: Inmediata
Definicion: Actualizar Centro de Costos. Permite consultar, dar de alta, modificar, borrar y reportar registros del catalogo de Centro de Costos.		

Entidades usadas	Cre	Con	Act	Borr
CENTRO COSTOS	S	S	S	S

Función: F004 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M002 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Subcuenta. Permite consultar, dar de alta, modificar, borrar y reportar registros del catálogo de Subcuenta.

Entidades usadas	Cre	Con	Act	Borr
SUBCUENTA	S	S	S	S

Función: F005 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M002 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Subsubcuenta. Permite consultar, dar de alta, modificar, borrar y reportar registros del catálogo de Subsubcuenta.

Entidades usadas	Cre	Con	Act	Borr
SUBSUBCUENTA	S	S	S	S

Función: F006 **Elemental ?:** S **Frecuencia:** 1 por Año
Antecesor: M002 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Áreas de Trabajo. Permite consultar, dar de alta, modificar, borrar y reportar registros del catálogo de Áreas de Trabajo.

Entidades usadas	Cre	Con	Act	Borr
AREA	S	S	S	S

Función: F007 **Elemental ?:** S **Frecuencia:** 1 por Año
Antecesor: M007 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Tipos de Pólizas. Permite consultar, dar de alta, modificar, borrar y reportar registros del catálogo de Tipos de Pólizas.

Entidades usadas	Cre	Con	Act	Borr
TIPO POLIZA	S	S	S	S

Función: F008 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M007 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Pólizas Modelo. Permite consultar, dar de alta, modificar, borrar y reportar registros del catálogo de Pólizas Modelo, estas se capturan organizadas por área de trabajo para evitar que usuarios de unas áreas accedan a pólizas modelo que no corresponden a sus actividades..

Entidades usadas	Cre	Con	Act	Borr
POLIZA MODELO	S	S	S	S
AREA		S		

Función: F009 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M007 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Asignar las Cuentas a las Pólizas Modelo. Esto es, indicar para cada póliza modelo cuáles cuentas contables debe utilizar para cargar y cuáles para abonar, al momento que sea seleccionada por el usuario.

Entidades usadas	Cre	Con	Act	Borr
POLIZA MODELO CUENTA	S	S	S	S
SUBSUBCUENTA		S		

Función: F010 **Elemental ?:** S **Frecuencia:** 50 por Mes
Antecesor: M003 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Capturar Pólizas Manuales. Permitir consultar, reportar y dar de alta, pólizas indicando cuáles cuentas contables deben ser usadas directamente por el usuario.

Entidades usadas	Cre	Con	Act	Borr
PARTIDA	S	S		
POLIZA	S	S		
SUBSUBCUENTA		S		
TIPO POLIZA		S		

Función: F011 **Elemental ?:** S **Frecuencia:** 15 por Mes
Antecesor: M003 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Capturar Pólizas Modelo. Una vez dados de alta los catálogos de pólizas modelo, en esta funcionalidad se puede seleccionar una de ellas para ser llenadas por el usuario dependiendo el área a la que pertenezca.

Entidades usadas	Cre	Con	Act	Borr
PARTIDA	S	S		
POLIZA	S	S		
POLIZA MODELO		S		
SUBSUBCUENTA		S		
TIPO POLIZA		S		
AREA		S		

Función: F012 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M003 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Ajustar Pólizas Manuales. Una vez que la póliza ha sido dada de alta, ya sea por medios automáticos o bien, manualmente, y se ha cerrado el mes contable, esta funcionalidad permite ingresar una nueva póliza (de ajuste) que repercuta contablemente en este mes pero con fecha del mes cerrado.

Entidades usadas	Cre	Con	Act	Borr
PARTIDA	S	S		
POLIZA	S	S		
SUBSUBCUENTA		S		
TIPO POLIZA		S		

Función: F013 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M005 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Cerrar meses. Al finalizar el mes contable en curso, se corre el cierre para traspasar los saldos de las cuentas contables y de los clientes al siguiente mes contable.

Entidades usadas	Cre	Con	Act	Borr
CIERRE	S	S		
CIERRE CUENTA	S	S		
CLIENTE		S		
CLIENTE_CIERRE	S	S		
PARTIDA		S		
SUBSUBCUENTA		S		

Función: F014 **Elemental ?:** S **Frecuencia:** 5 por Año
Antecesor: M006 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Grupos de Usuarios. En la parte de contabilidad se manejan los privilegios de los usuarios que pueden acceder a este módulo y al de Cuentas por pagar. Los Grupos delimitan ciertos privilegios que posteriormente heredarán los usuarios de dichos grupos.

Entidades usadas	Cre	Con	Act	Borr
GRUPO USUARIO	S	S	S	S

Función: F015 **Elemental ?:** S **Frecuencia:** 3 por Mes
Antecesor: M006 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Usuarios. Permite consultar, dar de alta, modificar, borrar y reportar registros del catálogo de Usuarios que podrán utilizar el sistema, bajo ciertos privilegios del grupo al que pertenecen.

Entidades usadas	Cre	Con	Act	Borr
AREA		S		
GRUPO USUARIO		S		
USUARIO	S	S	S	S

Función: F016 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M008 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar parámetros iniciales. Permite consultar, dar de alta, modificar, borrar y reportar registros de los parámetros de Inicio, es decir de los valores contables de default que el sistema automáticamente usará cuando realice operaciones previamente definidas.

Entidades usadas	Cre	Con	Act	Borr
AREA		S	S	
PARAMETRO_CUENTAS	S	S	S	
TIPO POLIZA		S	S	

Función: F007 **Elemental ?:** S **Frecuencia:** 5 por Año
Antecesor: M008 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar parámetros de salas. Como las salas son los centros de costo dentro de la contabilidad de la empresa, es necesario saber las cuentas de ventas y devoluciones de estas para poder realizar las pólizas automáticas en cada factura o cancelación de factura.

Entidades usadas	Cre	Con	Act	Borr
SALA	S	S		
SUBSUBCUENTA		S		

Función: F018 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M010 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar las cuentas para el estado financiero. Para formar el Estado Financiero es necesario indicar cuáles cuentas contables formarán cada uno de los elementos que intervienen en el cálculo de la utilidad neta.

Entidades usadas	Cre	Con	Act	Borr
COSTO PARAMETRO MES	S	S	S	S
COSTO SALA MES	S	S	S	S
PARAMETRO_CUENTA		S		
SALA		S		
SUBSUBCUENTA		S		
TIPO COSTO		S		

Función: F019 **Elemental ?:** S **Frecuencia:** 5 por Mes
Antecesor: M011 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar las cuentas para el balance general. Para formar el Balance General es necesario indicar cuáles cuentas contables formarán cada uno de los elementos del balance (activo, pasivo y capital).

Entidades usadas	Cre	Con	Act	Borr
COSTO BALANCE MES	S	S	S	S
SUBSUBCUENTA		S		
TIPO BALANCE		S		

Función: FC01 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC01 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Clientes. Permite consultar, dar de alta, modificar, borrar y reportar registros del catálogo de Clientes.

Entidades usadas	Cre	Con	Act	Borr
CLIENTE	S	S	S	S
ENTIDAD		S		

Función: FC02 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC01 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Bancos. Permite consultar, dar de alta, modificar, borrar y reportar registros del catálogo de Bancos.

Entidades usadas	Cre	Con	Act	Borr
BANCO	S	S	S	S

Función: FC03 **Elemental ?:** S **Frecuencia:** 10 por Día
Antecesor: MC02 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Capturar Pagos. Dada la información del cliente, este puede realizar un pago ordinario (pago de facturas), o anticipado ya sea en efectivo, con cheque, nota de crédito, tarjeta bancaria o transferencia electrónica del adeudo contraído.

Entidades usadas	Cre	Con	Act	Borr
BANCO		S		
CLIENTE		S		
FACTURA		S		
PAGO	S			
PAGO FACTURA	S			
TIPO PAGO		S		

Función: FC04 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC02 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Asignar Pagos Anticipados. En caso de haber capturado pagos anticipados del Cliente, esta función permite asignar dichos pagos a las facturas con fecha posterior al pago, para que queden saldadas.

Entidades usadas	Cre	Con	Act	Borr
BANCO		S		
CLIENTE		S		
FACTURA		S		
PAGO		S		
PAGO FACTURA	S			
TIPO PAGO		S		

Función: FC05 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC03 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Cancelar Factura. Una vez facturada una orden de trabajo, se realizan todos los movimientos de la venta, si esta es anulada se realizan los procesos inversos afectando las cuentas contables leídas de los parámetros de inicio y de la factura cancelada.

Entidades usadas	Cre	Con	Act	Borr
CLIENTE		S	S	
FACTURA		S	S	
PARTIDA	S			
PARAMETRO INICIO		S		
POLIZA	S			
STATUS		S		

Función: FC06 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC03 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Capturar Pago no Incluido. Una vez cerrado el mes contable en curso, no es posible meter pagos posteriores con la fecha del mes cerrado, esta funcionalidad lo permite generando pólizas de ajuste del pago no capturado.

Entidades usadas	Cre	Con	Act	Borr
BANCO		S		
CLIENTE		S		
FACTURA		S		
PAGO	S			
PAGO FACTURA	S			
TIPO PAGO		S		

Función: FC07 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC05 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Cheques Devueltos. Cuando el banco regresa el cheque del Cliente por alguna anomalía, la funcionalidad permite su cancelación y actualización de saldos y estatus.

Entidades usadas	Cre	Con	Act	Borr
BANCO		S		
CLIENTE		S	S	
PAGO		S	S	
PAGO FACTURA				S
TIPO PAGO		S		
STATUS		S		

Función: FC08 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC05 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Actualizar Pagos Mal Aplicados. Cuando un pago ha sido capturado incorrectamente, la única forma de corregirlo es por medio de esta funcionalidad. La función reescribe los datos incorrectos por los adecuados que un usuario con privilegios acordados realice.

Entidades usadas	Cre	Con	Act	Borr
BANCO		S		
CLIENTE		S	S	
PAGO		S	S	
PAGO FACTURA				S
TIPO PAGO		S		
STATUS		S		

Función: FC09 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC04 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Consultar Posible Cobranza. Dada una fecha, la clave de un cliente (o ninguna para seleccionar a todos), despliega las facturas pendientes de pago, ya sea total o parcial, no liquidadas hasta el día marcado en la fecha.

Entidades usadas	Cre	Con	Act	Borr
CLIENTE		S		
FACTURA		S		

Función: FC10 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC04 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Consultar Facturas a Revisión. Dependiendo el día de la semana en que el Cliente recibe facturas, el sistema presenta en pantalla aquéllas facturas que para una fecha dada deban ser enviadas al Cliente.

Entidades usadas	Cre	Con	Act	Borr
CLIENTE		S		
FACTURA		S		

Función: FC11 **Elemental ?:** S **Frecuencia:** 3 por Día
Antecesor: MC04 **Subdividida ?:** N **Respuesta:** Inmediata
Definición: Consultar Pagos. Funcionalidad que permite revisar todos los pagos que los Clientes han realizado hasta el momento.

Entidades usadas	Cre	Con	Act	Borr
BANCO		S		
CLIENTE		S		
PAGO		S		
PAGO FACTURA		S		

Función: M001 **Elemental ?:** No **Frecuencia:**
Antecesor: MP000 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Controlar Módulo de Contabilidad. (MENÚ).

Entidades usadas	Cre	Con	Act	Borr
------------------	-----	-----	-----	------

Función: M002 **Elemental ?:** No **Frecuencia:**
Antecesor: M001 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Controlar Catálogos. (MENÚ).

Entidades usadas	Cre	Con	Act	Borr
------------------	-----	-----	-----	------

Función: M010 **Elemental ?:** No **Frecuencia:**
Antecesor: M009 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Generar estados financieros. (MENÚ).

Entidades usadas **Cre** **Con** **Act** **Borr**

Función: M011 **Elemental ?:** No **Frecuencia:**
Antecesor: M009 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Generar balance general. (MENÚ).

Entidades usadas **Cre** **Con** **Act** **Borr**

Función: MC00 **Elemental ?:** No **Frecuencia:**
Antecesor: MP000 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Controlar Módulo de Cuentas por Cobrar. (MENÚ).

Entidades usadas **Cre** **Con** **Act** **Borr**

Función: MC01 **Elemental ?:** No **Frecuencia:**
Antecesor: MC00 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Controlar Catálogos. (MENÚ).

Entidades usadas **Cre** **Con** **Act** **Borr**

Función: MC02 **Elemental ?:** No **Frecuencia:**
Antecesor: MC00 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Controlar Pagos. (MENÚ).

Entidades usadas **Cre** **Con** **Act** **Borr**

Función: MC03 **Elemental ?:** No **Frecuencia:**
Antecesor: MC00 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Controlar Ajustes de Pagos. (MENÚ).

Entidades usadas **Cre** **Con** **Act** **Borr**

Función: MC04 **Elemental ?:** No **Frecuencia:**
Antecesor: MC00 **Subdividida ?:** S **Respuesta:** Inmediata
Definición: Controlar Consultas y Reportes. (MENÚ).

Entidades usadas **Cre** **Con** **Act** **Borr**

Función: R004 **Elemental ?:** S **Frecuencia:** 5 por Día
Antecesor: M004 **Subdividida ?:** No **Respuesta:** Inmediata
Definición: Reportar Balanza de Comprobación por Cuenta de Mayor. Reporta los cortes contables de la Cuenta de Mayor y sus saldos iniciales respectivos.

Entidades usadas	Cre	Con	Act	Borr
CIERRE CUENTA		S		
PARTIDA		S		
SUBSUBCUENTA		S		
CIERRE CUENTA		S		

Función: R005 **Elemental ?:** S **Frecuencia:** 5 por Día
Antecesor: M004 **Subdividida ?:** No **Respuesta:** Inmediata
Definición: Reportar Auxiliares de todas las Cuentas. Genera las cuentas con sus pólizas para un rango de fechas dado y para todas las cuentas.

Entidades usadas	Cre	Con	Act	Borr
CIERRE CUENTA		S		
PARTIDA		S		
SUBSUBCUENTA		S		

Función: R006 **Elemental ?:** S **Frecuencia:** 5 por Día
Antecesor: M004 **Subdividida ?:** No **Respuesta:** Inmediata
Definición: Reportar Auxiliares por Nivel de Cuenta. Genera las cuentas con sus pólizas para un rango de fechas dado y por nivel de cuenta, es decir, por clasificación, cuenta de mayor, centro de costos, subcuenta y subsubcuenta.

Entidades usadas	Cre	Con	Act	Borr
CIERRE CUENTA		S		
PARTIDA		S		
SUBSUBCUENTA		S		

Función: R007 **Elemental ?:** S **Frecuencia:** 5 por Día
Antecesor: M010 **Subdividida ?:** No **Respuesta:** Inmediata
Definición: Reportar el estado financiero, de resultados. Genera la tabla con los elementos del estado de resultados y cada uno de los centros de costos con su respectivo presupuesto, valor real y porcentaje de variación.

Entidades usadas	Cre	Con	Act	Borr
COSTO CUENTA MES		S		
COSTO SALA MES		S		
PARAMETRO_CUENTA		S		
SALA		S		
SUBSUBCUENTA		S		
TIPO COSTO		S		

Función: R008 **Elemental ?:** S **Frecuencia:** 5 por Día
Antecesor: M011 **Subdividida ?:** No **Respuesta:** Inmediata
Definición: Reportar el balance general. Genera el balance general para una fecha dada tomando la información de los datos previamente seleccionados.

Entidades usadas	Cre	Con	Act	Borr
COSTO BALANCE MES		S		
SUBSUBCUENTA		S		
TIPO BALANCE		S		

Función: R009 **Elemental ?:** S **Frecuencia:** 5 por Día
Antecesor: M004 **Subdividida ?:** No **Respuesta:** Inmediata
Definición: Reportar cuentas con saldo. Genera un listado de las cuentas contables añadiéndoles su saldo actual.

Entidades usadas	Cre	Con	Act	Borr
CIERRE CUENTA		S		
PARTIDA		S		
SUBSUBCUENTA		S		

Función: RC01 **Elemental ?:** S **Frecuencia:** 5 por Día
Antecesor: MC02 **Subdividida ?:** No **Respuesta:** Inmediata
Definición: Reportar Cifras de Control. Genera un listado de todos los pagos realizados en el día.

Entidades usadas	Cre	Con	Act	Borr
BANCO		S		
CLIENTE		S		
PAGO		S		
TIPO PAGO		S		

Función: RC02 **Elemental ?:** S **Frecuencia:** 5 por Día
Antecesor: MC04 **Subdividida ?:** No **Respuesta:** Inmediata
Definición: Reportar Estado de Cuenta. En este reporte se muestra en forma cronológica, tanto los pagos, las facturas y los cheques cancelados que uno o todos los cliente hayan tenido en un rango de fechas dadas.

Entidades usadas	Cre	Con	Act	Borr
CIERRE CLIENTE		S		
CLIENTE		S		
FACTURA		S		
PAGO		S		
STATUS		S		
TIPO PAGO		S		

Nombre : **BANCO**

Nombre Corto : **BAN**

Subtipo de :

Volumen Inicial : 7

Volumen Promedio: 7

Volumen Máximo : 15

%Crecimiento Anual : 1

Descripción:

Cada una de las instituciones bancarias de quienes se reciben pagos, y de donde, posteriormente, se generarán cheques.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE	CVEINT	N	INTEGER	6
DIRECCION	DESCLARGA	N	VARCHAR2	60
NOMBRE	DESCORTA	N	VARCHAR2	15
RESPONSABLE	DESCORTA	Y	VARCHAR2	15
TELEFONO	DESCORTA	Y	VARCHAR2	15
CVE	Identificador único del BANCO.			
DIRECCION	Dirección completa de la sucursal del BANCO.			
NOMBRE	Nombre completo del BANCO.			
RESPONSABLE	Nombre de la persona con la cual se tiene relación en el BANCO.			
TELEFONO	Teléfonos y faxes del BANCO.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR Agrupado a uno o más PAGO

Nombre : **CENTRO COSTOS**

Nombre Corto : **CCO**

Subtipo de :

Volumen Inicial : 10

Volumen Promedio: 10

Volumen Máximo : 20

%Crecimiento Anual : 0.5

Descripción:

Parte de la cuenta contable que identifica cada uno de los elementos contables principales. En las empresas de cine y vídeo, éstas pueden ser las salas donde se generan los trabajos que se facturan a los clientes.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	3
DESCRIPCION	DESCLARGA	N	VARCHAR2	60
TIPO CUENTA		N	INTEGER	1
CVE	Identificador único del CENTRO DE COSTOS.			
DESCRIPCION	Nombre completo del CENTRO DE COSTOS.			
TIPO CUENTA	Indicador para saber si se trata de una cuenta detalle o totalizadora, es decir, que puede no tener o tener cuentas que dependan de ella.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR Conteniendo una o más SUBCUENTA
 DEBE ESTAR Sujeta a una y sólo una CUENTA MAYOR

Nombre : CIERRE

Nombre Corto : CIE

Subtipo de :

Volumen Inicial : 0

V. Promedio: 50

Volumen Máximo : 100

%Crecimiento Anual : 1

Descripción:

Muestra los meses contables que se han trabajado, limitando las operaciones diarias al mes contable en curso. Engloba cada uno de los meses contables que son terminados, revisados, cotejados y declarados.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CERRADO		N	INTEGER	1
FECHA		N	DATE	*
CERRADO	Indicador para saber si el mes contable está o no cerrado.			
FECHA	Fecha que se cierra (interesa sólo mes y año).			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR Cerrando una o más CIERRE CUENTA

Nombre : CIERRE CUENTA

Nombre Corto : CIC

Subtipo de :

Volumen Inicial : 50

Volumen Promedio: 300 Volumen

Máximo : 700

%Crecimiento Anual : 10

Descripción:

Es una tabla intermedia donde se concentran todas las cuentas contables con su respectivo saldo de cierre de mes contable.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
SALDO CIERRE	MONTO	N MONEY	10,2	

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR Cerrada por uno y sólo un CIERRE *

DEBE ESTAR Cerrando uno y sólo una SUBSUBCUENTA *

Nombre : CLIENTE

Nombre Corto : CLI

Subtipo de : COMPRADOR

Volumen Inicial : 100

Volumen Promedio : 300

Volumen Máximo : 1000

%Crecimiento Anual : 10

Descripción

Cada uno de las personas que solicitan un servicio a la Empresa, que se les factura y se les cobra.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	6
DIAS_CREDITO		N	INTEGER	2
DIA_PAGO		N	INTEGER	2
DIA_REVISION		N	INTEGER	2
NOMBRE		N	VARCHAR2	15
RAZON_SOCIAL		N	VARCHAR2	60
RFC		N	CHAR	14
%_DESCUENTO		Y	NUMBER	3,1
CLASIFICACION		Y	VARCHAR2	60
COLONIA		Y	VARCHAR2	60
CP		Y	CHAR	5
DELEGACION		Y	VARCHAR2	60
DIRECCION		Y	VARCHAR2	60
FAX		Y	VARCHAR2	60
LIMITE_CREDITO	MONTO	Y	MONEY	10,2
REFERENCIA		Y	VARCHAR2	60
SALDO_VENCIDO	MONTO	Y	MONEY	10,2
TELEFONO		Y	VARCHAR2	60

CVE	Identificador único del CLIENTE.
DIAS_CREDITO	Número de días que el CLIENTE tiene de gracia para pagar.
DIA_PAGO	Día de la semana cuando el CLIENTE emite sus cheques.
DIA_REVISION	Día de la semana cuando el CLIENTE recibe facturas.
NOMBRE	Nombre completo del CLIENTE.
RAZON_SOCIAL	Nombre dado de alta en Hacienda como comercial.
RFC	Registro Federal de Causantes del CLIENTE.
%_DESCUENTO	Porcentaje de descuento que en la venta tiene el CLIENTE.
CLASIFICACION	Calificación que se le hace al CLIENTE con respecto a su comportamiento comercial.
COLONIA	Parte de la dirección del CLIENTE.
CP	Código Postal parte de la dirección del CLIENTE.
DELEGACION	Parte de la dirección del CLIENTE.
DIRECCION	Dirección completa del CLIENTE.
FAX	Número de faxes que el CLIENTE posea.
LIMITE_CREDITO	Máxima cantidad de endeudamiento permitida al CLIENTE.
REFERENCIA	Observaciones o notas sobre el CLIENTE.
SALDO_VENCIDO	Saldo que actualmente debe el CLIENTE a la empresa.
TELEFONO	Teléfonos completos del CLIENTE.

Relaciones

Cada ocurrencia de esta Entidad :

PUEDA ESTAR	Facturando una o más	FACTURA
PUEDA ESTAR	Pagando uno o más	PAGO
PUEDA ESTAR	Saldando uno o más	CLIENTE_CIERRE

Nombre : CLIENTE_CIERRE

Nombre Corto : CLC

Subtipo de :

Volumen Inicial : 100

Volumen Promedio: 300 Volumen

Máximo : 400

%Crecimiento Anual : 10

Descripción -

Relaciona los saldos finales del mes que se CIERRA de cada uno de los CLIENTES. En esta entidad, tenemos para cada mes contable que ha sido cerrado, el saldo final de cada uno de los CLIENTES.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
SALDO	MONTO	N	MONEY	10,2
SALDO	Monto total con el que cerró el CLIENTE el mes contable.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR	Cerrado por uno y sólo un	CIERRE	*
DEBE ESTAR	Saldando uno y sólo un	CLIENTE	*

Nombre : COSTO BALANCE MES

Nombre Corto : CBM

Subtipo de :

Volumen Inicial : 4

Volumen Promedio: 24 Volumen

Máximo : 42

%Crecimiento Anual : 0

Descripción

Se almacena aquí los valores de cada uno de los elementos con los que se generan los Balances mensuales.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
FECHA	FECHA	N	DATE	*
PRESUPUESTO	MONTO	Y	MONEY	10,2
FECHA	Fecha del Balance para la cual se ingresan los valores.			
PRESUPUESTO	Cantidad presupuestada para este valor.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR	Clasificado por una y sólo una	SUBSUBCUENTA	*
DEBE ESTAR	Clasificado por uno y sólo un	TIPO BALANCE	*

Nombre : COSTO CUENTA MES

Nombre Corto : CCM

Subtipo de :

Volumen Inicial : 8

Volumen Promedio: 42 Volumen

Máximo : 84

%Crecimiento Anual : 5

Descripción

Se almacena aquí los valores de cada uno de los elementos con los que se generan los Balances mensuales, los cuales pueden tener más de una cuenta contable.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
FECHA	FECHA	N	DATE	*
PRESUPUESTO	MONTO	Y	MONEY	10,2
FECHA	Fecha del Balance para la cual se ingresan los valores.			
PRESUPUESTO	Cantidad presupuestada para este valor.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR	Clasificado por una y sólo una	SUBSUBCUENTA	*
------------	--------------------------------	--------------	---

Nombre : COSTO PARAMETRO MES

Nombre Corto : CPM

Subtipo de : Volumen Inicial : 4 Volumen Promedio: 24 Volumen
Máximo : 42 %Crecimiento Anual : 0

Descripción

Se almacena aquí los valores de cada uno de los elementos con los que se generan los Estados Financieros independientes de las salas de producción por mes.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
FECHA	FECHA	N	DATE	*
PRESUPUESTO	MONTO	N	MONEY	10,2
FECHA	Fecha del Balance para la cual se ingresan los valores.			
PRESUPUESTO	Cantidad presupuestada para este valor.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR	Clasificada por uno y sólo un	TIPO COSTO	*
DEBE ESTAR	Perteneciendo a una y sólo una	SUBSUBCUENTA	*

Nombre : COSTO SALA MES

Nombre Corto : CSM

Subtipo de : Volumen Inicial : 4 Volumen Promedio: 24 Volumen
Máximo : 42 %Crecimiento Anual : 0

Descripción

Se almacena aquí los valores de cada uno de los elementos con los que se generan los Estados Financieros dependientes de las salas de producción por mes.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
FECHA	FECHA	N	DATE	*
PRESUPUESTO	MONTO	N	MONEY	10,2
FECHA	Fecha del Balance para la cual se ingresan los valores.			
PRESUPUESTO	Cantidad presupuestada para este valor.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR	Agrupada por una y sólo una	SALA	*
DEBE ESTAR	Clasificada por uno y sólo un	TIPO COSTO	*
DEBE ESTAR	Perteneciendo a una y sólo un	SUBSUBCUENTA	*

Nombre : CUENTA

Nombre Corto : CUE

Subtipo de : Volumen Inicial : 50 Volumen Promedio: 60 Volumen
Máximo : 100 %Crecimiento Anual : 10

Descripción:

Es la clasificación de todos los niveles de cuentas contables con que cuenta la empresa, para efecto de selección de reportes. En este proyecto, el máximo es de 5.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	1 *
DESCRIPCION	DESCLARGA	N	VARCHAR2	60
CVE	Identificador único de la CUENTA.			
DESCRIPCION	Nombre completo de la CUENTA.			

Nombre : CUENTA TIPO COSTO

Nombre Corto : CTC

Subtipo de :

Volumen Inicial : 4

Volumen Promedio: 24 Volumen

Máximo : 42

%Crecimiento Anual : 0

Descripción

Se almacena aquí los valores de cada uno de los elementos con los que se generan los Estados Financieros independientes de las salas de producción por mes y que acceden a una cuenta contable en especial.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
FECHA	FECHA	N	DATE	*
PRESUPUESTO	MONTO	N	MONEY	10,2
FECHA	Fecha del Balance para la cual se ingresan los valores.			
PRESUPUESTO	Cantidad presupuestada para este valor.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR Clasificado por una y sólo un SUBSUBCUENTA *

Nombre : FACTURA

Nombre Corto : FAC

Subtipo de :

Volumen Inicial : 300

Volumen Promedio: 2000

Volumen Máximo : 4000

%Crecimiento Anual : 10

Descripción

Cada uno de los documentos donde se indican los servicios y materiales que el CLIENTE utilizó en una fecha dada. Este documento es entregado al CLIENTE para que sea liquidado según las políticas de crédito que se tengan.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
FECHA	FECHA	N	DATE	
IVA	MONTO	N	MONEY	10,2
NUM	CVEINT	N	INTEGER	6 *
FECHA_PAGO	FECHA	Y	DATE	
FECHA_PROB_PAGO	FECHA	Y	DATE	
FECHA_REVISION	FECHA	Y	DATE	
FECHA_VENCIMIENTO	FECHA	Y	DATE	
OBESTARVACION	DESCORTA	Y	VARCHAR2	15
SUBTOTAL	MONTO	Y	MONEY	10,2
TOTAL	MONTO	Y	MONEY	10,2
TOTALIVA	MONTO	Y	MONEY	10,2

FECHA Fecha cuando se facturan los servicios y materiales.
 IVA Valor actual del porcentaje del IVA.
 NUM Número consecutivo de la FACTURA.
 FECHA_PAGO Fecha cuando es pagada la FACTURA.
 FECHA_PROB_PAGO Fecha cuando probablemente sea pagada la FACTURA.
 FECHA_REVISION Fecha cuando la FACTURA debe ser entregada al CLIENTE.
 FECHA_VENCIMIENTO Fecha cuando vence la FACTURA según los días de crédito.
 OBESTARVACION Comentarios adicionales de la FACTURA.
 SUBTOTAL Monto del total antes de impuestos.
 TOTAL Monto del total después de impuestos.
 TOTALIVA Monto del total del IVA.

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR	Representado por uno y sólo un	CLIENTE
PUEDE ESTAR	Perteneciendo a uno o más	PAGO_FACTURA
DEBE ESTAR	Clasificado por uno y sólo un	STATUS
DEBE ESTAR	Facturado por uno y sólo un	CLIENTE

Nombre : FORMA_PAGO

Nombre Corto : FOP

Subtipo de : Volumen Inicial : 5 Volumen Promedio: 5 Volumen
 Máximo : 5 %Crecimiento Anual : 0

Descripción -

Cada una de las formas de pago de una FACTURA: Cheque, Efectivo, Nota de crédito, Tarjeta de crédito y Traspaso electrónico.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE	CVEINT	N	INTEGER	6 *
DESC_CORTA	DESCORTA	N	VARCHAR2	15
DESC_LARGA	DESCLARGA	N	VARCHAR2	60
CVE	Identificador único de la FORMA DE PAGO.			
DESC_CORTA	Nombre corto de la FORMA DE PAGO.			
DESC_LARGA	Nombre largo y completo de la FORMA DE PAGO.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR Clasificando uno o más PAGO

Nombre : GRUPO USUARIO

Nombre Corto : GRU

Subtipo de : Volumen Inicial : 4 Volumen Promedio: 10
 Volumen Máximo : 20 %Crecimiento Anual : 1

Descripción:

Conjunto de usuarios con iguales características de permisos y restricciones, quienes pueden tener acceso a los módulos del Sistema.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	3 *
DESC_CORTA	DESCORTA	N	VARCHAR2	15
DESC_LARGA	DESCLARGA	N	VARCHAR2	60
TIPO_GRUPO		N	INTEGER	2
TIPO_MODULO		N	INTEGER	2
CVE	Identificador único del GRUPO DE USUARIO.			
DESC_CORTA	Nombre corto del GRUPO DE USUARIO.			
DESC_LARGA	Nombre largo del GRUPO DE USUARIO.			
TIPO_GRUPO	Especifica el nivel de grupo que se trate: Superusuario, usuario de trabajo o usuario de consulta.			
TIPO_MODULO	Especifica la función o módulo que genera al GRUPO DE USUARIO, es decir, que parte del Sistema está trabajando con el GRUPO.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR Conteniendo uno o más USUARIO

Nombre : PAGO

Nombre Corto : PAG

Subtipo de :

Volumen Inicial : 200

Volumen Promedio: 1000

Volumen Máximo : 1500

%Crecimiento Anual : 20

Descripción -

Cada uno de las liquidaciones que el CLIENTE hace ya sea para pagar sus FACTURAS adeudadas, o para hacer PAGOS anticipados de servicios futuros.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE	CVEINT	N	INTEGER	6 *
FECHA	FECHA	N	DATE	
MONTO	MONTO	N	MONEY	10,2
NUM_DOCTO	CVEINT	N	INTEGER	6
OBSERVACIONES	DESCLARGA	N	VARCHAR2	60
ANTICIPO	MONTO	Y	MONEY	10,2

CVE Identificador único del PAGO.
 FECHA Fecha cuando se realiza el PAGO.
 MONTO Monto total del PAGO realizado.
 NUM_DOCTO Número del documento (cheque, nota de crédito, tarjeta, etc.) con el que se realiza el PAGO.
 OBSERVACIONES Comentarios adicionales del PAGO.
 ANTICIPO Monto total que se aplicará posteriormente.

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR	Agrupado en uno y sólo un	BANCO
PUEDE ESTAR	Pagando uno o más	PAGO_FACTURA
DEBE ESTAR	Pagado por una y sólo una	FORMA_PAGO
DEBE ESTAR	Clasificado pro uno y sólo un	STATUS
DEBE ESTAR	Pagado por uno y sólo un	CLIENTE

Nombre : PAGO_FACTURA

Nombre Corto : PAF

Subtipo de :

Volumen Inicial : 400

Volumen Promedio: 3000

Volumen Máximo : 5000

%Crecimiento Anual : 20

Descripción -

Relación de las FACTURAS que fueron liquidadas parcial o totalmente por un PAGO.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
MONTO	MONTO	N	MONEY	10,2
MONTO	Cantidad pagada para dicha factura.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR	Perteneciendo a uno y sólo un	PAGO	*
DEBE ESTAR	Pagando una y sólo una	FACTURA	*

Nombre : PARAMETRO_CUENTA

Nombre Corto : PC

Subtipo de :

Volumen Inicial : 13

Volumen Promedio:13

Volumen Máximo : 13

%Crecimiento Anual : 0

Descripción:

Contiene los valores generales del sistema como el valor actual del IVA, área de trabajo, cuentas contables especiales, etc.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE	CVEINT	N	INTEGER	6 *
DESCRIPCION	DESCLARGA	N	VARCHAR2	60
VALOR	DESCLARGA	N	VARCHAR2	60
CVE	Identificador único del PARAMETRO CUENTA.			
DESCRIPCION	Nombre completo del PARAMETRO CUENTA.			
VALOR	Valor actual de dicho PARAMETRO CUENTA.			

Nombre : PARTIDA

Nombre Corto : PAR

Subtipo de :

Volumen Inicial : 2000

Volumen Promedio: 12000

Volumen Máximo :30000

%Crecimiento Anual : 20

Descripción:

Cada uno de los conceptos que forman a una póliza. Estos detalles pueden ser de cargo o de abono, corresponden a una cuenta contable y contienen un concepto y un monto.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CONCEPTO		N	VARCHAR2	100
CONSEC	CVEINT	N	INTEGER	6 *
DEBEHABER		N	INTEGER	1
DEBE	MONTO	Y	MONEY	10,2
HABER	MONTO	Y	MONEY	10,2
CONCEPTO	Razón por la cual se genera la PARTIDA.			
CONSEC	Parte del identificador único, consecutivo de una misma póliza			
DEBEHABER	Indica si se trata de una PARTIDA de crédito o de abono.			
DEBE	Monto en caso de haber sido una PARTIDA de cargo.			
HABER	Monto en caso de haber sido una PARTIDA de abono.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR Afectando una y sólo una

SUBSUBCUENTA

DEBE ESTAR Formando una y sólo una

POLIZA *

Nombre : POLIZA

Nombre Corto : POL

Subtipo de

Volumen Inicial : 50

Volumen Promedio:300 Volumen

Máximo : 600

%Crecimiento Anual : 20

Descripción:

Cada uno de los movimientos contables que se generan en la empresa durante su ejercicio diario. Consta de partidas donde se detallan los conceptos que se utilizaron para formar la póliza contable.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
FECHA	FECHA	N	DATE	*
FOLIO	CVEINT	N	INTEGER	6 *
CONCEPTO	DESCLARGA	Y	VARCHAR2	60

FECHA Fecha cuando es registrada la POLIZA.
 FOLIO Consecutivo por mes de la POLIZA generada.
 CONCEPTO Razón por la cual se genera la PARTIDA.

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR	Formada por una o más	PARTIDA
DEBE ESTAR	Clasificada por uno y sólo un	TIPO POLIZA
DEBE ESTAR	Contenida en una y sólo una	AREA

Nombre : POLIZA MODELO

Nombre Corto : POM

Subtipo de : Volumen Inicial : 10 Volumen Promedio: 30 Volumen
 Máximo : 40 %Crecimiento Anual : 20

Descripción:

En las áreas de la empresa se generan pólizas específicas y repetitivas, por lo tanto, se construyeron meta pólizas o pólizas modelo por área, para que el usuario sólo llene los montos sin escoger las cuentas contables.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	4 *
DESCRIPCION	DECLARGA	N	VARCHAR2	60
CVE	Identificador único de la POLIZA MODELO.			
DESCRIPCION	Nombre completo de la POLIZA MODELO.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR	Detallada por una o más	POLIZA MODELO CUENTA
DEBE ESTAR	Clasificada por una y sólo una	AREA
DEBE ESTAR	Clasificada por uno y sólo un	TIPO POLIZA

Nombre : POLIZA MODELO CUENTA

Nombre Corto : PMC

Subtipo de : Volumen Inicial : 40 Volumen Promedio: 120 Volumen
 Máximo : 160 %Crecimiento Anual : 20

Descripción:

Para cada póliza modelo que se genera, finalmente se deberá indicar cuáles son las cuentas contables que debe afectar cada vez que se invocada. En una entidad hija se indican estas cuentas contables especificando si son de cargo y abono.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
DEBEHABER		N	INTEGER	1 *
DEBEHABER	Indica si la cuenta contable debe ser de cargo o de abono.			

Relaciones

Cada ocurrencia de esta Entidad :

DEBE ESTAR	Detallada por una y sólo una	SUBSUBCUENTA *
DEBE ESTAR	Detallando una y sólo una	POLIZA MODELO *

Nombre : SALA**Nombre Corto :** SAL

Subtipo de :

Volumen Inicial : 6

Volumen Promedio: 6 Volumen

Máximo : 10

%Crecimiento Anual : 0.1

Descripción

Cada uno de los centros de trabajo donde se elaboran las órdenes de trabajo y se facturan los servicios. Son los salones donde se encuentra el equipo de postproducción que se alquilan a los clientes.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE	CVEINT	N	INTEGER	6 *
DESC_CORTA	DESCORTA	N	VARCHAR2	15
CUENTA	DESCMEDIA	Y	CHAR	30
CUENTA_DEV	DESCMEDIA	Y	CHAR	30
DESC_LARGA	DESCLARGA	Y	VARCHAR2	60
CVE	Identificador único de la SALA.			
DESC_CORTA	Nombre corto de la SALA.			
CUENTA	Cuenta contable usada en la venta o facturación.			
CUENTA_DEV	Cuenta contable usada en las devoluciones o cancelaciones.			
DESC_LARGA	Nombre largo y completo de la SALA.			

Relaciones**Cada ocurrencia de esta Entidad :**

PUEDE ESTAR	Agrupando uno o más	COSTO_SALA_MES
PUEDE ESTAR	Clasificando uno o más	COSTO_BALANCE_MES
PUEDE ESTAR	Clasificando una o más	CUENTA_TIPO_COSTO
PUEDE ESTAR	Perteneciendo uno o más	COSTO_PARAMETRO_MES
PUEDE ESTAR	Perteneciendo uno o más	COSTO_SALA_MES
PUEDE ESTAR	Clasificando uno o más	COSTO_CUENTA_MES

Nombre : SECUENCIA**Nombre Corto :** SEC

Subtipo de :

Volumen Inicial : 1

Volumen Promedio: 1 Volumen

Máximo : 1

%Crecimiento Anual : 0

Descripción:

Esta entidad funciona como una secuencia convencional, donde se almacena el último número de una serie lineal y continua, que al momento de generar una nueva póliza, se calcula el siguiente folio con base en esta entidad.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CONSEC		N	INTEGER	*
CONSEC	Valor actual de la serie lineal y consecutiva.			

Relaciones**Cada ocurrencia de esta Entidad :**

PUEDE ESTAR	Numerando uno o más	DETALLE
-------------	---------------------	---------

Nombre : STATUS

Nombre Corto : STA

Subtipo de :

Volumen Inicial : 20

Volumen Promedio: 20 Volumen

Máximo : 20

%Crecimiento Anual : 0

Descripción -

Tabla donde se guardan cada uno de los diferentes estados en los que se pueden encontrar las FACTURAS y los PAGOS.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	6 *
DESC_CORTA		N	VARCHAR2	15
DESC_LARGA		Y	VARCHAR2	60
CVE	Identificador único del STATUS.			
DESC_CORTA	Nombre corto del STATUS.			
DESC_LARGA	Nombre largo y completo del STATUS.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR Clasificando una o más

FACTURA

PUEDE ESTAR Definiendo uno o más

PAGO

Nombre : SUBCUENTA

Nombre Corto : SCU

Subtipo de :

Volumen Inicial : 60

Volumen Promedio: 120 Volumen

Máximo : 400

%Crecimiento Anual : 10

Descripción:

De la clasificación general de las cuentas contables, se ramifica de tal forma que se llega a la subcuenta. otro nivel contable según el catálogo de cuentas que se maneje en la compañía para llevar el control de los movimientos diarios.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	2 *
DESCRIPCION	DESCLARGA	N	VARCHAR2	60
TIPO CUENTA		N	INTEGER	1
CVE	Identificado único de la SUBCUENTA.			
DESCRIPCION	Nombre completo de la SUBCUENTA.			
TIPO CUENTA	Indicador para saber si se trata de una cuenta detalle o totalizadora.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR Conteniendo una o más

SUBSUCUENTA

DEBE ESTAR Contenida en uno y sólo un

CENTRO COSTOS *

Nombre : SUBSUBCUENTA

Nombre Corto : SSC

Subtipo de :

Volumen Inicial : 120

Volumen Promedio: 240 Volumen

Máximo : 500

%Crecimiento Anual : 10

Descripción:

De la clasificación general de las cuentas contables, se ramifica de tal forma que se llega a la subcuenta y de ésta a la subsubcuenta, otro nivel contable según el catálogo de cuentas que se maneje en la compañía y con esto llevar el control de los movimientos diarios.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	2 *
DESCRIPCION	DESCLARGA	N	VARCHAR2	60
TIPO CUENTA		N	INTEGER	1
CVE	Identificado único de la SUBSUBCUENTA.			
DESCRIPCION	Nombre completo de la SUBSUBCUENTA.			
TIPO CUENTA	Indicador para saber si se trata de una cuenta detalle o totalizadora.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR	Afectada por una o más	PARTIDA
PUEDE ESTAR	Cerrada por uno o más	CIERRE CUENTA
PUEDE ESTAR	Conteniendo uno o más	DETALLE
PUEDE ESTAR	Detallando una o más	POLIZA MODELO CUENTA
DEBE ESTAR	Contenida en uno y sólo uno	SUBCUENTA *

Nombre : TIPO BALANCE

Nombre Corto : TIB

Subtipo de :

Volumen Inicial : 7

Volumen Promedio 7 Volumen

Máximo 7

%Crecimiento Anual : 0

Descripción:

Cada uno de los elementos necesarios para realizar el cálculo del Balance General Mensual

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE	CVEINT	N	INTEGER	6 *
DESC_LARGA	DESCLARGA	N	VARCHAR2	60
CVE	Identificador único del TIPO DE BALANCE.			
DESC_LARGA	Nombre largo y completo del TIPO DE BALANCE.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDE ESTAR	Clasificando uno o más	COSTO_BALANCE_MES
-------------	------------------------	-------------------

Nombre : TIPO COSTO

Nombre Corto : TCO

Subtipo de :

Volumen Inicial : 15

Volumen Promedio: 15 Volumen

Máximo : 15

%Crecimiento Anual : 0

Descripción:

Cada uno de los elementos necesarios para realizar el cálculo del Estado Financiero Mensual.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE	CVEINT	N	INTEGER	6 *
DESC_LARGA	DESCLARGA	N	VARCHAR2	60
CVE	Identificador único del TIPO COSTO.			
DESC_LARGA	Nombre largo y completo del TIPO COSTO.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDA ESTAR	Clasificando uno o más	COSTO_PARAMETRO_MES
PUEDA ESTAR	Clasificando uno o más	COSTO_SALA_MES

Nombre : TIPO POLIZA

Nombre Corto : TPO

Subtipo de :

Volumen Inicial : 2

Volumen Promedio: 3 Volumen

Máximo : 3

%Crecimiento Anual : 0

Descripción:

Cada una de las modalidades en que se puede encontrar una póliza y su respectivo reporte.

Generalmente se manejan dos: de diario y de cheque, aunque es posible que en algunas contabilidades se detallen algunas más, aunque son casos aislados y específicos.

Atributos

Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	INTEGER	3 *
CHEQUE		N	INTEGER	1
DESC CORTA	DESCORTA	N	VARCHAR2	15
DESC_LARGA	DESCLARGA	N	VARCHAR2	60
CVE	Identificador único del TIPO DE POLIZA.			
CHEQUE	Indica si este TIPO DE POLIZA genera un cheque o no.			
DESC CORTA	Nombre corto del TIPO DE POLIZA.			
DESC_LARGA	Nombre largo del TIPO DE POLIZA.			

Relaciones

Cada ocurrencia de esta Entidad :

PUEDA ESTAR	Clasificando una o más	POLIZA
PUEDA ESTAR	Clasificando una o más	POLIZA MODELO

Nombre : USUARIO

Nombre Corto : USU

Subtipo de :

Volumen Inicial : 10

Volumen Promedio: 20 Volumen

Máximo : 20

%Crecimiento Anual : 20

Descripción:

Cada una de las personas, clientes o empleados, que pueden acceder a los módulos de Contabilidad y de Cuentas por cobrar, detallando se grupo y con esto, sus privilegios funcionales dentro del sistema. Es necesaria también para llevar una bitácora por usuario de las operaciones que realice a la información del sistema.

Atributos

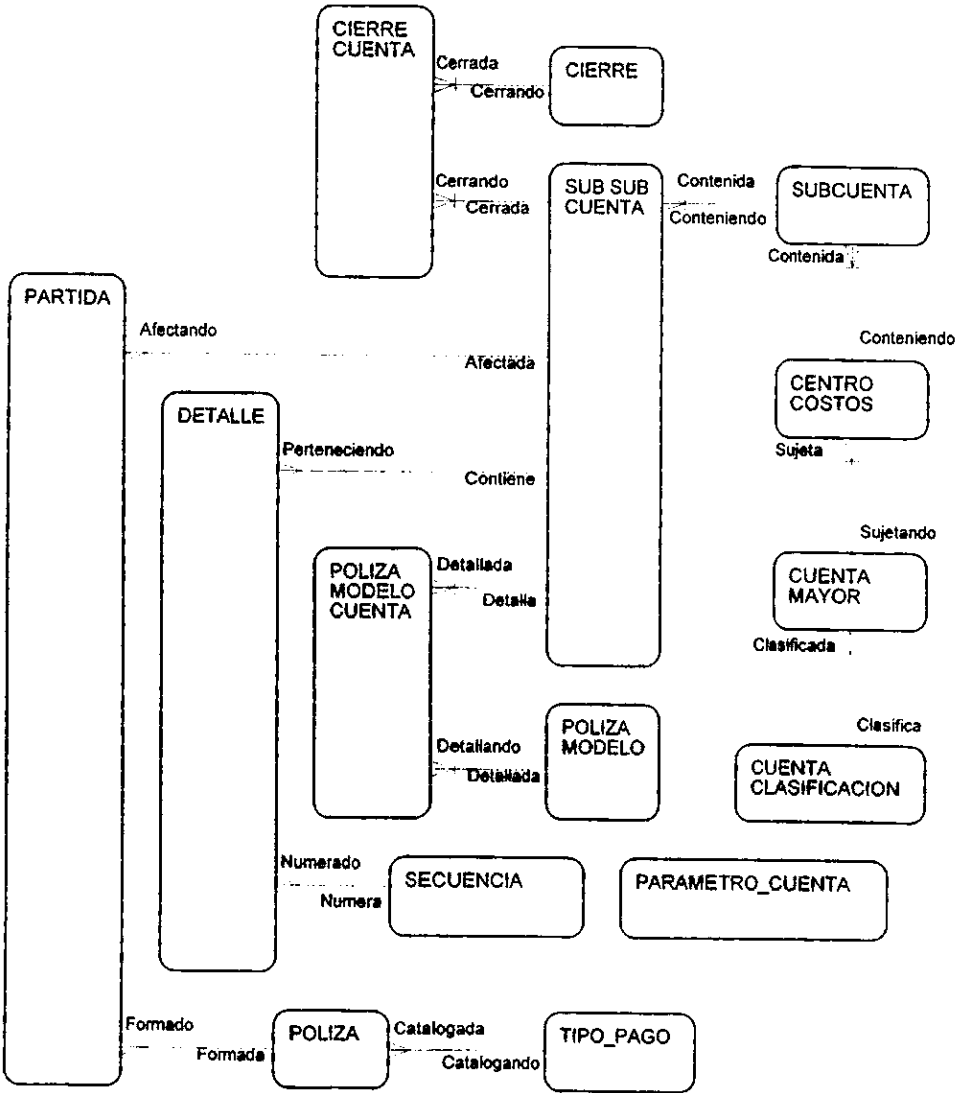
Nombre	Dominio	Opcional	Formato	Longitud
CVE		N	VARCHAR2	5 *

NOMBRE	N	VARCHAR2	60
PASSWORD	N	VARCHAR2	10
CVE	Identificador único del USUARIO		
NOMBRE	Nombre completo del USUARIO.		
PASSWORD	Contraseña personal para verificar la identidad del USUARIO.		

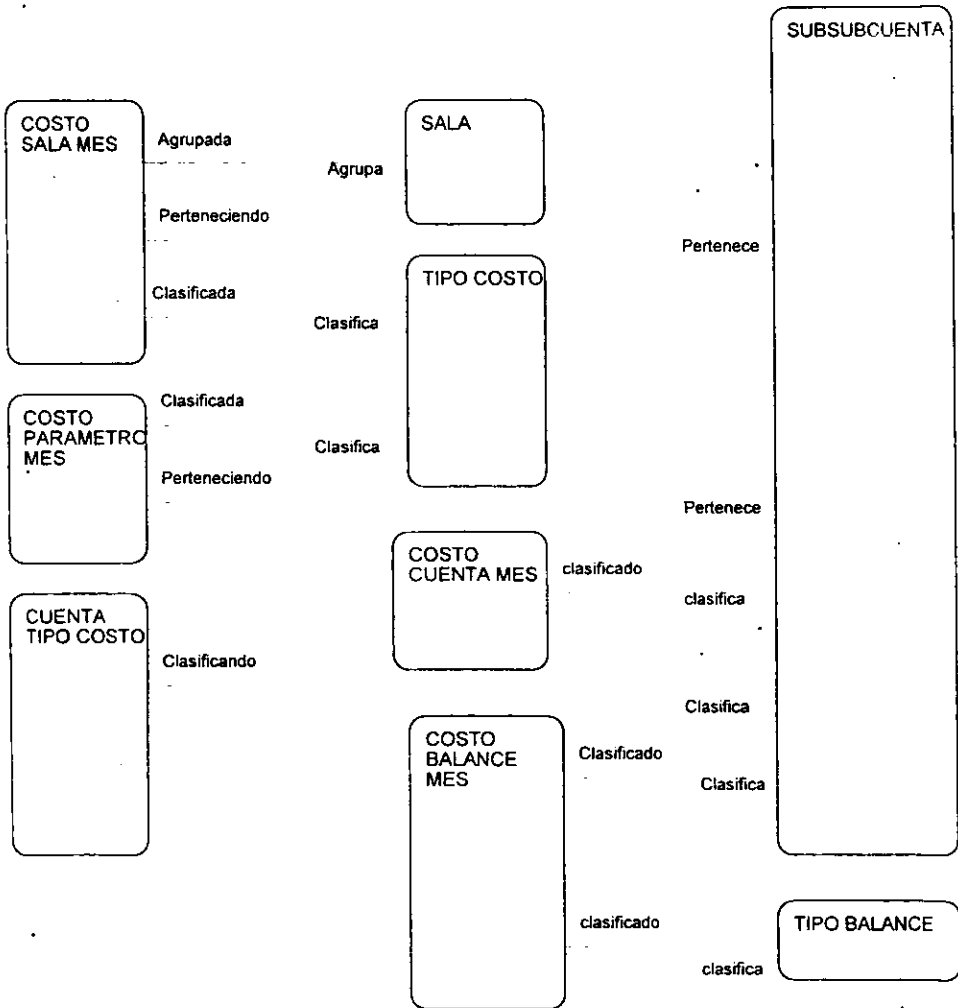
Relaciones

Cada ocurrencia de esta Entidad :

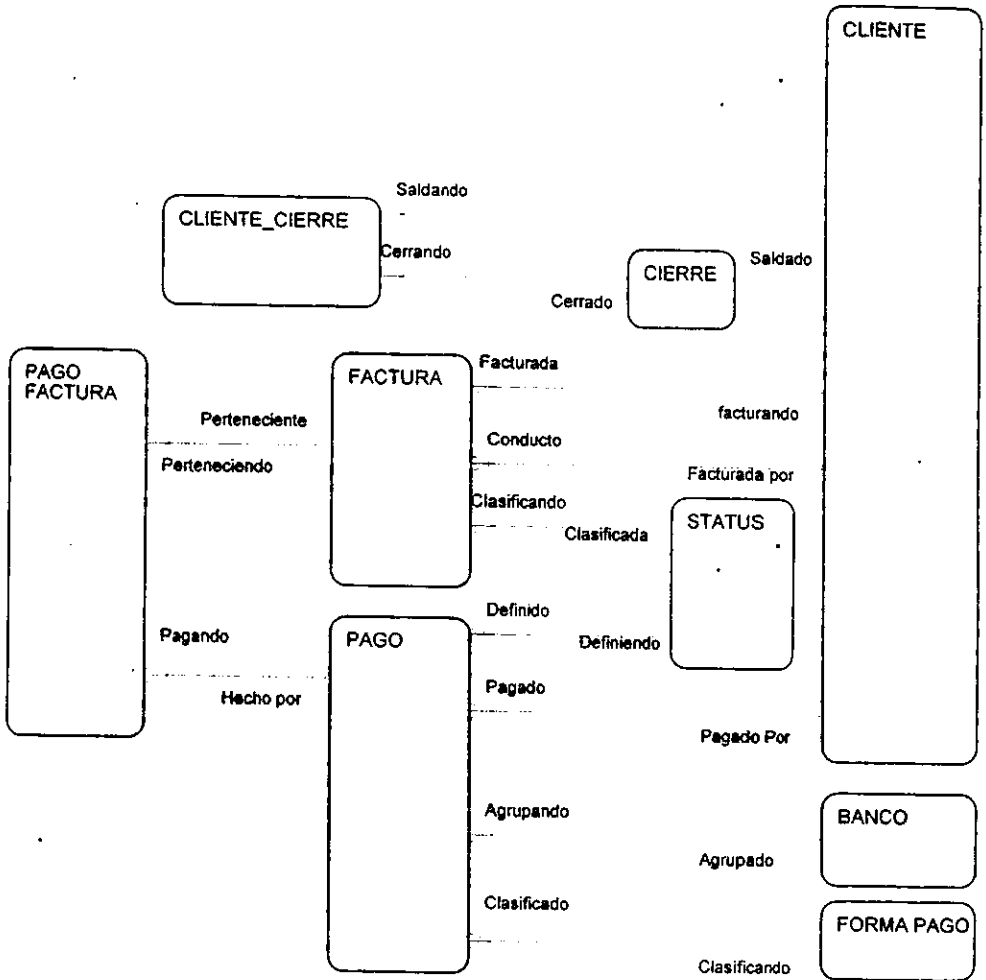
DEBE ESTAR	Contenido en uno y sólo un	GRUPO USUARIO
DEBE ESTAR	Perteneciendo a una y sólo una	AREA



Title : Fig. 3.2.3.A. Diagrama Entidad-Relación de Contabilidad (1a parte).



Title : Fig. 3.2.3.B. Diagrama Entidad-Relación (Estados Financieros).



Title : Fig. 3.2.3.C. Diagrama Entidad-Relación de Cuentas x Cobrar.

3.2.4. NORMALIZACIÓN.

Introducción.

Las diferentes etapas (progresivas) en este proceso se conocen como FORMAS NORMALES, cada una de las cuales es una refinación a su predecesora (Fig. 3.2.4.A Formas Normales):

1FN	PRIMERA FORMA NORMAL	E.F. Codd	Insiste en que los registros deben ser de la misma forma
2FN	SEGUNDA FORMA NORMAL	E.F. Codd	Remueve dependencias hacia parte de la llave (aplica a llaves primarias compuestas)
3FN	TERCERA FORMA NORMAL	E.F. Codd	Remueve dependencias a atributos no llave
BCFN	BOYCE/CODD FORMA NORMAL	E.F. Codd/ R.F. Boyce	Refinamiento tecnico a 3FN
4FN	CUARTA FORMA NORMAL	R. Fagin	Resuelve relaciones independientes
5FN	QUINTA FORMA NORMAL	R. Fagin	Clarifica relaciones dependientes

Fig. 3.2.4.A - Formas Normales.

Las formas normales no son parte del modelo relacional persé, constituyen una teoría separada sobre el modelo.

Cada Forma Normal, asume que su predecesora se debe cumplir.

Un modelo de datos en 3FN, debe estar también en 2FN y en 1FN.

Para efectos de ejemplificar el proceso de normalización, en este punto se tomarán una serie de datos crudos sobre los cuales se aplicará el proceso de normalización.

Pasos en la normalización de la base de datos.

FORMA	PASOS
1a forma	Cambiar todas las estructuras que no sean bidimensionales (es decir, grupos de repetición) en estructuras de registro bidimensionales.
2a forma	Eliminar los datos que no dependan totalmente de las llaves de registro.
3a forma	Eliminar los datos que dependan transitivamente de las llaves primarias.

Paso 0. Obtener datos en "0 Forma Normal" (0FN).

Numero	Nombre	Ciudad	Factura	\$Monto	Tipo Pago	No. factura	\$Pago
10	CINE S.A.	D.F.	A-1	5000	Cuentas x cobrar	A-1	\$550
					Facturación	A-1	\$450
20	El SIGLO	Monterrey	A-2	5000	Facturación		\$1000
30	JR PROD.	Puebla	A-3	9500	Facturación	A-3	\$4000
					Cuentas x cobrar	A-3	\$500
					Facturación	A-3	\$3000
					Facturación	A-3	\$2000

Tomara los grupos repetidos y convertirlos a una forma bidimensional.



Primera Forma Normal (1FN).

Cientes (1FN)

Numero	Nombre	Dirección	Delegación o Mpio.	FACTURA
10	CINE S.A.	Calle 23 Núm 45	Álvaro Obregón	A-1
20	El SIGLO	Insurgentes Sur 345	Álvaro Obregón	A-2
30	JR PROD.	Av. Centenario 16	Coyoacán	A-3

Facturas (1FN)

Factura	\$Monto	Fecha	Cliente
A-1	5000	15-Abr-97	10
A-2	5000	18-May-97	20
A-3	9500	3-Jun-97	30

Cientes - Pagos (1FN).

Cliente	Tipo Pago	No. factura	\$Pago
10	Cuentas x cobrar	A-1	\$550
	Facturación	A-1	\$450
20	Facturación		\$1000
30	Facturación	A-3	\$4000
	Cuentas x cobrar	A-3	\$500
	Facturación	A-3	\$3000
	Facturación	A-3	\$2000

Segunda Forma Normal. (2FN).

Cientes. Segunda Forma Normal (2FN).

Numero	Nombre	Dirección	Delegación o Mpio.
10	CINE S.A.	Calle 23 Núm 45	1
20	EI SIGLO	Insurgentes Sur 345	1
30	JR PROD.	Av. Centenario 16	2

Municipios. Segunda forma normal (2FN).

Cve	Nombre Delegación o Mpio.
1	Álvaro Obregón
2	Coyoacán
3	Gustavo A. Madero

Facturas (2FN)

Factura	Ciente	\$Monto	Fecha
A-1	10	5000	15-Abr-97
A-2	20	5000	18-May-97
A-3	30	9500	3-Jun-97

Cientes - Pagos (2FN).

Ciente	Tipo Pago	No. factura	\$Pago
10	1	A-1	\$550
	2	A-1	\$450
20	2		\$1000
30	2	A-3	\$4000
	1	A-3	\$500
	2	A-3	\$3000
	2	A-3	\$2000

Cientes - Pagos (2FN).

Ciente	No Pago	Tipo Pago	\$Pago
10	1	1	\$550
10	2	2	\$450
20	3	2	\$1000
30	4	2	\$4000
30	5	1	\$500
30	6	2	\$3000
30	7	2	\$2000

Tipos de Pagos (2FN).

Cve	Descripción
1	Cuentas x cobrar
2	Facturación
3	Poliza

Tercera Forma Normal.

Clientes. Segunda Forma Normal (3FN).

Numero	Nombre	Dirección	Delegación o Mpio.
10	CINE S.A.	Calle 23 Núm 45	1
20	EL SIGLO	Insurgentes Sur 345	1
30	JR PRÓD.	Av. Centenario 16	2

Municipios. Segunda forma normal (3FN).

Cve	Nombre Delegación o Mpio.
1	Álvaro Obregón
2	Coyoacán
3	Gustavo A. Madero

Facturas (3FN)

Factura	Cliente	\$Monto	Fecha
A-1	10	5000	15-Abr-97
A-2	20	5000	18-May-97
A-3	30	9500	3-Jun-97

Clientes - Pagos (3FN).

Cliente	No Pago	Tipo Pago	\$Pago
10	1	1	\$550
10	2	2	\$450
20	3	2	\$1000
30	4	2	\$4000
30	5	1	\$500
30	6	2	\$3000
30	7	2	\$2000

Pago Factura (3FN).

No Pago	Factura	\$Pago
1	A-1	\$550
2	A-1	\$450
4	A-3	\$4000
5	A-3	\$500
6	A-3	\$3000
7	A-3	\$2000

En términos de presentación en Diagrama Entidad Relación, obtenemos:

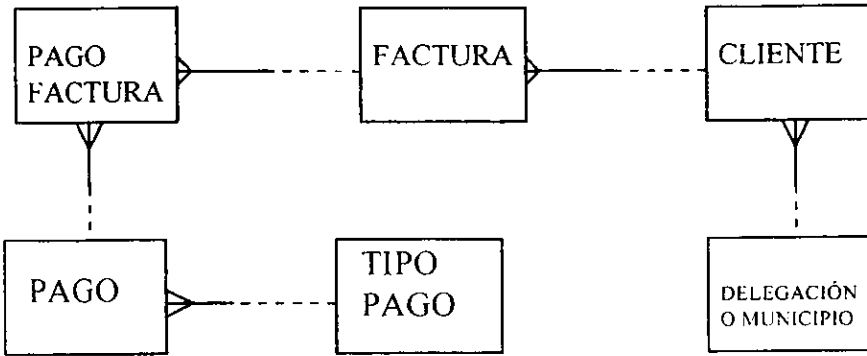


Fig. 3.2.4.B. Modelo Entidad Relación después del proceso de Normalización.

Table: cliente

Page: 1

Properties

Date Created:	26/06/97 11:56:44	Def. Updatable:	True
Last Updated:	12/06/97 17:29:17	Order By On:	True
Order By:	cliente.ci_cve	Record Count:	59

Columns

Name	Type	Size
ci_cve	Number (Integer)	2
ci_rfc	Text	14
ci_razon_social	Text	60
ci_nombre	Text	15
ci_direccion	Text	60
ci_colonia	Text	60
ci_delegacion	Text	60
ci_ent_cve	Number (Integer)	2
ci_cp	Text	5
ci_telefono	Text	60
ci_fax	Text	60
ci_clasificacion	Text	60
ci_dia_pago	Number (Integer)	2
ci_limite_credito	Number (Double)	8
ci_dias_credito	Number (Double)	8
ci_descuento	Number (Double)	8
ci_referencia	Text	60
ci_dia_revision	Number (Integer)	2
ci_saldo_real	Number (Double)	8
ci_saldo_pendiente	Number (Double)	8
ci_saldo_anticipado	Number (Double)	8
ci_fecha_ultima_venta	Date/Time	8

Table Indexes

Name	Number of Fields
ci_cve	1
Fields:	ci_cve, Ascending
ci_ent_cve	1
Fields:	ci_ent_cve, Ascending
PrimaryKey	1
Fields:	ci_cve, Ascending

Sábado 21 de Junio de 1997

Table: factura

Page: 1

Properties

Data Created:	26/06/97 11:57:18	Def. Updatable:	True
Last Updated:	12/06/97 17:27:49	Order By On:	False
Record Count:	431		

Columns

Name	Type	Size
fac_serie	Text	1
fac_num	Number (Integer)	2
fac_fecha	Date/Time	8
fac_cli_cve	Number (Integer)	2
fac_sta_cve	Number (Integer)	2
fac_observacion	Text	60
fac_cli_conduc	Number (Integer)	2
fac_iva	Number (Double)	8
fac_subtotal	Number (Double)	8
fac_totaliva	Number (Double)	8
fac_total	Number (Double)	8
fac_tmo_cve	Number (Integer)	2
fac_tipo_cambio	Number (Double)	8
fac_fecha_revision	Date/Time	8
fac_fecha_vencimiento	Date/Time	8
fac_fecha_pago	Date/Time	8
fac_pago_pendiente	Number (Double)	8
fac_fecha_prob_pago	Date/Time	8

Table Indexes

Name	Number of Fields
FAC_CLI_FK_I	1
Fields:	fac_cli_cve, Ascending
FAC_Null	7
Fields:	fac_serie, Ascending fac_num, Ascending fac_fecha, Ascending fac_cli_cve, Ascending fac_sta_cve, Ascending fac_iva, Ascending fac_tmo_cve, Ascending
FAC_PK	2
Fields:	fac_serie, Ascending fac_num, Ascending
FAC_STA_FK_I	1
Fields:	fac_sta_cve, Ascending
FAC_TMO_FK_I	1

Table: factura

Sábado 21 de Junio de 1997

Page: 2

Fields:

fac_tmo_cve, Ascending

*

Table: pago

Sábado 21 de Junio de 1997

Page: 3

Properties

Date Created:	25/06/97 12:02:45	Def. Updatable:	True
Last Updated:	12/06/97 16:51:52	Order By On:	False
Record Count:	115		

Columns

Name	Type	Size
pag_cve	Number (Integer)	2
pag_fecha	Date/Time	8
pag_fop_cve	Number (Integer)	2
pag_cli_cve	Number (Integer)	2
pag_sta_cve	Number (Integer)	2
pag_monto	Number (Double)	8
pag_antipico	Number (Double)	8
pag_num_docto	Text	20
pag_observaciones	Text	60
pag_ban_cve	Number (Integer)	2

Table Indexes

Name	Number of Fields
PAG_Null	7
Fields:	pag_cve, Ascending pag_fecha, Ascending pag_fop_cve, Ascending pag_cli_cve, Ascending pag_sta_cve, Ascending pag_monto, Ascending pag_antipico, Ascending pag_num_docto, Ascending

Table: pago_factura

Sábado 21 de Junio de 1997

Page: 4

Properties

Date Created:	26/06/97 12:02:52	Def. Updatable:	True
Last Updated:	12/06/97 16:51:52	Order By On:	True
Record Count:	131		

Columns

Name	Type	Size
pag_pag_cve	Number (Integer)	2
pag_fac_serie	Text	1
pag_fac_num	Number (Integer)	2
pag_monto	Number (Double)	8

Table Indexes

Name	Number of Fields
PAF_Null	4
Fields:	pag_pag_cve, Ascending pag_fac_serie, Ascending pag_fac_num, Ascending pag_monto, Ascending

Table: tipo_pago

Sábado 21 de Junio de 1997

Page: 5

Properties

Date Created:	26/06/97 12:06:50	Def. Updatable:	True
Last Updated:	12/06/97 16:52:54	Order By On:	False
Record Count:	6		

Columns

Name	Type	Size
tip_cve	Number (Integer)	2
tip_descripcion	Text	60

Table Indexes

Name	Number of Fields
TIP_Null	2
Fields:	tip_cve, Ascending tip_descripcion, Ascending
TIP_PK	1
Fields:	tip_cve, Ascending

3.2.5. Matriz Entidad vs. Funciones.

A continuación presentamos la representación matricial de las relaciones que existen entre todas las funciones de los módulos analizados (funciones) contra las entidades que existen en nuestra empresa. Con este reporte podemos revisar si existen entidades que no sean usadas por ninguna función o bien, funciones elementales que no accedan a la información que ofrecen las entidades.

Las funciones ejercen su acción sobre la información contenida en las entidades de la siguiente manera:

- C: La función permite la creación de nuevos registros dentro de la entidad.
- N: La función permite la consulta de la información de la entidad.
- A: La función permite la actualización o modificación de la información guardada en la entidad.
- B: La función permite el borrado de registros de la entidad.

Toda la información de este documento son resultado a su vez, de los reportes entregados por la herramienta de CASE.

	F001	F002	F003	F004	F005	F006	F007	F008	F009	F010
AREA						CNAB		N		
BANCO										
CENTRO COSTOS			CNAB							
CIERRE										
CIERRE CUENTA										
CLASIFICACION CUENTA	CNAB									
CLIENTE										
CLIENTE CIERRE										
COSTO BALANCE MES										
COSTO CUENTA MES										
COSTO PARAMETRO MES										
COSTO SALA MES										
CUENTA CLASIFICACION	CNAB									
CUENTA MAYOR		CNAB								
ENTIDAD										
FACTURA										
GRUPO USUARIO										
PAGO										
PAGO FACTURA										
PARAMETRO_CUENTAS										
PARTIDA										CN
POLIZA										CN
POLIZA MODELO								CNAB		
POLIZA MODELO CUENTA									CNAB	
SALA										
STATUS										
SUBCUENTA				CNAB						
SUBSUBCUENTA					CNAB				N	N
TIPO COSTO										
TIPO BALANCE										
TIPO PAGO										
TIPO POLIZA							CNAB			N
USUARIO										

	F011	F012	F013	F014	F015	F016	F017	F018	F019	FC01
AREA	N				N					
BANCO										
CENTRO COSTOS										
CIERRE			CN							
CIERRE CUENTA			CN							
CLASIFICACION CUENTA										
CLIENTE			N							
CLIENTE CIERRE			CN							CNAB
COSTO BALANCE MES										
COSTO CUENTA MES									CNAB	
COSTO PARAMETRO MES										
COSTO SALA MES								CNAB		
CUENTA CLASIFICACION								CNAB		
CUENTA MAYOR										
ENTIDAD										
FACTURA										N
GRUPO USUARIO				CNAB	N					
PAGO										
PAGO FACTURA										
PARAMETRO CUENTAS						CNA				
PARTIDA	CN	CN	N							
POLIZA	CN	CN								
POLIZA MODELO	N									
POLIZA MODELO CUENTA										
SALA							CA	N		
STATUS										
SUBCUENTA										
SUBSUBCUENTA	N	N	N				N	N		
TIPO COSTO								N		
TIPO BALANCE									N	
TIPO PAGO										N
TIPO POLIZA	N	N				N				
USUARIO					CNAB					

	FC02	FC03	FC04	FC05	FC06	FC07	FC08	FC09	FC10	FC11
ÁREA										
BANCO	CNAB	N	N		N	N	N			N
CENTRO COSTOS										
CIERRE										
CIERRE CUENTA										
CLASIFICACION CUENTA										
CLIENTE		N	N	CA	N	NA	NA	N	N	N
CLIENTE CIERRE										
COSTO BALANCE MES										
COSTO CUENTA MES										
COSTO PARAMETRO MES										
COSTO SALA MES										
CUENTA CLASIFICACION										
CUENTA MAYOR										
ENTIDAD										
FACTURA		N	N	CA	N			N	N	
GRUPO USUARIO										
PAGO		C	N		C	NA	NA			N
PAGO FACTURA		C	C		C	B	B			N
PARAMETRO CUENTAS				N						
PARTIDA				C						
POLIZA				C						
POLIZA MODELO										
POLIZA MODELO CUENTA										
SALA										
STATUS				N		N	N			
SUBCUENTA										
SUBSUBCUENTA										
TIPO COSTO										
TIPO BALANCE										
TIPO POLIZA										
TIPO PAGO		N	N		N	N	N			
USUARIO										

	R001	R002	R003	R004	R005	R006	R007	R008	R009	
AREA										
BANCO										
CENTRO COSTOS										
CIERRE										
CIERRE CUENTA	N	N	N	N		N			N	
CLASIFICACION CUENTA										
CLIENTE										
CLIENTE CIERRE										
COSTO BALANCE MES								N		
COSTO CUENTA MES							N			
COSTO PARAMETRO MES										
COSTO SALA MES							N			
CUENTA CLASIFICACION										
CUENTA MAYOR										
ENTIDAD										
FACTURA										
GRUPO USUARIO										
PAGO										
PAGO FACTURA										
PARAMETRO CUENTAS							N			
PARTIDA	N	N	N	N	N	N			N	
POLIZA	N	N								
POLIZA MODELO										
POLIZA MODELO CUENTA										
SALA							N			
STATUS										
SUBCUENTA										
SUBSUBCUENTA			N	N	N	N	N	N	N	
TIPO COSTO							N			
TIPO BALANCE								N		
TIPO POLIZA	N	N								
TIPO PAGO										
USUARIO										

	RC01	RC02	RC03	RC04	RC05	RC06	RC07						
AREA													
BANCO	N												
CENTRO COSTOS													
CIERRE													
CIERRE CUENTA													
CLASIFICACION CUENTA													
CLIENTE	N	N	N	N	N	N	N						
CLIENTE CIERRE		N											
COSTO BALANCE MES													
COSTO CUENTA MES													
COSTO PARAMETRO MES													
COSTO SALA MES													
CUENTA CLASIFICACION													
CUENTA MAYOR													
ENTIDAD													
FACTURA		N	N	N	N	N							
GRUPO USUARIO													
PAGO	N	N											
PAGO FACTURA													
PARAMETRO CUENTAS													
PARTIDA													
POLIZA													
POLIZA MODELO													
POLIZA MODELO CUENTA													
SALA													
STATUS		N	N	N	N	N							
SUBCUENTA													
SUBSUBCUENTA													
TIPO COSTO													
TIPO BALANCE													
TIPO POLIZA													
TIPO PAGO	N	N											
USUARIO													

3.2.6. Diccionario de datos.

A continuación presentamos la descripción de la BD finalmente como queda después de ser mapeada y creadas las tablas con sus respectivos índices y reglas del negocio.

Table: area

Lunes 14 de Julio de 1997

Page: 1

Properties

Date Created:	26/06/97 11:56:18	Def. Updatable:	True
Last Updated:	12/06/97 16:50:59	Record Count:	2

Columns

Name	Type	Size
are_cve	Number (Integer)	2
are_desc_corta	Text	15
are_desc_larga	Text	60

Table Indexes

Name	Number of Fields
ARE_Null	3
Fields:	are_cve, Ascending are_desc_corta, Ascending are_desc_larga, Ascending
ARE_PK	1
Fields:	are_cve, Ascending

Table: banco

Page: 2

Properties

Date Created: 26/06/97 11:56:25 Def. Updatable: True
Last Updated: 12/06/97 16:50:59 Record Count: 1

Columns

Name	Type	Size
ban_cve	Number (Integer)	2
ban_nombre	Text	15
ban_direccion	Text	60
ban_responsable	Text	15
ban_telefono	Text	15

Table indexes

Name	Number of Fields
BAN_Null	3
Fields:	ban_cve, Ascending ban_nombre, Ascending ban_direccion, Ascending
BAN_PK	1
Fields:	ban_cve, Ascending

Lunes 14 de Julio de 1997

Page: 3

Table: centro_de_costos

Properties

Date Created:	30/06/97 10:34:36	Def. Updatable:	True
Last Updated:	25/06/97 14:51:00	Order By On:	False
Record Count:	21		

Columns

Name	Type	Size
cco_cma_ccl_cve	Number (Integer)	2
cco_cma_cve	Number (Integer)	2
cco_cve	Number (Integer)	2
cco_tipo_cuenta	Number (Integer)	2
cco_descripcion	Text	60

Table Indexes

Name	Number of Fields
CCO_CMA_FK_I Fields:	2 cco_cma_ccl_cve, Ascending cco_cma_cve, Ascending
CCO_Null Fields:	5 cco_cma_ccl_cve, Ascending cco_cma_cve, Ascending cco_cve, Ascending cco_tipo_cuenta, Ascending cco_descripcion, Ascending
CCO_PK Fields:	3 cco_cma_ccl_cve, Ascending cco_cma_cve, Ascending cco_cve, Ascending
CUENTA_MAYORCENTRO_DE_COSTO Fields:	2 cco_cma_ccl_cve, Ascending cco_cma_cve, Ascending

Table: cierre

Lunes 14 de Julio de 1997

Page: 4

Properties

Date Created: 26/06/97 11:56:31 Def. Updatable: True
Last Updated: 12/06/97 16:50:59 Record Count: 1

Columns

Name	Type	Size
cie_fecha	Date/Time	8
cie_cerrado	Number (Integer)	2

Table Indexes

Name	Number of Fields
CIE_Null	2
Fields:	cie_fecha, Ascending cie_cerrado, Ascending
CIE_PK	1
Fields:	cie_fecha, Ascending

Lunes 14 de Julio de 1997

Table: cierre_cuenta

Page: 5

Properties

Date Created:	26/06/97 11:56:38	Def. Updatable:	True
Last Updated:	12/06/97 16:51:00	Order By On:	False
Record Count:	29		

Columns

Name	Type	Size
cic_cie_fecha	Date/Time	8
cic_ssc_scu_cco_cma_ccl_cve	Number (Integer)	2
cic_ssc_scu_cco_cma_cve	Number (Integer)	2
cic_ssc_scu_cco_cve	Number (Integer)	2
cic_ssc_scu_cve	Number (Integer)	2
cic_ssc_cve	Number (Integer)	2
cic_saldo_cierre	Number (Double)	8

Table Indexes

Name	Number of Fields
CIC_CIE_FK_I Fields:	1 cic_cie_fecha, Ascending
CIC_Null Fields:	7 cic_ssc_scu_cco_cma_ccl_cve, Ascending cic_ssc_scu_cco_cma_cve, Ascending cic_ssc_scu_cco_cve, Ascending cic_ssc_scu_cve, Ascending cic_ssc_cve, Ascending cic_cie_fecha, Ascending cic_saldo_cierre, Ascending
CIC_PK Fields:	6 cic_cie_fecha, Ascending cic_ssc_scu_cco_cma_ccl_cve, Ascending cic_ssc_scu_cco_cma_cve, Ascending cic_ssc_scu_cco_cve, Ascending cic_ssc_scu_cve, Ascending cic_ssc_cve, Ascending
CIC_SSC_FK_I Fields:	5 cic_ssc_scu_cco_cma_ccl_cve, Ascending cic_ssc_scu_cco_cma_cve, Ascending cic_ssc_scu_cco_cve, Ascending cic_ssc_scu_cve, Ascending cic_ssc_cve, Ascending

Table: cliente

Page: 6

Properties

Date Created:	26/06/97 11:56:44	Def. Updatable:	True
Last Updated:	24/06/97 12:10:12	Order By On:	True
Order By:	cliente.cli_cve	Record Count:	60

Columns

Name	Type	Size
cli_cve	Number (Integer)	2
cli_rfc	Text	14
cli_razon_social	Text	60
cli_nombre	Text	15
cli_direccion	Text	60
cli_colonia	Text	60
cli_delegacion	Text	60
cli_ent_cve	Number (Integer)	2
cli_cp	Text	5
cli_telefono	Text	60
cli_fax	Text	60
cli_clasificacion	Text	60
cli_dia_pago	Number (Integer)	2
cli_limite_credito	Number (Double)	8
cli_dias_credito	Number (Double)	8
cli_descuento	Number (Double)	8
cli_referencia	Text	60
cli_dia_revision	Number (Integer)	2
cli_saldo_real	Number (Double)	8
cli_saldo_penaliente	Number (Double)	8
cli_saldo_anticipado	Number (Double)	8
cli_fecha_ultima_venta	Date/Time	8

Table indexes

Name	Number of Fields
cli_cve	1
Fields:	cli_cve, Ascending
cli_ent_cve	1
Fields:	cli_ent_cve, Ascending
PrimaryKey	1
Fields:	cli_cve, Ascending

Lunes 14 de Julio de 1997

Table: cliente_cierre

Page: 7

Properties

Date Created:	26/06/97 11:56:53	Def. Updatable:	True
Last Updated:	12/06/97 16:51:00	Record Count:	22

Columns

Name	Type	Size
clc_cli_cve	Number (Integer)	2
clc_cie_fecha	Date/Time	8
clc_saldo	Number (Double)	8

Table Indexes

Name	Number of Fields
CIERRECLIENTE_CIERRE	1
Fields:	clc_cie_fecha, Ascending
CLC_Null	3
Fields:	clc_cli_cve, Ascending clc_cie_fecha, Ascending clc_saldo, Ascending

Table: costo_balance_mes

Properties

Date Created:	10/06/97 11:25:03	Def. Updatable:	True
Last Updated:	12/06/97 16:50:59	Order By On:	False
Record Count:	0		

Columns

Name	Type	Size
cbm_ssc_scu_cco_cma_cci_cve	Number (Integer)	2
cbm_ssc_scu_cco_cma_cve	Number (Integer)	2
cbm_ssc_scu_cco_cve	Number (Integer)	2
cbm_ssc_scu_cve	Number (Integer)	2
cbm_ssc_cve	Number (Integer)	2
cbm_tib_cve	Number (Integer)	2
cbm_fecha	Date/Time	8

Table Indexes

Name	Number of Fields
PrimaryKey	7
Fields:	cbm_ssc_scu_cco_cma_cci_cve, Ascending
	cbm_ssc_scu_cco_cma_cve, Ascending
	cbm_ssc_scu_cco_cve, Ascending
	cbm_ssc_scu_cve, Ascending
	cbm_ssc_cve, Ascending
	cbm_tib_cve, Ascending
	cbm_fecha, Ascending

Table: costo_cuenta_mes

Lunes 14 de Julio de 1997

Page: 9

Properties

Date Created:	06/06/97 14:39:59	Def. Updatable:	True
Last Updated:	12/06/97 16:50:59	Order By On:	True
Order By:	costo_cuenta_mes.ccm_tco_cve	Record Count:	3

Columns

Name	Type	Size
ccm_ssc_scu_cco_cma_ccl_cve	Number (Integer)	2
ccm_ssc_scu_cco_cma_cve	Number (Integer)	2
ccm_ssc_scu_cco_cve	Number (Integer)	2
ccm_ssc_scu_cve	Number (Integer)	2
ccm_ssc_cve	Number (Integer)	2
ccm_tco_cve	Number (Integer)	2
ccm_fecha	Date/Time	8
ccm_presupuesto	Number (Double)	8

Table Indexes

Name	Number of Fields
CCM_PK	7
Fields:	ccm_fecha, Ascending ccm_ssc_cve, Ascending ccm_ssc_scu_cve, Ascending ccm_ssc_scu_cco_cve, Ascending ccm_ssc_scu_cco_cma_cve, Ascending ccm_ssc_scu_cco_cma_ccl_cve, Ascending ccm_tco_cve, Ascending
CCM_SSC_FK	5
Fields:	ccm_ssc_scu_cco_cma_ccl_cve, Ascending ccm_ssc_scu_cco_cma_cve, Ascending ccm_ssc_scu_cco_cve, Ascending ccm_ssc_scu_cve, Ascending ccm_ssc_cve, Ascending
CCM_SSC_FK_!	5
Fields:	ccm_ssc_scu_cco_cma_ccl_cve, Ascending ccm_ssc_scu_cco_cma_cve, Ascending ccm_ssc_scu_cco_cve, Ascending ccm_ssc_scu_cve, Ascending ccm_ssc_cve, Ascending
CCM_TCO_FK	1
Fields:	ccm_tco_cve, Ascending
ccm_Null	6

Table: costo_cuenta_mes

Fields:

ccm_ssc_scu_cco_cma_ccl_cve, Ascending
ccm_ssc_scu_cco_cma_cve, Ascending
ccm_ssc_scu_cco_cve, Ascending
ccm_ssc_scu_cve, Ascending
ccm_ssc_cve, Ascending
ccm_fecha, Ascending

CTC_TCO_FK_I

Fields:

1
ccm_tco_cve, Ascending

Lunes 14 de Julio de 1997

Table: costo_parametro_mes

Page: 11

Properties

Date Created: 06/06/97 14:39:58 Def. Updatable: True
 Last Updated: 12/06/97 16:50:59 Order By On: True
 Order By: costo_parametro_mes.cpm_tco_c Record Count: 5
 ve

Columns

Name	Type	Size
cpm_ssc_scu_cco_cma_cci_cve	Number (Integer)	2
cpm_ssc_scu_cco_cma_cve	Number (Integer)	2
cpm_ssc_scu_cco_cve	Number (Integer)	2
cpm_ssc_scu_cve	Number (Integer)	2
cpm_ssc_cve	Number (Integer)	2
cpm_par_cve	Number (Integer)	2
cpm_tco_cve	Number (Integer)	2
cpm_fecha	Date/Time	8
cpm_presupuesto	Number (Double)	8

Table Indexes

Name	Number of Fields
cpm_Null	8
Fields:	cpm_ssc_scu_cco_cma_cci_cve, Ascending cpm_ssc_scu_cco_cma_cve, Ascending cpm_ssc_scu_cco_cve, Ascending cpm_ssc_scu_cve, Ascending cpm_par_cve, Ascending cpm_ssc_cve, Ascending cpm_tco_cve, Ascending cpm_fecha, Ascending
CPM_PAR_FK	1
Fields:	cpm_par_cve, Ascending
CPM_PAR_FK_I	1
Fields:	cpm_par_cve, Ascending
CPM_PK	8
Fields:	cpm_fecha, Ascending cpm_ssc_cve, Ascending cpm_ssc_scu_cve, Ascending cpm_ssc_scu_cco_cve, Ascending cpm_ssc_scu_cco_cma_cve, Ascending cpm_ssc_scu_cco_cma_cci_cve, Ascending cpm_par_cve, Ascending cpm_tco_cve, Ascending
CPM_SSC_FK	5

Table: costo_parametro_mes

Fields:	cpm_ssc_scu_cco_cma_ccd_cve, Ascending
	cpm_ssc_scu_cco_cma_cve, Ascending
	cpm_ssc_scu_cco_cve, Ascending
	cpm_ssc_scu_cve, Ascending
	cpm_ssc_cve, Ascending
CPM_SSC_FK_I	5
Fields:	cpm_ssc_scu_cco_cma_ccd_cve, Ascending
	cpm_ssc_scu_cco_cma_cve, Ascending
	cpm_ssc_scu_cco_cve, Ascending
	cpm_ssc_scu_cve, Ascending
	cpm_ssc_cve, Ascending
CPM_TCO_FK	1
Fields:	cpm_tco_cve, Ascending
CPM_TCO_FK_I	1
Fields:	cpm_tco_cve, Ascending

Table: costo_sala_mes

Lunes 14 de Julio de 1997

Page: 13

Properties

Date Created:	06/06/97 14:39:58	Def. Updatable:	True
Last Updated:	12/06/97 16:50:59	Order By On:	True
Order By:	costo_sala_mes.csm_tco_cve	Record Count:	8

Columns

Name	Type	Size
csm_ssc_scu_cco_cma_ccl_cve	Number (Integer)	2
csm_ssc_scu_cco_cma_cve	Number (Integer)	2
csm_ssc_scu_cco_cve	Number (Integer)	2
csm_ssc_scu_cve	Number (Integer)	2
csm_ssc_cve	Number (Integer)	2
csm_sal_cve	Number (Integer)	2
csm_tco_cve	Number (Integer)	2
csm_fecha	Date/Time	8
csm_presupuesto	Number (Double)	8

Table Indexes

Name	Number of Fields
csm_Null	7
Fields:	csm_ssc_scu_cco_cma_ccl_cve, Ascending csm_ssc_scu_cco_cma_cve, Ascending csm_ssc_scu_cco_cve, Ascending csm_ssc_scu_cve, Ascending csm_ssc_cve, Ascending csm_sal_cve, Ascending csm_fecha, Ascending
CSM_PK	8
Fields:	csm_fecha, Ascending csm_ssc_cve, Ascending csm_ssc_scu_cve, Ascending csm_ssc_scu_cco_cve, Ascending csm_ssc_scu_cco_cma_cve, Ascending csm_ssc_scu_cco_cma_ccl_cve, Ascending csm_sal_cve, Ascending csm_tco_cve, Ascending
CSM_SAL_FK	1
Fields:	csm_sal_cve, Ascending
CSM_SAL_FK_I	1
Fields:	csm_sal_cve, Ascending
CSM_SSC_FK_I	5

Table: costo_sala_mes

Fields:	csm_ssc_scu_cco_cma_ccl_cve, Ascending csm_ssc_scu_cco_cma_cve, Ascending csm_ssc_scu_cco_cve, Ascending csm_ssc_scu_cve, Ascending csm_ssc_cve, Ascending
CSM_SSC_TPO_FK	5
Fields:	csm_ssc_scu_cco_cma_ccl_cve, Ascending csm_ssc_scu_cco_cma_cve, Ascending csm_ssc_scu_cco_cve, Ascending csm_ssc_scu_cve, Ascending csm_ssc_cve, Ascending
CSM_TCO_FK	1
Fields:	csm_tco_cve, Ascending
CSM_TCO_FK_I	1
Fields:	csm_tco_cve, Ascending

Lunes 14 de Julio de 1997

Page: 15

Table: cuenta

Properties

Date Created:	13/06/97 13:47:19	Def. Updatable:	True
Last Updated:	12/06/97 16:50:59	Order By On:	False
Record Count:	5		

Columns

Name	Type	Size
cue_cve	Number (Integer)	2
cue_descripcion	Text	60

Table Indexes

Name	Number of Fields
CUE_Null	2
Fields:	cue_cve, Ascending cue_descripcion, Ascending
CUE_PK	1
Fields:	cue_cve, Ascending

Table: cuenta_clasificacion

Properties

Date Created: 13/06/97 13:47:19
Last Updated: 26/06/97 17:46:38
Record Count: 7

Def. Updatable: True
Order By On: False

Columns

Name	Type	Size
ccl_cve	Number (Integer)	2
ccl_tipo_cuenta	Number (Integer)	2
ccl_descripcion	Text	60

Table Indexes

Name	Number of Fields
CCL_Null	3
Fields:	ccl_cve, Ascending ccl_tipo_cuenta, Ascending ccl_descripcion, Ascending
CCL_PK	1
Fields:	ccl_cve, Ascending

Lunes 14 de Julio de 1997

Page: 17

Table: cuenta_mayor

Properties

Date Created:	13/06/97 13:47:19	Def. Updatable:	True
Last Updated:	25/06/97 14:40:09	Order By On:	False
Record Count:	17		

Columns

Name	Type	Size
cma_cci_cve	Number (Integer)	2
cma_cve	Number (Integer)	2
cma_tipo_cuenta	Number (Integer)	2
cma_descripcion	Text	60

Table Indexes

Name	Number of Fields
CMA_CCL_FK_I	1
Fields:	cma_cci_cve, Ascending
CMA_Null	4
Fields:	cma_cci_cve, Ascending cma_cve, Ascending cma_tipo_cuenta, Ascending cma_descripcion, Ascending
CMA_PK	2
Fields:	cma_cci_cve, Ascending cma_cve, Ascending
CUENTA_CLASIFICACIONCUENTA_MA	1
Fields:	cma_cci_cve, Ascending

Table: detalle

Properties

Date Created: 13/06/97 13:47:20 Def. Updatable: True
 Last Updated: 12/06/97 16:50:59 Record Count: 0

Columns

Name	Type	Size
det_sec_consec	Number (Long)	4
det_consec	Number (Integer)	2
det_ccl	Number (Integer)	2
det_cma	Number (Integer)	2
det_cco	Number (Integer)	2
det_scu	Number (Integer)	2
det_ssc	Number (Integer)	2
det_descripcion	Text	60
det_debehaber	Number (Integer)	2
det_debe	Number (Double)	8
det_haber	Number (Double)	8
det_concepto	Text	100

Table Indexes

Name	Number of Fields
DET_Null	9
Fields:	det_sec_consec, Ascending det_consec, Ascending det_ccl, Ascending det_cma, Ascending det_cco, Ascending det_scu, Ascending det_ssc, Ascending det_debehaber, Ascending det_descripcion, Ascending
DET_PK	2
Fields:	det_sec_consec, Ascending det_consec, Ascending
DET_SEC_FK_I	1
Fields:	det_sec_consec, Ascending
DET_SSC_FK	5
Fields:	det_ccl, Ascending det_cma, Ascending det_cco, Ascending det_scu, Ascending det_ssc, Ascending
DET_SSC_FK_I	5

Table: detalle

Lunes 14 de Julio de 1997

Page: 19

Fields:

det_cci, Ascending
det_cma, Ascending
det_oco, Ascending
det_scu, Ascending
det_ssc, Ascending

Table: Dia_semana

Lunes 14 de Julio de 1997

Page: 20

Properties

Date Created: 26/06/97 11:57:03 Def. Updatable: True
Last Updated: 12/06/97 16:51:00 Record Count: 8

Columns

Name	Type	Size
dia_cve	Number (Integer)	2
dia_descripcion	Text	30

Table Indexes

Name	Number of Fields
DIA_Null	2
Fields:	dia_cve, Ascending dia_descripcion, Ascending
DIA_PK	1
Fields:	dia_cve, Ascending

Lunes 14 de Julio de 1997

Page: 21

Table: entidad

Properties

Date Created:	26/06/97 11:57:10	Def. Updatable:	True
Last Updated:	20/06/97 15:47:07	Order By On:	True
Order By:	entidad.ent_cve	Record Count:	32

Columns

Name	Type	Size
ent_cve	Number (Integer)	2
ent_desc_corta	Text	15
ent_desc_larga	Text	60

Table indexes

Name	Number of Fields
ENT_Null	2
Fields	ent_cve, Ascending ent_desc_corta, Ascending
ENT_PK	1
Fields:	ent_cve, Ascending

Table: factura

Properties

Date Created: 26/06/97 11:57:18 Def. Updatable: True
 Last Updated: 04/07/97 19:13:42 Order By On: False
 Record Count: 16

Columns

Name	Type	Size
fac_serie	Text	1
fac_num	Number (Integer)	2
fac_fecha	Date/Time	8
fac_cli_cve	Number (Integer)	2
fac_sta_cve	Number (Integer)	2
fac_observacion	Text	60
fac_cli_conduc	Number (Integer)	2
fac_iva	Number (Double)	8
fac_subtotal	Number (Double)	8
fac_totaliva	Number (Double)	8
fac_total	Number (Double)	8
fac_tmo_cve	Number (Integer)	2
fac_tipo_cambio	Number (Double)	8
fac_fecha_revision	Date/Time	8
fac_fecha_vencimiento	Date/Time	8
fac_fecha_pago	Date/Time	8
fac_pago_pendiente	Number (Double)	8
fac_fecha_prob_pago	Date/Time	8

Table Indexes

Name	Number of Fields
FAC_CLI_FK_I	1
Fields:	fac_cli_cve, Ascending
FAC_Null	7
Fields:	fac_serie, Ascending fac_num, Ascending fac_fecha, Ascending fac_cli_cve, Ascending fac_sta_cve, Ascending fac_iva, Ascending fac_tmo_cve, Ascending
FAC_PK	2
Fields:	fac_serie, Ascending fac_num, Ascending
FAC_STA_FK_I	1
Fields:	fac_sta_cve, Ascending
FAC_TMO_FK_I	1

Table: factura

Lunes 14 de Julio de 1997

Page: 23

Fields:

fac_tmo_cve, Ascending

Table: factura_orden

Lunes 14 de Julio de 1997

Page: 25

Properties

Date Created:	26/06/97 11:57:34	Def. Updatable:	True
Last Updated:	12/06/97 16:51:05	Record Count:	7

Columns

Name	Type	Size
fao_ord_num	Number (Integer)	2
fao_fac_serie	Text	1
fao_fac_num	Number (Integer)	2

Table Indexes

Name	Number of Fields
FAO_FAC_FK_I	2
Fields:	fao_fac_serie, Ascending fao_fac_num, Ascending
FAO_Null	3
Fields:	fao_ord_num, Ascending fao_fac_serie, Ascending fao_fac_num, Ascending
FAO_ORD_FK_I	1
Fields:	fao_ord_num, Ascending

Lunes 14 de Julio de 1997

Page: 26

Table: factura_servicio

Properties

Date Created: 26/06/97 11:57:42 Def. Updatable: True
 Last Updated: 12/06/97 16:51:25 Record Count: 16

Columns

Name	Type	Size
fas_fac_serie	Text	1
fas_fac_num	Number (Integer)	2
fas_consec	Number (Integer)	2
fas_ser_cve	Number (Integer)	2
fas_ord_num	Number (Integer)	2
fas_hr_ini_util	Date/Time	8
fas_hr_fin_util	Date/Time	8
fas_hr_ini_fac	Date/Time	8
fas_hr_fin_fac	Date/Time	8
fas_cant	Number (Double)	8
fas_precio_unidad	Number (Double)	8
fas_descuento	Number (Double)	8
fas_tio_cve	Number (Integer)	2

Table Indexes

Name	Number of Fields
FAS_FAC_FK_I	2
Fields:	fas_fac_serie, Ascending fas_fac_num, Ascending
FAS_Null	9
Fields:	fas_fac_serie, Ascending fas_fac_num, Ascending fas_consec, Ascending fas_ser_cve, Ascending fas_hr_ini_fac, Ascending fas_hr_fin_fac, Ascending fas_cant, Ascending fas_precio_unidad, Ascending fas_tio_cve, Ascending
FAS_ORD_FK_I	1
Fields:	fas_ord_num, Ascending
FAS_SER_FK_I	1
Fields:	fas_ser_cve, Ascending
FAS_TIO_FK_I	1
Fields:	fas_tio_cve, Ascending

Lunes 14 de Julio de 1997

Table: forma_pago

Page: 27

Properties

Date Created: 26/06/97 11:59:19 Def. Updatable: True
Last Updated: 12/06/97 16:51:25 Record Count: 5

Columns

Name	Type	Size
fop_cve	Number (Integer)	2
fop_desc_corta	Text	15
fop_desc_larga	Text	60

Table indexes

Name	Number of Fields
FOP_Null	3
Fields:	fop_cve, Ascending fop_desc_corta, Ascending fop_desc_larga, Ascending

Lunes 14 de Julio de 1997

Table: grupo_usuario

Page: 28

Properties

Date Created: 26/06/97 11:59:26 Def. Updatable: True
 Last Updated: 12/06/97 16:51:25 Record Count: 1

Columns

Name	Type	Size
gru_cve	Number (Integer)	2
gru_desc_corta	Text	15
gru_desc_larga	Text	60
gru_tipo_grupo	Number (Integer)	2
gru_tipo_modulo	Number (Integer)	2

Table Indexes

Name	Number of Fields
GRU_Null	5
Fields:	gru_cve, Ascending gru_desc_corta, Ascending gru_desc_larga, Ascending gru_tipo_grupo, Ascending gru_tipo_modulo, Ascending
GRU_PK	1
Fields:	gru_cve, Ascending

Table: ImporteTexto

Properties

Date Created:	26/06/97 11:59:39	Def. Updatable:	True
Last Updated:	12/06/97 16:51:25	Order By On:	False
Record Count:	1		

Columns

Name	Type	Size
FAC_Num	Number (Integer)	2
ImporteTexto	Text	100
TotFac	Number (Double)	8
TotIva	Number (Double)	8
GranTotal	Number (Double)	8

Table Indexes

Name	Number of Fields
fac_nUM	1

Fields: FAC_Num, Ascending

Lunes 14 de Julio de 1997

Page: 46

Table: pago

Properties

Date Created:	26/06/97 12:02:45	Def. Updatable:	True
Last Updated:	12/06/97 16:51:52	Order By On:	False
Record Count:	1		

Columns

Name	Type	Size
pag_cve	Number (Integer)	2
pag_fecha	Date/Time	8
pag_fop_cve	Number (Integer)	2
pag_cli_cve	Number (Integer)	2
pag_sta_cve	Number (Integer)	2
pag_monto	Number (Double)	8
pag_anticipo	Number (Double)	8
pag_num_docto	Text	20
pag_observaciones	Text	60
pag_ban_cve	Number (Integer)	2

Table indexes

Name	Number of Fields
PAG_Null	7
Fields	pag_cve, Ascending
	pag_fecha, Ascending
	pag_fop_cve, Ascending
	pag_cli_cve, Ascending
	pag_sta_cve, Ascending
	pag_monto, Ascending
	pag_num_docto, Ascending

Table: pago_factura

Properties

Date Created: 26/06/97 12:02:52 Def. Updatable: True
Last Updated: 12/06/97 16:51:52 Order By On: True
Record Count: 0

Columns

Name	Type	Size
paf_pag_cve	Number (Integer)	2
paf_fac_serie	Text	1
paf_fac_num	Number (Integer)	2
paf_monto	Number (Double)	8

Table Indexes

Name	Number of Fields
PAF_Null	4
Fields:	paf_pag_cve, Ascending paf_fac_serie, Ascending paf_fac_num, Ascending paf_monto, Ascending

Lunes 14 de Julio de 1997

Table: parametro_cuenta

Page: 48

Properties

Date Created:	26/06/97 12:03:00	Def. Updatable:	True
Last Updated:	27/06/97 13:07:52	Order By On:	False
Record Count:	20		

Columns

Name	Type	Size
par_Cve	Number (Integer)	2
par_Desc_larga	Text	60
par_Valor	Text	15

Table Indexes

Name	Number of Fields
PAR_Null	3
Fields	par_Cve, Ascending par_Desc_larga, Ascending par_Valor, Ascending
PrimaryKey	1
Fields.	par_Cve, Ascending

Table: partida

Properties

Date Created: 25/06/97 12:03:06 Def. Updatable: True
 Last Updated: 20/06/97 15:17:52 Order By On: True
 Order By: partida.par_ssc_scu_cco_cma_ccl Record Count: 79
 _cve DESC

Columna

Name	Type	Size
par_pol_fecha	Date/Time	8
par_pol_folio	Number (Integer)	2
par_consec	Number (Integer)	2
par_ssc_scu_cco_cma_ccl_cve	Number (Integer)	2
par_ssc_scu_cco_cma_cve	Number (Integer)	2
par_ssc_scu_cco_cve	Number (Integer)	2
par_ssc_scu_cve	Number (Integer)	2
par_ssc_cve	Number (Integer)	2
par_debehaber	Number (Integer)	2
par_debe	Number (Double)	8
par_haber	Number (Double)	8
par_concepto	Text	100

Table Indexes

Name	Number of Fields
PAR_Null	9
Fields:	par_pol_fecha, Ascending par_pol_folio, Ascending par_consec, Ascending par_ssc_scu_cco_cma_ccl_cve, Ascending par_ssc_scu_cco_cma_cve, Ascending par_ssc_scu_cco_cve, Ascending par_ssc_scu_cve, Ascending par_ssc_cve, Ascending par_debehaber, Ascending
PAR_PK	3
Fields:	par_pol_fecha, Ascending par_pol_folio, Ascending par_consec, Ascending
PAR_POL_FK_I	2
Fields:	par_pol_fecha, Ascending par_pol_folio, Ascending
PAR_SSC_FK_I	5

Lunes 14 de Julio de 1997

Page: 50

Table: partida

Fields:

par_ssc_scu_cco_cma_cc_cve, Ascending
par_ssc_scu_cco_cma_cve, Ascending
par_ssc_scu_cco_cve, Ascending
par_ssc_scu_cve, Ascending
par_ssc_cve, Ascending

POLIZAPARTIDA

2

Fields:

par_pol_fecha, Ascending
par_pol_folio, Ascending

Table: poliza

Page: 51

Properties

Date Created:	25/06/97 12:03:36	Def. Updatable:	True
Last Updated:	13/06/97 14:25:30	Order By On:	True
Order By:	poliza.pol_fecha	Record Count:	24

Columns

Name	Type	Size
pol_fecha	Date/Time	8
pol_folio	Number (Integer)	2
pol_tip_cve	Number (Integer)	2
pol_tpo_cve	Number (Integer)	2
pol_are_cve	Number (Integer)	2
pol_concepto	Text	100

Table Indexes

Name	Number of Fields
POL_ARE_FK	1
Fields:	pol_are_cve, Ascending
POL_Null	5
Fields:	pol_fecha, Ascending pol_folio, Ascending pol_tpo_cve, Ascending pol_are_cve, Ascending pol_tip_cve, Ascending
POL_PK	2
Fields:	pol_fecha, Ascending pol_folio, Ascending
POL_TIP_FK_I	1
Fields:	pol_tip_cve, Ascending
POL_TPO_FK_I	1
Fields:	pol_tpo_cve, Ascending

Lunes 14 de Julio de 1997

Table: poliza_modelo

Page: 52

Properties

Date Created: 26/06/97 12:03:44 Def. Updatable: True
 Last Updated: 12/06/97 16:52:33 Record Count: 1

Columns

Name	Type	Size
pom_cve	Number (Integer)	2
pom_are_cve	Number (Integer)	2
pom_tpo_cve	Number (Integer)	2
pom_descripcion	Text	60

Table Indexes

Name	Number of Fields
POM_ARE_FK	1
Fields:	pom_are_cve, Ascending
POM_ARE_FK_I	1
Fields:	pom_are_cve, Ascending
POM_Null	4
Fields:	pom_cve, Ascending pom_are_cve, Ascending pom_tpo_cve, Ascending pom_descripcion, Ascending
POM_PK	1
Fields:	pom_cve, Ascending
POM_TPO_FK	1
Fields:	pom_tpo_cve, Ascending
POM_TPO_FK_I	1
Fields:	pom_tpo_cve, Ascending

Properties

Date Created: 26/06/97 12:04:11 Def. Updatable: True
 Last Updated: 12/06/97 16:52:33 Record Count: 0

Columns

Name	Type	Size
pmc_pom_cve	Number (Integer)	2
pmc_ssc_scu_cco_cma_ccl_cve	Number (Integer)	2
pmc_ssc_scu_cco_cma_cve	Number (Integer)	2
pmc_ssc_scu_cco_cve	Number (Integer)	2
pmc_ssc_scu_cve	Number (Integer)	2
pmc_ssc_cve	Number (Integer)	2
pmc_cargoabono	Number (Integer)	2

Table Indexes

Name	Number of Fields
PMC_Null	7
Fields:	pmc_pom_cve, Ascending pmc_ssc_scu_cco_cma_ccl_cve, Ascending pmc_ssc_scu_cco_cma_cve, Ascending pmc_ssc_scu_cco_cve, Ascending pmc_ssc_scu_cve, Ascending pmc_ssc_cve, Ascending pmc_cargoabono, Ascending
PMC_PK	7
Fields:	pmc_pom_cve, Ascending pmc_ssc_scu_cco_cma_ccl_cve, Ascending pmc_ssc_scu_cco_cma_cve, Ascending pmc_ssc_scu_cco_cve, Ascending pmc_ssc_scu_cve, Ascending pmc_ssc_cve, Ascending pmc_cargoabono, Ascending
PMC_POM_FK	1
Fields:	pmc_pom_cve, Ascending
PMC_POM_FK_I	1
Fields:	pmc_pom_cve, Ascending
PMC_SSC_FK	5
Fields:	pmc_ssc_scu_cco_cma_ccl_cve, Ascending pmc_ssc_scu_cco_cma_cve, Ascending pmc_ssc_scu_cco_cve, Ascending pmc_ssc_scu_cve, Ascending pmc_ssc_cve, Ascending
PMC_SSC_FK_I	5

Lunes 14 de Julio de 1997

Table: poliza_modelo_cuenta

Page: 54

Fields:

pmc_ssc_scu_cco_cma_ccl_cve, Ascending
pmc_ssc_scu_cco_cma_cve, Ascending
pmc_ssc_scu_cco_cve, Ascending
pmc_ssc_scu_cve, Ascending
pmc_ssc_cve, Ascending

Table: rep_aux_conta

Page: 57

Properties

Date Created:	26/06/97 12:04:30	Def. Updatable:	True
Last Updated:	12/06/97 16:52:38	Order By On:	True
Order By:	rep_aux_conta.rco_cve	Record Count:	17

Columns

Name	Type	Size
rco_cve	Number (Integer)	2
rco_ccl	Number (Integer)	2
rco_cma	Number (Integer)	2
rco_cco	Number (Integer)	2
rco_scu	Number (Integer)	2
rco_sec	Number (Integer)	2
rco_fecha_inicial	Date/Time	8
rco_fecha_final	Date/Time	8
rco_saldo	Number (Double)	8
rco_debe	Number (Double)	8
rco_haber	Number (Double)	8
rco_debe_par	Number (Double)	8
rco_haber_par	Number (Double)	8
rco_concepto	Text	60
rco_fecha_par	Date/Time	8
rco_tipo	Number (Integer)	2

Table Indexes

Name	Number of Fields
RCO_Nu#	1
Fields:	rco_cve, Ascending

Table: rep_aux_conta2

Lunes 14 de Julio de 1997

Page: 58

Properties

Date Created:	09/06/97 14:03:55	Def. Updatable:	True
Last Updated:	11/06/97 12:34:52	Order By On:	False
Record Count:	24		

Columns

Name	Type	Size
rca_cve	Number (Integer)	2
rca_descripcion	Text	30
rca_fecha	Date/Time	8

Table: rep_aux_cxc

Properties

Date Created:	26/06/97 12:04:52	Def. Updatable:	True
Last Updated:	16/03/97 12:35:35	Order By On:	False
Record Count:	8		

Columns

Name	Type	Size
rac_cve	Number (Integer)	2
rac_nombre	Text	60
rac_direccion	Text	60
rac_rfc	Text	14
rac_cl	Number (Integer)	2
rac_fecha	Date/Time	8
rac_serie	Text	1
rac_num	Number (Double)	8
rac_subtotal	Number (Double)	8
rac_iva	Number (Double)	8
rac_total	Number (Double)	8
rac_stafac	Number (Integer)	2
rac_recibo	Number (Integer)	2
rac_monto	Number (Double)	8
rac_estapag	Number (Integer)	2
rac_saldoposterior	Number (Double)	8
rac_fecha_inicial	Date/Time	8
rac_fecha_final	Date/Time	8

Table Indexes

Name	Number of Fields
rac_Null	1
Fields:	rac_cve, Ascending

Table: sala

Lunes 14 de Julio de 1997

Page: 61

Properties

Date Created:	26/06/97 12:05:04	Def. Updatable:	True
Last Updated:	27/06/97 13:10:01	Order By On:	False
Record Count:	5		

Columns

Name	Type	Size
sal_cve	Number (Integer)	2
sal_desc_corta	Text	15
sal_desc_larga	Text	60
sal_cuenta	Text	30
sal_cuenta_dev	Text	30

Table Indexes

Name	Number of Fields
SAL_Null	2
Fields:	sal_cve, Ascending sal_desc_corta, Ascending
SAL_PK	1
Fields:	sal_cve, Ascending

Table: secuencia

Page: 63

Properties

Date Created: 26/06/97 12:05:17 Def. Updatable: True
Last Updated: 12/06/97 16:52:44 Record Count: 1

Columns

Name	Type	Size
sec_consec	Number (Long)	4

Table Indexes

Name	Number of Fields
SEC_PK	1

Fields: sec_consec, Ascending

Table: subcuenta

Lunes 14 de Julio de 1997

Page: 67

Properties

Date Created:	13/06/97 13:47:20	Def. Updatable:	True
Last Updated:	25/06/97 14:50:44	Order By On:	False
Record Count:	15	Row Height:	270

Columns

Name	Type	Size
scu_cco_cma_ccl_cve	Number (Integer)	2
scu_cco_cma_cve	Number (Integer)	2
scu_cco_cve	Number (Integer)	2
scu_cve	Number (Integer)	2
scu_tipo_cuenta	Number (Integer)	2
scu_descripcion	Text	60

Table Indexes

Name	Number of Fields
CENTRO_DE_COSTOSSUBCUENTA	3
Fields:	scu_cco_cma_ccl_cve, Ascending scu_cco_cma_cve, Ascending scu_cco_cve, Ascending
SCU_CCO_FK_I	3
Fields:	scu_cco_cma_ccl_cve, Ascending scu_cco_cma_cve, Ascending scu_cco_cve, Ascending
SCU_Null	6
Fields:	scu_cco_cma_ccl_cve, Ascending scu_cco_cma_cve, Ascending scu_cco_cve, Ascending scu_cve, Ascending scu_tipo_cuenta, Ascending scu_descripcion, Ascending
SCU_PK	4
Fields:	scu_cco_cma_ccl_cve, Ascending scu_cco_cma_cve, Ascending scu_cco_cve, Ascending scu_cve, Ascending

Table: subsubcuenta

Page: 68

Properties

Date Created: 13/06/97 13:47:20 Def. Updatable: True
 Last Updated: 12/06/97 16:50:59 Order By On: False
 Record Count: 61

Columns

Name	Type	Size
ssc_scu_cco_cma_ccl_cve	Number (Integer)	2
ssc_scu_cco_cma_cve	Number (Integer)	2
ssc_scu_cco_cve	Number (Integer)	2
ssc_scu_cve	Number (Integer)	2
ssc_cve	Number (Integer)	2
ssc_tipo_cuenta	Number (Integer)	2
ssc_descripcion	Text	60

Table indexes

Name	Number of Fields
SSC_Null	7
Fields:	ssc_scu_cco_cma_ccl_cve, Ascending ssc_scu_cco_cma_cve, Ascending ssc_scu_cco_cve, Ascending ssc_scu_cve, Ascending ssc_cve, Ascending ssc_tipo_cuenta, Ascending ssc_descripcion, Ascending
SSC_PK	5
Fields:	ssc_scu_cco_cma_ccl_cve, Ascending ssc_scu_cco_cma_cve, Ascending ssc_scu_cco_cve, Ascending ssc_scu_cve, Ascending ssc_cve, Ascending
SSC_SCU_FK_I	4
Fields:	ssc_scu_cco_cma_ccl_cve, Ascending ssc_scu_cco_cma_cve, Ascending ssc_scu_cco_cve, Ascending ssc_scu_cve, Ascending

Lunes 14 de Julio de 1997

Table: tipo_costo

Page. 71

Properties

Date Created:	06/06/97 14:39:59	Def. Updatable:	True
Last Updated:	12/06/97 16:50:59	Order By On:	False
Record Count:	15		

Columns

Name	Type	Size
tco_cve	Number (Integer)	2
tco_desc_larga	Text	60

Table Indexes

Name	Number of Fields
tco_Null	1
Fields:	tco_cve, Ascending
TCO_PK	1
Fields:	tco_cve, Ascending

Table: tipo_balance

Properties

Date Created:	10/06/97 11:22:19	Def. Updatable:	True
Last Updated:	12/06/97 16:50:59	Order By On:	False
Record Count:	5		

Columns

Name	Type	Size
tib_cve	Number (Integer)	2
tib_desc_larga	Text	60

Table Indexes

Name	Number of Fields
PrimaryKey	1
Fields:	tib_cve, Ascending

Lunes 14 de Julio de 1997

Table: tipo_pago

Page: 75

Properties

Date Created: 26/06/97 12:06:50
Last Updated: 12/06/97 16:52:54
Record Count: 6

Def. Updatable: True
Order By On: False

Columns

Name	Type	Size
tip_cve	Number (Integer)	2
tip_descripcion	Text	60

Table Indexes

Name	Number of Fields
TIP_Null	2
Fields:	tip_cve, Ascending tip_descripcion, Ascending
TIP_PK	1
Fields:	tip_cve, Ascending

Table: tipo_poliza

Properties

Date Created:	26/06/97 12:06:56	Def. Updatable:	True
Last Updated:	12/06/97 16:52:54	Order By On:	False
Record Count:	2		

Columns

Name	Type	Size
tpo_cve	Number (Integer)	2
tpo_desc_corta	Text	15
tpo_desc_larga	Text	60
tpo_cheque	Number (Integer)	2

Table Indexes

Name	Number of Fields
TPO_Null	4
Fields:	tpo_cve, Ascending tpo_desc_corta, Ascending tpo_desc_larga, Ascending tpo_cheque, Ascending
TPO_PK	1
Fields:	tpo_cve, Ascending

Table: usuario

Lunes 14 de Julio de 1997

Page: 80

Properties

Date Created: 26/06/97 12:07:18
 Last Updated: 12/06/97 16:52:54
 Record Count: 1

Def. Updatable: True
 Order By On: False

Columns

Name	Type	Size
usu_cve	Text	5
usu_password	Text	10
usu_nombre	Text	60
usu_are_cve	Number (Integer)	2
usu_gru_cve	Numbe. (Integer)	2

Table Indexes

Name	Number of Fields
USU_ARE_FK_I	1
Fields:	usu_are_cve, Ascending
USU_GRU_FK_I	1
Fields:	usu_gru_cve, Ascending
USU_Null	5
Fields:	usu_cve, Ascending usu_password, Ascending usu_nombre, Ascending usu_are_cve, Ascending usu_gru_cve, Ascending
USU_PK	1
Fields:	usu_cve, Ascending

3.3 DESARROLLO DE PANTALLAS

3.3.1 DISEÑO Y CONSTRUCCIÓN DE PANTALLAS

Para el diseño y construcción de pantallas se utilizará la herramienta visual "Visual Basic". Visual Basic es una herramienta muy amigable que trabaja en base a un proyecto.

Un proyecto consta de formas, módulos, clases y recursos los cuales al ser compilados generarán un programa ejecutable (ver fig. 3.3.A)

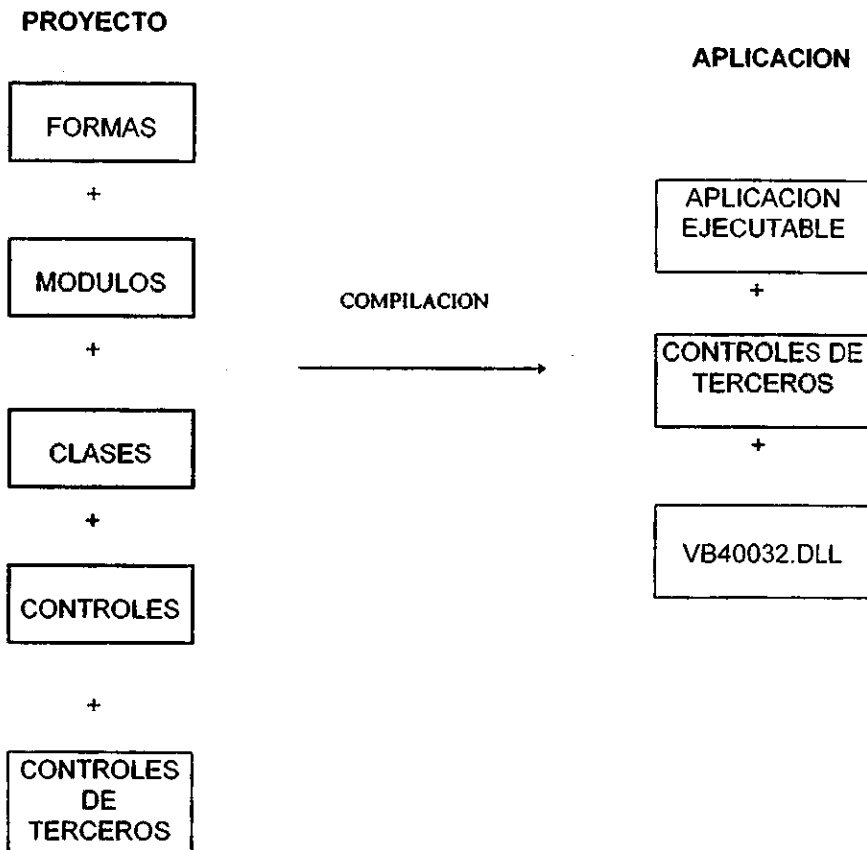


FIG. 1.3.A UN PROYECTO EN VISUAL BASIC

Visual Basic trabaja por medio de objetos llamados controles, los cuales son incrustados en formas para generar una pantalla de captura o despliegue.

Cada control contiene propiedades, métodos y eventos. A través de ellos es como se diseña y construye una pantalla.

- Las propiedades son características de un control, por ejemplo ancho, largo, color, posición en la pantalla, nombre ,etc.
- Los métodos son acciones asociadas a cada control con las cuales podemos hacer que el control haga cierta función. Por ejemplo desplegar u ocultar una pantalla, llenarse con datos, etc.
- Los eventos son acciones que se disparan automáticamente, por ejemplo la entrada de datos por el teclado, el clic del ratón, etc.

Existen diferentes controles, entre los más usados en nuestra aplicación están:

- Cajas de texto : son usados para obtener información del usuario, o bien para despliegue de datos.
- Combo box : es un control con múltiples cajas de texto, de las cuales se puede seleccionar una.
- Etiquetas : son para poner nombres a campos o en general para colocar diferentes descripciones en las pantallas.
- Botones : son usados para que el usuario nos indique alguna acción a realizar, como realizar una alta, baja, reporte, salirse de una forma, etc.
- DB grid : son matrices para poder capturar o desplegar varios registros de datos de una tabla.
- Data : realiza la interfaz entre una base de datos y los controles donde se va a capturar y desplegar la información.
- Menús : un menu organiza y agrupa, las funciones y módulos de la aplicación para que al usuario le sea más fácil su acceso.

- **Crystal Report** : este control se usa para llamar desde una pantalla un reporte hecho en el reporteador Crystal Report,

En general para desarrollar una aplicación en Visual Basic se deben seguir los siguientes pasos :

1. **Crear una interface de usuario** : esto se hará dibujando controles sobre una forma, asignándoles un nombre para hacer más fácil la lectura del código y la depuración de la aplicación.
2. **Seleccionar las propiedades de los controles** : Después de incrustar los controles dentro de una forma se deben de dar propiedades a los controles. Estas propiedades se pueden dar desde el tiempo de diseño de la pantalla, o bien cuando la aplicación esta corriendo.
3. **Escribir código para los eventos** : Se puede agregar código que se va ejecutar en respuesta a algún evento generado.
4. **Salvar el proyecto** : Se debe de dar un nombre al proyecto para que se pueda escribir a disco.
5. **Probar y depurar la aplicación** : dentro de Visual Basic existen herramientas para realizar estas funciones.

Los módulos de contabilidad y cuentas por cobrar constan de muchas formas, por lo tanto solo se presentarán una muestra de las pantallas más importantes.

El código de estas pantallas mas importantes se muestra en el apéndice C.

CONTABILIDAD

Pantalla principal

Es la primera pantalla que se despliega cuando es ejecutado el modulo de contabilidad. Esta pantalla pedirá un usuario y una contraseña de usuario para poder acceder el sistema. Una vez validado el usuario se desplegara un menu del cual el usuario podrá escoger una opción para trabajar. Las opciones a las que podrá acceder el operador dependerá del nivel con el que fue dado de alta.

El menu esta clasificado de la siguiente manera :

Controlar catálogos

- Actualizar clasificación
- Actualizar cuentas de mayor
- Actualizar centros de costos
- Actualizar subcuentas
- Actualizar subsubcuentas
- Actualizar áreas de trabajo
- Póliza Modelo
 - Actualizar tipo de póliza
 - Actualizar póliza modelo
 - Actualizar póliza modelo cuenta

Pólizas

- Pólizas manuales
- Pólizas modelo
- Ajustes de pólizas

Reportes

Auxiliares por nivel de cuenta

Diario general

Balanza de comprobación

Balanza de comprobación X cuenta de mayor

Auxiliares por cuenta

Cuentas con saldo

Cierre

Cierre

Usuarios

Actualizar grupos

Actualizar usuarios

Estados financieros

Estados de resultados financieros

Captura de costos para estados de resultados financieros

Generar reporte de estados de resultados financieros

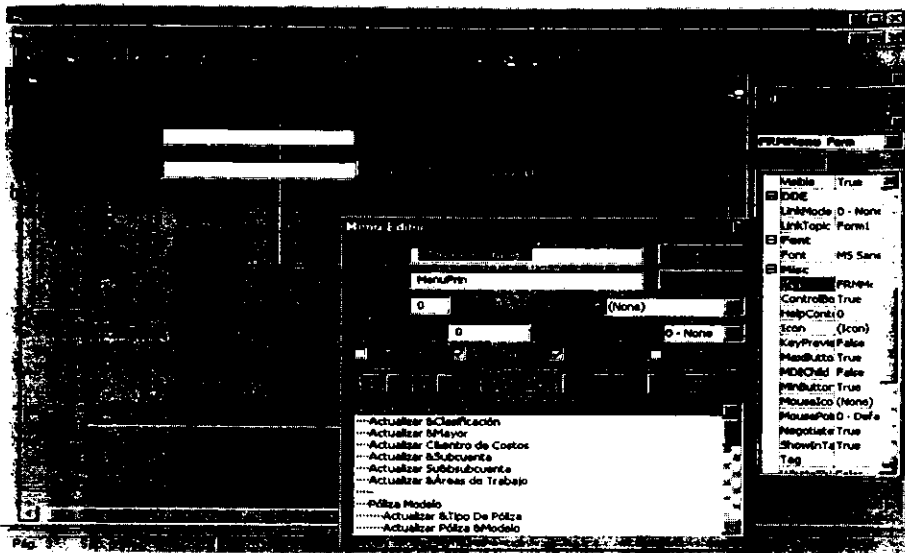
Balance general

Captura de costos para balance general

Generar reporte de balance general

Salir

Diseño de la pantalla principal



La pantalla contiene los siguientes controles:

- Forma principal : Esta es una forma MDI (*Multiple Document Interface - Interfaz de múltiples documentos*) que contiene al menu principal. Desde esta forma se pueden mandar a llamar a diferentes formas hijas .
- Menu principal : contiene la definición de todas las opciones del módulo de contabilidad. Desde este menu se dispararán el despliegue de todas las pantallas del sistema, así como la generación de reportes.
- Etiqueta con el nombre del módulo de contabilidad.
- Una caja de texto para capturar el nombre de usuario a operar.
- Una caja de texto para capturar la contraseña del usuario a operar.

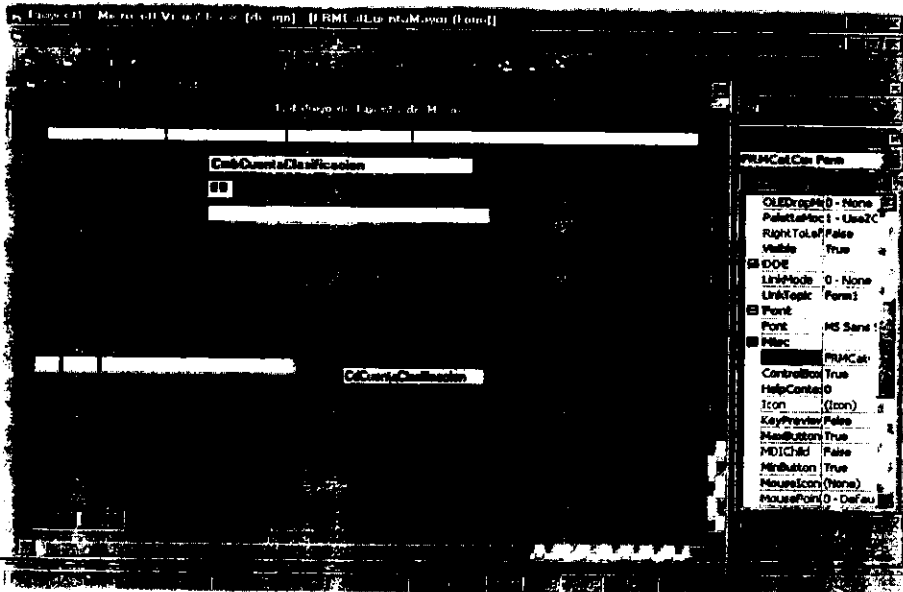
Pantalla de cuentas de mayor

Esta pantalla se usará para realizar actualizaciones (altas, bajas, modificaciones) al catálogo de cuentas de mayor a través de diversos botones. En un grid (cuadrícula o matriz) se presentarán todos los registros que contiene actualmente el catálogo. Los datos que manejará esta pantalla son :

- Clasificación de la cuenta
- Numero de cuenta de mayor
- Tipo de cuenta
- Descripción

La relación de datos se explica en el diseño y construcción de la base de datos.

Diseño de pantalla

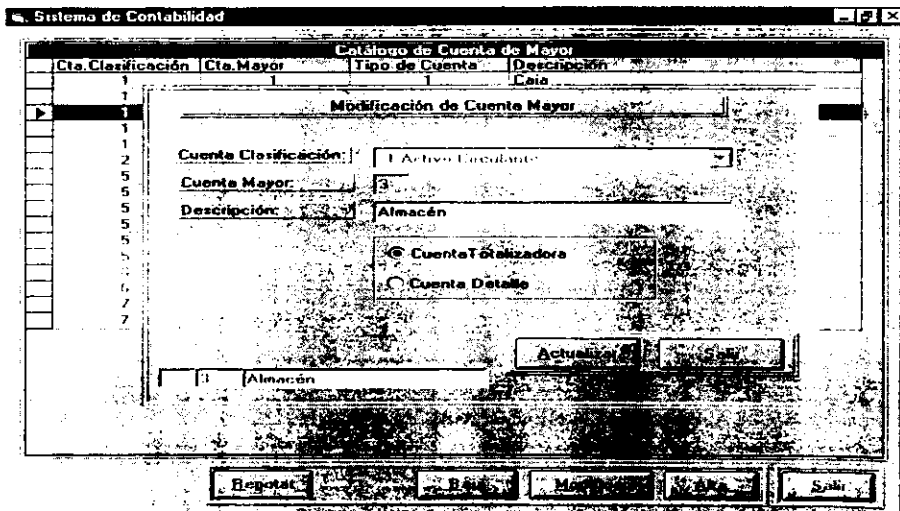


La pantalla contiene los siguientes controles:

- Un control de datos para realizar interfaz con la tabla de cuentas de mayor.
- Un control de datos para realizar interfaz con la tabla de clasificación de cuentas

- Un grid asociado al control de datos de cuentas de mayor para vaciar los registros del catalogo de cuentas de mayor.
- Botón para realizar alta de registro.
- Botón para realizar baja de registro.
- Botón para realizar modificación de registro
- Botón para realizar reporte del catalogo de cuentas de mayor.
- Botón para salirse de esta pantalla.
- Combo Box asociado al control de datos de clasificación de cuentas para seleccionar la clasificación de la cuenta.
- Caja de texto para capturar cuenta de mayor.
- Caja de texto para capturar tipo de cuenta.
- Caja de texto para capturar descripción de la cuenta.

A continuación se presenta la pantalla pidiendo los datos de un registro de cuenta de mayor.



Pantalla para poliza modelo

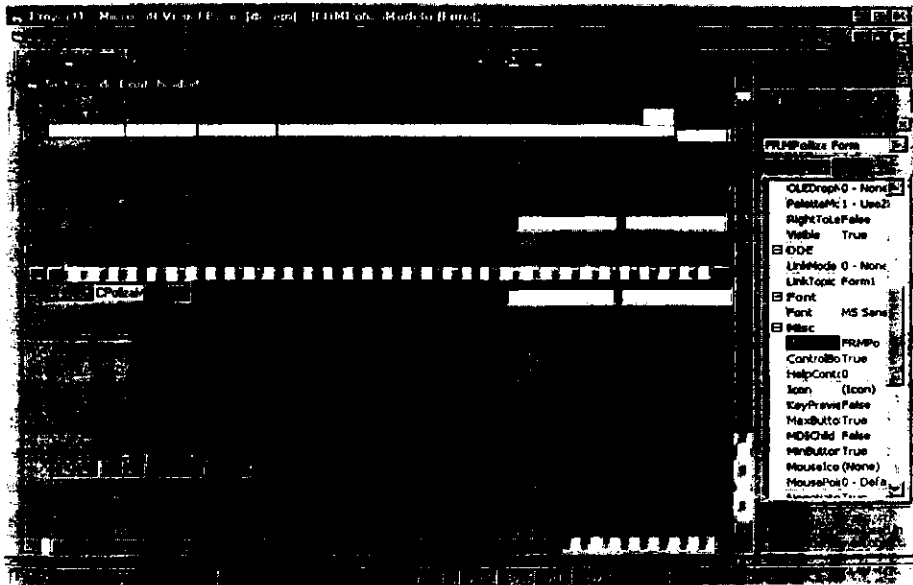
Esta pantalla se usara para realizar consultas de pólizas modelo existentes y para realizar altas de nuevas pólizas modelo a través de la opción de alta.

Cuando se realiza la consulta de pólizas modelos se despliegan dos grids sobre la pantalla. En un grid se presentarán todas las pólizas modelos existentes, mientras que en el otro se presentará las partidas que forman a cada póliza:

Cuando se invoca la alta de una nueva póliza se pedirán los siguientes datos:

- Tipo de póliza
- Concepto
- Partidas que conforman a la póliza, estas partidas pueden ser de cargo o de abono.

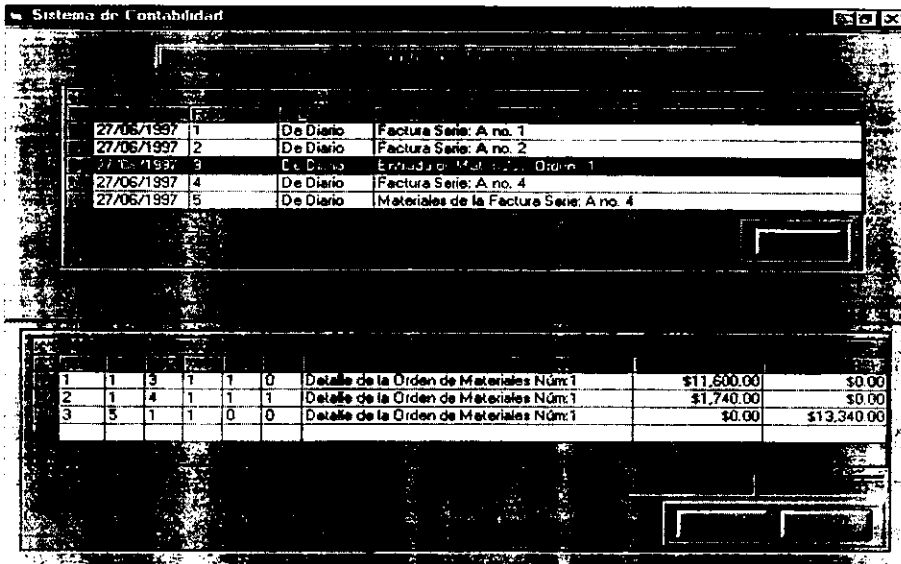
Diseño de pantalla



La pantalla contiene los siguientes controles:

- Un control de datos para realizar interfaz con la tabla de tipos de pólizas
- Un control de datos para realizar interfaz con la tabla de pólizas
- Un control de datos para realizar interfaz con la tabla de partidas
- Un control de datos para realizar interfaz con la tabla de subsubcuentas
- Un control de datos para realizar interfaz con la tabla de detalle
- Un grid asociado al control de datos de pólizas para desplegar las pólizas modelo existentes
- Un grid asociado al control de datos de detalle para desplegar el detalle de las pólizas
- Botón para realizar alta de registro.
- Botón para realizar baja de registro.
- Botón para realizar modificación de registro
- Botón para realizar reporte del catalogo de cuentas de mayor.
- Botón para salirse de esta pantalla
- Combo box asociado al control de datos de clasificación de cuentas para seleccionar la clasificación de la cuenta.
- Caja de texto para capturar cuenta de mayor.
- Caja de texto para capturar tipo de cuenta.
- Caja de texto para capturar descripción de la cuenta.

A continuación se presenta la pantalla con datos.



Pantalla para usuarios

Esta pantalla se usara para realizar actualizaciones (altas, bajas, modificaciones) al catálogo de usuarios que pueden operar el sistema. A través de esta pantalla se darán las facultades a cada uno de los usuarios, dependiendo de las facultades que se le asignen podrán acceder a los diferentes módulos. Las facultades asignadas dependerán del área y grupo seleccionado. Los datos que se requieren para dar de alta a un usuario son :

- Area del usuario
- Grupo del usuario
- Clave del usuario
- Nombre del usuario
- Contraseña

Diseño de la pantalla

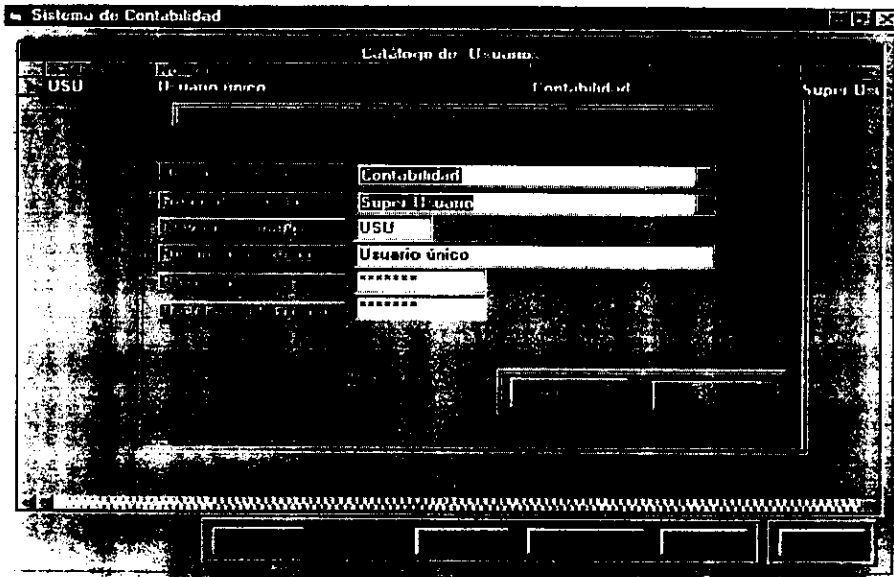


La pantalla contiene los siguientes controles:

- Un control de datos para realizar interfaz con la tabla de grupos de usuario.
- Un control de datos para realizar interfaz con la tabla de áreas.
- Un grid asociado a la tabla de usuarios para desplegar actualizar y consultar la tabla de usuarios.
- Botón para realizar alta de registro.
- Botón para realizar baja de registro.
- Botón para realizar modificación de registro
- Botón para realizar reporte del catalogo de cuentas de mayor.
- Botón para salirse de esta pantalla
- Combo box asociado al control de datos de grupo de usuarios para seleccionar el grupo.

- Combo box asociado al control de de datos área para seleccionar el área.
- Caja de texto para capturar clave del usuario.
- Caja de texto para capturar nombre del usuario.
- Caja de texto para capturar contraseña del usuario.
- Caja de texto para confirmar contraseña del usuario.

A continuación se presenta la pantalla con datos.



Pantalla para captura de estados financieros

A través de esta pantalla se podrán capturar los datos requeridos para generar diversos tipos de estados financieros para cada una de las salas. Esta pantalla da la opción de diferentes tipos de costos:

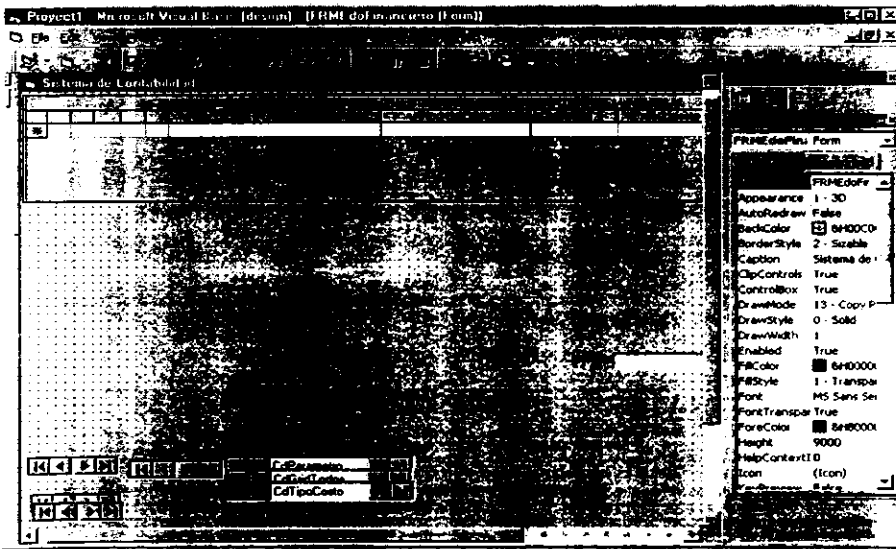
- Costos de ingresos
- Costos variables

- Costos fijos
- Gastos de operación
- Productos financieros
- No deducibles.

Esta pantalla permite realizar de una forma amigable la asociación de diferentes cuentas a cada una de la salas para poder obtener el estado financiero por sala.

Una vez realizada esta asociación de cuentas-sala existe otra pantalla con la cual se podrá obtener los reportes de los diferentes estados financieros generados.

Diseño de la pantalla



La pantalla contiene los siguientes controles:

- Un control de datos para realizar interfaz con la tabla de tipos de costos
- Un control de datos para realizar interfaz con la tabla de salas.

- Un control de datos para realizar interfaz con la tabla Subsubcuentas
- Un grid asociado a la tabla de cuentas
- Un grid para guardar asociación de cuentas seleccionadas, salas y tipos de costos.
- Botón para aceptar captura de estado financiero
- Combo box asociado al control de datos de tipos de costos para seleccionar tipo de costo.
- Combo box asociado al control de datos salas para seleccionar sala.
- Botones en forma de flecha para realizar asociación entre cuentas y salas.

A continuación se presenta la pantalla con datos capturados

The screenshot shows a window titled "Sistema de Contabilidad". At the top, there is a text box containing "02-Costo de Devoluciones" and a date field showing "4/08/97". Below this is another text box labeled "Sala de Paint Box".

A list of accounts is displayed with columns for account numbers and names:

5	1	1	0	0	Proveedores
6	1	1	0	0	Bank
6	1	3	0	0	Paint Box
6	1	4	0	0	Costado

At the bottom, a table shows the captured data for associations:

6	1	1	0	0	Sala de Paint Box	02-Costo de Devoluciones	31/08/97	\$0.00
6	1	2	0	0	Sala de Paint Box	02-Costo de Devoluciones	31/08/97	\$0.00

CUENTAS POR COBRAR

Pantalla principal

Es la primera pantalla que se despliega cuando es ejecutado el modulo de cuentas por cobrar. Esta pantalla pedirá un usuario y una contraseña de usuario para poder acceder el sistema. Una vez validado el usuario se desplegara un menue del cual el usuario podrá escoger una opción para trabajar. Las opciones a las que podrá acceder el operador dependerá del nivel con el que fue dado de alta en el módulo de contabilidad.

El menue esta clasificado de la siguiente manera :

Controlar catálogos

- Actualizar clientes

- Actualizar bancos

Controlar pagos

- Pagos

- Aplicar pagos anticipados

- Reporte de cifras de control

Controlar ajustes

- Cancelar factura

- Pago no considerado

- Cancelación de pago mal aplicado

 - Por cheque devuelto

 - Por error de aplicación.

Consultas y reportes

- Estado de cuenta general

- Pronostico de cobranza

- Pronostico de revisión

Saldo vencido

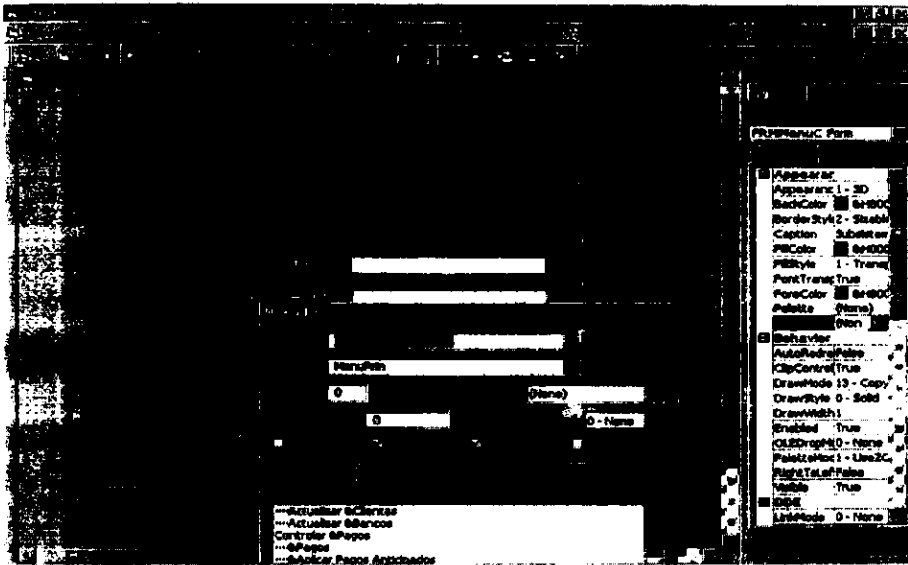
Antigüedad de saldos

Consultar pagos

Reporte de facturas

Salir

Diseño de pantalla



La pantalla contiene los siguientes controles:

- Forma principal : Es una forma MDI que contiene al menu principal. Desde esta forma se pueden mandar a llamar a diferentes formas hijas .
- Menu principal : contiene la definición de todas las opciones del módulo de cuentas por cobrar. Desde este menu se dispararán el despliegue de todas las pantallas del sistema, así como la generación de reportes.
- Etiqueta con el nombre del módulo de cuentas por cobrar

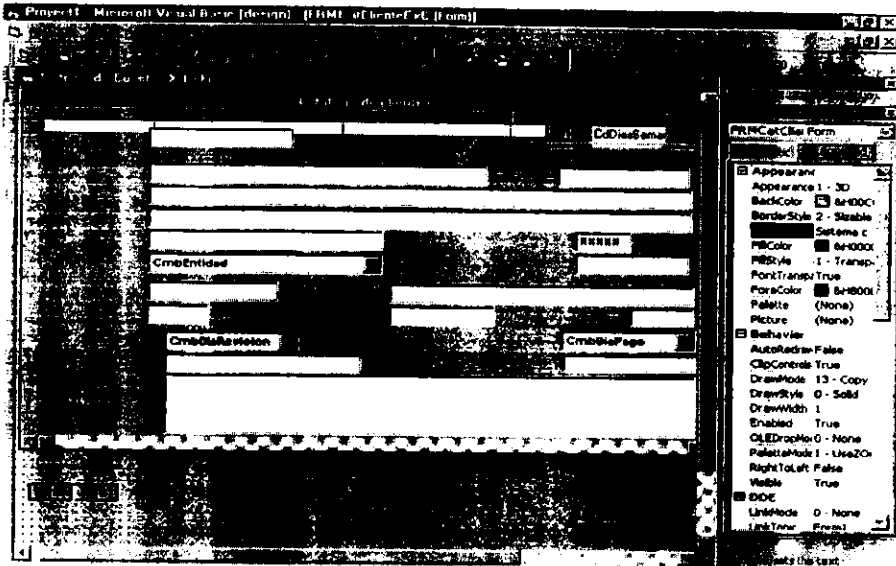
- Una caja de texto para capturar el nombre de usuario a operar.
- Una caja de texto para capturar la contraseña del usuario a operar.

Pantalla de clientes

Esta pantalla se usara para realizar actualizaciones (altas, bajas, modificaciones) al catalogo de clientes a través de diversos botones. En un grid (cuadrícula o matriz) se presentarán todos los registros que contiene actualmente el catalogo. Los datos que manejará esta pantalla son :

- Nombre corto
- Razón social
- Registro federal de causantes
- Dirección
- Colonia
- Ciudad
- Código postal
- Entidad
- Telefonos
- Fax
- Clasificación
- % descuento
- Limite de credito
- meses de credito
- Día de revisión.
- Día de pago
- Saldo
- Fecha de ultima venta
- Referencia

Diseño de la pantalla



La pantalla contiene los siguientes controles:

- Un control de datos para realizar interfaz con la tabla de clientes
- Un grid asociado al control de datos de clientes para vaciar los registros del catalogo de clientes
- Botón para realizar alta de registro.
- Botón para realizar baja de registro.
- Botón para realizar modificación de registro
- Botón para realizar reporte del catalogo de cuentas de mayor.
- Botón para salirse de esta pantalla
- Combo box asociado a la tabla de días para selección de día de pago
- Combo box asociado a la tabla de días para selección de día de revisión
- Combo box asociado a la tabla de entidades para selección de entidad.

- Cajas de texto para capturar los siguientes datos:

nombre corto, razón social, registro federal de causantes, dirección, colonia, ciudad, código postal, teléfonos, fax, clasificación, % descuento, límite de crédito, meses de crédito, saldo, fecha de última venta, referencia

A continuación se presenta la pantalla pidiendo los datos de un cliente

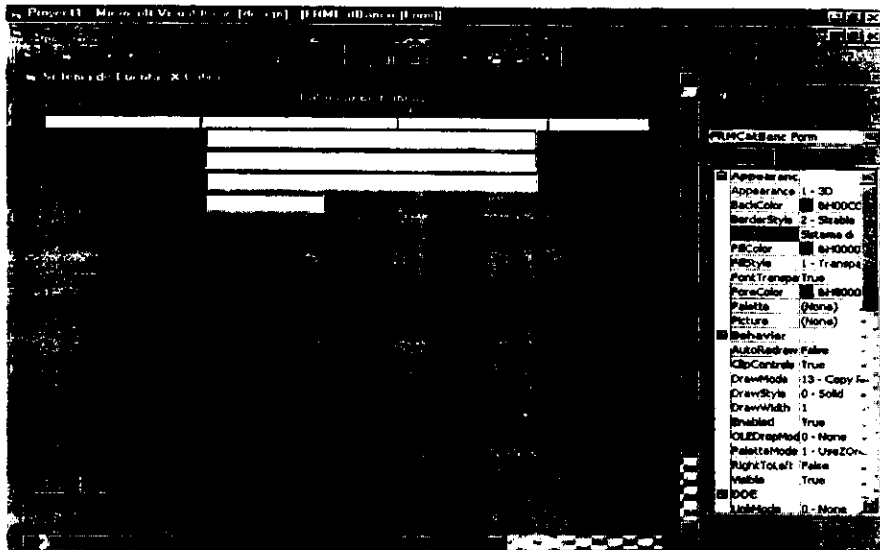
The screenshot shows a window titled "Sistema de Cuentas X Cobrar" with a data entry form. The form contains the following fields and values:

Nombre:	Jurado		
Razón Social:	Jurado Meza Antonio	Registro Federal:	LLL 102938 WW
Dirección:	Navegante y Fundadores 123		
Colonia:	Joyas del Pedregal		
Ciudad:	Coyoacán	Código Postal:	2400
Ejido:	Chinubun	Teléfono:	2 51 29 44
Fax:	2 51 29 44	Clasificación:	Reglas
Descuento:	0.00	Límite de crédito:	700,000.00
Supervisor:	Juarez	Fecha de última venta:	Juarez
Saldo:	0	Fecha de última venta:	4/08/97
Referencia:	Pagos en la Oficina Central		

Pantalla del catálogo de bancos

Esta pantalla se usará para realizar actualizaciones (altas, bajas, modificaciones) al catálogo de bancos a través de diversos botones. En un grid (cuadrícula o matriz) se presentarán todos los registros que contiene actualmente el catálogo.

Diseño de la pantalla



Los datos que se manejan en esta pantalla son :

- Nombre
- Dirección
- Responsable
- Teléfono

La pantalla contiene los siguientes controles:

- Un control de datos para realizar interfaz con la tabla de bancos
- Un grid asociado al control de datos de bancos para vaciar los registros del catalogo de bancos.
- Botón para realizar alta de registro.
- Botón para realizar baja de registro.

- Botón para realizar modificación de registro
- Botón para realizar reporte del catalogo de cuentas de mayor.
- Botón para salirse de esta pantalla
- Cajas de texto para capturar los siguientes datos:

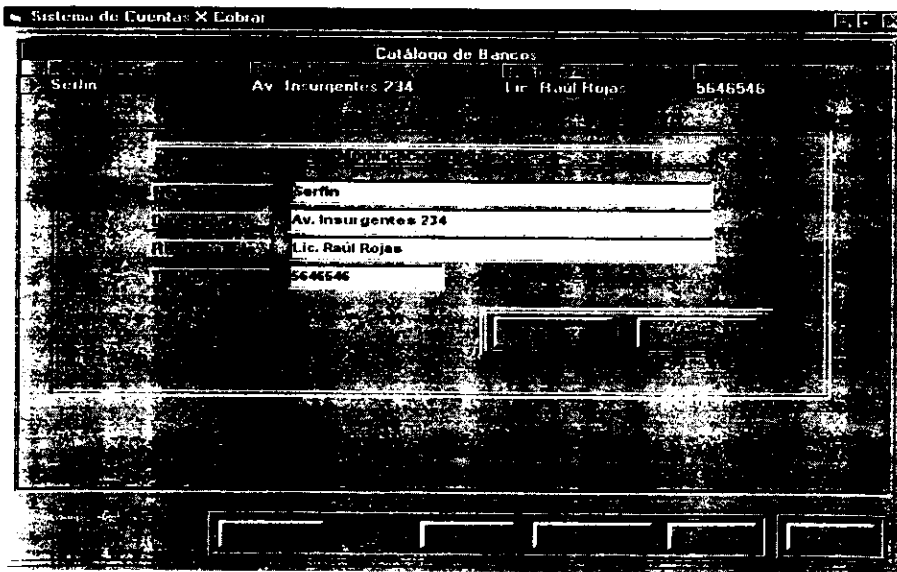
Nombre

Dirección

Responsable

Teléfono

A continuación se presenta la pantalla pidiendo los datos de un banco



Pantalla para cancelación de facturas

A través de esta pantalla se podrán realizar cancelación de facturas. La pantalla se divide en dos partes, en la parte superior aparecerá la información del cliente

seleccionado y en la parte inferior se desplegará una lista con todas las facturas del cliente. Solo se debe seleccionar una factura y seleccionar la opción de cancelar factura.

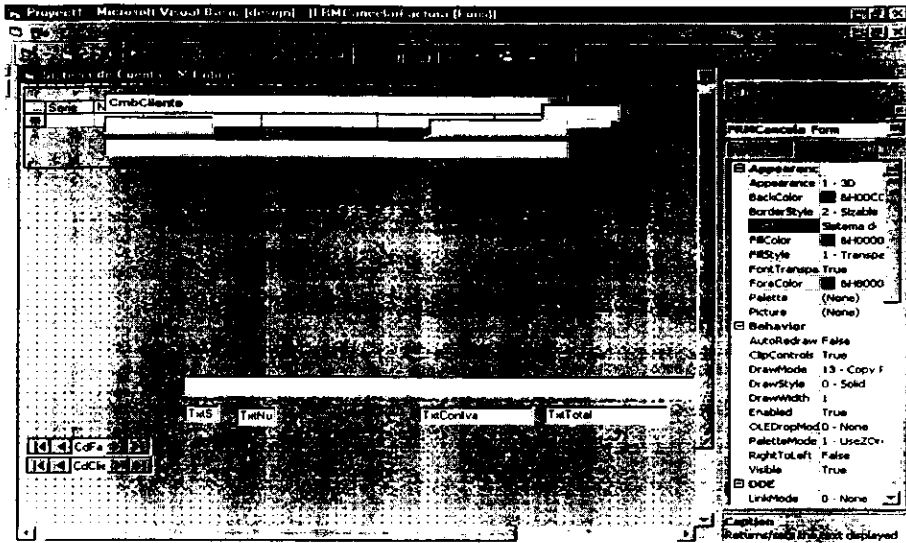
La información del cliente que se despliega es la siguiente:

- Razón social
- Registro federal de causantes
- Saldo
- Dirección.

Mientras que la información de las facturas es :

- Serie
- Numero
- Fecha
- Subtotal
- Iva
- Total

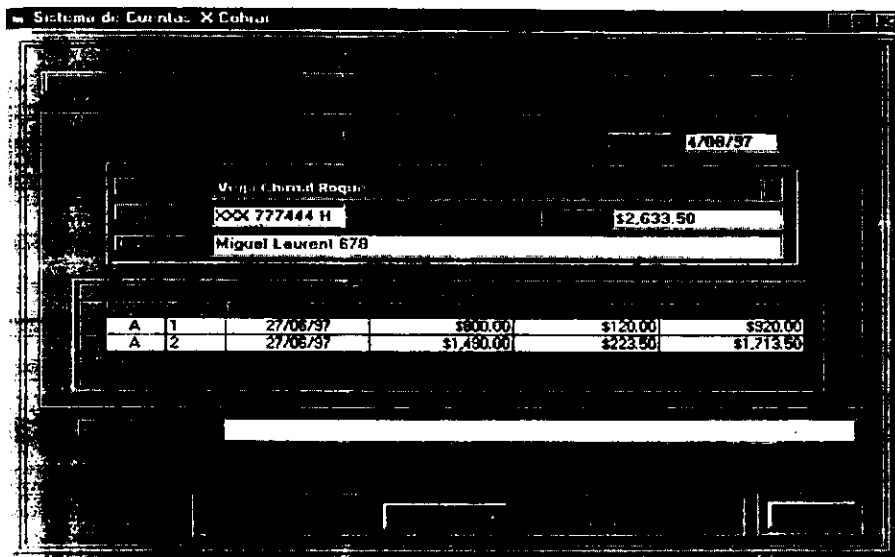
Diseño de la pantalla



La pantalla contiene los siguientes controles:

- Un control de datos para realizar interfaz con la tabla de clientes
- Un control de datos para realizar interfaz con la tabla de facturas
- Un grid asociado al control de datos de facturas para listar las facturas de un determinado cliente.
- Combo Box asociado al control de datos de clientes para seleccionar un cliente específico.
- Botón para realizar cancelación de factura.
- Cajas de texto para desplegar los siguientes datos:
 - Registro federal de causantes
 - Saldo
 - Dirección.

A continuación se presenta la cancelación de una factura.



Pantalla para realización de pagos

Esta pantalla permitirá realizar pagos tanto ordinarios de facturas que se deben como anticipados.

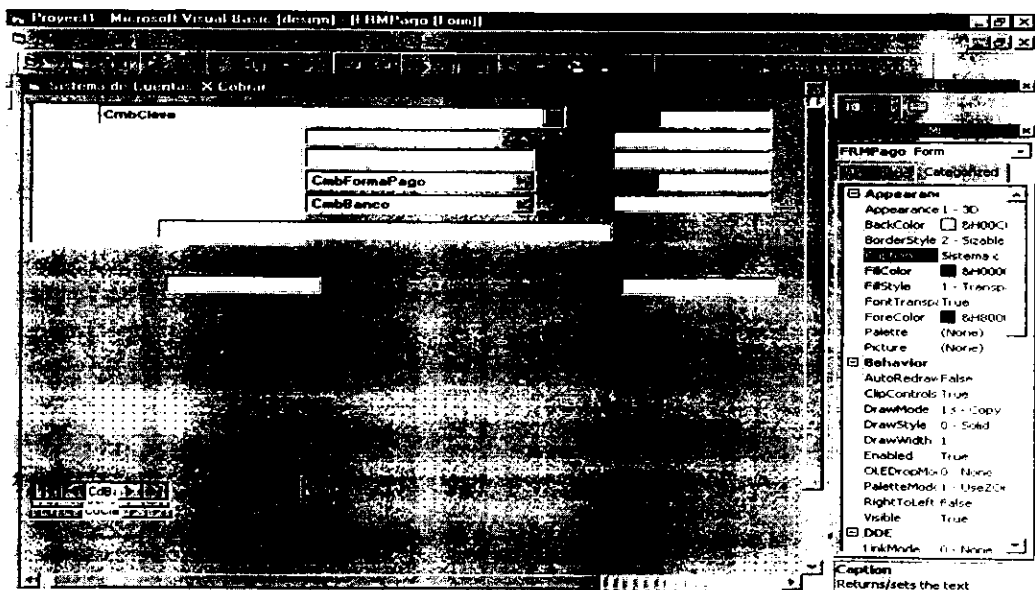
La pantalla se divide en dos partes. En la parte superior aparecerá una lista de clientes, de la cual se podrá seleccionar alguno. Una vez seleccionado el cliente se desplegará tanto su rfc como su dirección, además de pedir información adicional para el pago como: forma de pago, número de documento y banco si se trata de un cheque, y el importe a pagar.

En la parte inferior de la pantalla aparecerán todas las facturas que están pendientes por pagar. Existe un botón para ir seleccionado que facturas deseamos cubrir con este pago.

Por lo tanto la información que manejará esta pantalla será :

- Datos de clientes.
- Facturas pendientes por pagar
- Pagos realizados.

Diseño de la pantalla

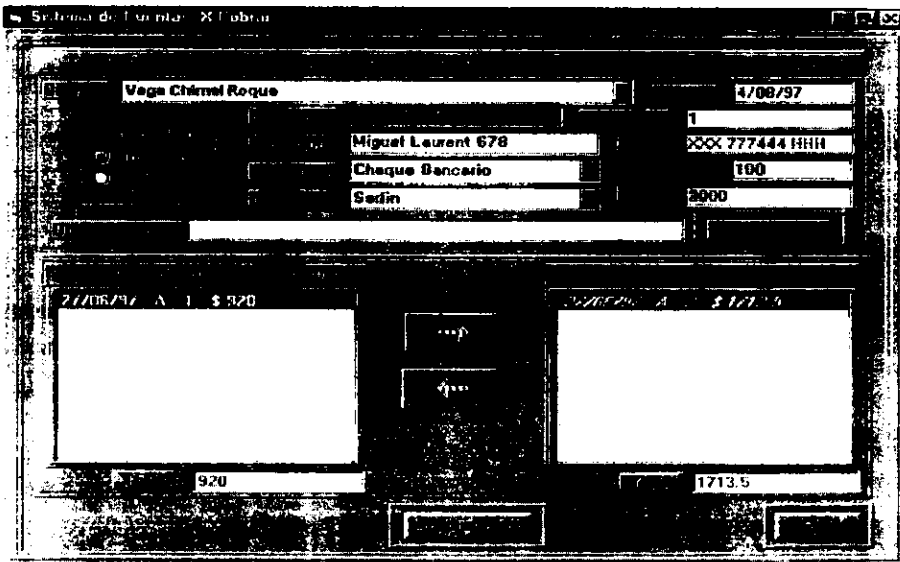


La pantalla contiene los siguientes controles:

- Un control de datos para realizar interfaz con la tabla de clientes
- Un control de datos para realizar interfaz con la tabla de facturas
- Un control de datos para realizar interfaz con la tabla de bancos
- Un control de datos para realizar interfaz con la tabla de formas de pagos.
- formas de pagos.

- Combo box asociado al control de datos de clientes para seleccionar un cliente específico.
- Combo box asociado al control de datos de formas de pago para seleccionar un forma de pago.
- Combo box asociado al control de datos de bancos para seleccionar un banco.
- Botón para seleccionar facturas a pagar.
- Caja de texto para desplegar facturas pendientes de pagar.
- Caja de texto para desplegar facturas seleccionadas para pagar.
- Caja de texto para capturar monto del pago.
- Cajas de texto para desplegar información del cliente seleccionado.

A continuación se presenta la cancelación de una factura.



3.3.2. DESARROLLO DE REPORTES DEL SISTEMA.

Introducción.

En este apartado se cubre el desarrollo de los Reportes del Sistema, de los cuales se incluyen los más representativos de cada módulo, a fin de mostrar la técnica utilizada en el desarrollo a través de la herramienta seleccionada.

De cada reporte incluido, se presenta el nombre, descripción, tablas utilizadas, sentencia SQL y un ejemplo de ejecución.

En forma tabular, quedará:

Nombre del Reporte	<i>Es el nombre del reporte en el sistema.</i>
Descripción	<i>Breve descripción y nombre del reporte al usuario.</i>
Tablas	<i>Principales tablas involucradas en el reporte.</i>
Sentencia(s) SQL	<i>Sentencia(s) SQL para obtener los resultado.</i>

La ejecución se incluye en hoja separada a fin de mostrar realmente la salida del reporte, incluyendo márgenes, tabulaciones, fonts, totales, etc.

RepAuxCMA

Nombre del Reporte	RepAuxCMA
Descripción	Reporte de Auxiliares de Cuentas de Mayor.
Tablas	centro_de_costos cuenta_clasificacion subcuenta subsubcuenta cuenta_mayor rep_aux_conta tipo_poliza
Sentencia(s) SQL	<pre> SELECT subsubcuenta.'ssc_scu_cco_cma_ccl_cve', subsubcuenta.'ssc_scu_cco_cma_cve', subsubcuenta.'ssc_scu_cco_cve', subsubcuenta.'ssc_scu_cve', subsubcuenta.'ssc_cve', subsubcuenta.'ssc_descripcion', subcuenta.'scu_descripcion', centro_de_costos.'cco_descripcion', cuenta_mayor.'cma_descripcion', rep_aux_conta.'rco_cve', rep_aux_conta.'rco_fecha_inicial', rep_aux_conta.'rco_fecha_final', rep_aux_conta.'rco_saldo', rep_aux_conta.'rco_debe', rep_aux_conta.'rco_haber', rep_aux_conta.'rco_debe_par', rep_aux_conta.'rco_haber_par', rep_aux_conta.'rco_concepto', rep_aux_conta.'rco_fecha_par', cuenta_clasificacion.'ccl_descripcion', tipo_poliza.'tpo_desc_corta' FROM (((('subsubcuenta' subsubcuenta LEFT OUTER JOIN 'centro_de_costos' centro_de_costos ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = centro_de_costos.'cco_cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = centro_de_costos.'cco_cma_cve' AND subsubcuenta.'ssc_scu_cco_cve' = centro_de_costos.'cco_cve') LEFT OUTER JOIN 'cuenta_mayor' cuenta_mayor ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = cuenta_mayor.'cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = cuenta_mayor.'cma_cve') INNER JOIN 'rep_aux_conta' rep_aux_conta ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = rep_aux_conta.'rco_ccl' AND subsubcuenta.'ssc_scu_cco_cma_cve' = rep_aux_conta.'rco_cma' AND subsubcuenta.'ssc_scu_cco_cve' = rep_aux_conta.'rco_cco' AND subsubcuenta.'ssc_scu_cve' = rep_aux_conta.'rco_scu' AND subsubcuenta.'ssc_cve' = rep_aux_conta.'rco_ssc') LEFT OUTER JOIN 'cuenta_clasificacion' cuenta_clasificacion ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = cuenta_clasificacion.'ccl_cve') LEFT OUTER JOIN 'subcuenta' subcuenta ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = subcuenta.'scu_cco_cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = subcuenta.'scu_cco_cma_cve' AND </pre>

	<pre>subsubcuenta.'ssc_scu_cco_cve' = subcuenta.'scu_cco_cve' AND subsubcuenta.'ssc_scu_cve' = subcuenta.'scu_cve') LEFT OUTER JOIN 'tipo_poliza' tipo_poliza ON rep_aux_conta.'rco_tipo' = tipo_poliza.'tpo_cve'</pre>
--	---

Sistema de Contabilidad
Casas PostProductoras de Video

Fecha: 13/08/97

Hoja

Nombre:

Dirección:

Reporte de Auxiliares Cuenta 01-02 del 01/01/97 al 21/07/97

Nombre Cuenta Tipo No. y FechaPol	Concepto del Movimiento	Cargos	Abonos	Saldos
01-02-000-00-00	Bancos TOTAL de Cargos y Abonos y Saldo	\$7,100.00	Saldo Inicial \$300.00	\$0.00 \$6,800.00
01-02-00 -00-00	Banco Central TOTAL de Cargos y Abonos y Saldo	\$7,100.00	Saldo Inicial \$300.00	\$0.00 \$6,800.00
01-02-001-01-00	Banco Internacional TOTAL de Cargos y Abonos y Saldo	\$7,100.00	Saldo Inicial \$300.00	\$0.00 \$6,800.00
01-02-001-01-01	Cuenta Núm. 12345643 TOTAL de Cargos y Abonos y Saldo	\$7,100.00	Saldo Inicial \$300.00	\$0.00 \$6,800.00
01-03-000-00-00	Almacén TOTAL de Cargos y Abonos y Saldo	\$14,489.00	Saldo Inicial \$7,964.00	\$0.00 \$6,535.00
01-03-00 -00-00	Almacén 1 TOTAL de Cargos y Abonos y Saldo	\$13,489.00	Saldo Inicial \$7,964.00	\$0.00 \$6,536.00
01-03-001-01-00	Materiales TOTAL de Cargos y Abonos y Saldo	\$13,489.00	Saldo Inicial \$7,964.00	\$0.00 \$6,531.00
01-03-001-02-00	Papelaria y Suministros TOTAL de Cargos y Abonos y Saldo	\$4.00	Saldo Inicial \$0.00	\$0.00 \$4.00
01-03-00 -00-00	Almacén 2 TOTAL de Cargos y Abonos y Saldo	\$1,000.00	Saldo Inicial \$0.00	\$0.00 \$1,000.00
01-03-002-01-00	Alimentos y Bebidas TOTAL de Cargos y Abonos y Saldo	\$1,000.00	Saldo Inicial \$0.00	\$0.00 \$1,000.00
01-04-000-00-00	Impuestos anticipados TOTAL de Cargos y Abonos y Saldo	\$2,013.00	Saldo Inicial \$0.00	\$0.00 \$2,013.00
01-04-00 -00-00	Impuestos Anuales TOTAL de Cargos y Abonos y Saldo	\$2,013.00	Saldo Inicial \$0.00	\$0.00 \$2,013.00
01-04-001-01-00	I.V.A. Acreditable TOTAL de Cargos y Abonos y Saldo	\$2,013.00	Saldo Inicial \$0.00	\$0.00 \$2,013.00
01-04-001-01-01	I.V.A. Acreditable TOTAL de Cargos y Abonos y Saldo	\$2,013.00	Saldo Inicial \$0.00	\$0.00 \$2,013.00

RepBalanzaComprobacion

Nombre del Reporte	<i>RepBalanzaComprobacion</i>
Descripción	<i>Reporte de Balanza de Comprobación por periodo..</i>
Tablas	<i>centro_de_costos subcuenta subsubcuenta cuenta_mayor rep_aux_conta</i>
Sentencia(s) SQL	<pre> SELECT subsubcuenta.'ssc_scu_cco_cma_ccl_eve', subsubcuenta.'ssc_scu_cco_cma_eve', subsubcuenta.'ssc_scu_cco_eve', subsubcuenta.'ssc_scu_eve', subsubcuenta.'ssc_eve', subsubcuenta.'ssc_descripcion', rep_aux_conta.'rco_fecha_inicial', rep_aux_conta.'rco_fecha_final', rep_aux_conta.'rco_saldo', rep_aux_conta.'rco_debe', rep_aux_conta.'rco_haber', cuenta_mayor.'cma_descripcion', subcuenta.'scu_descripcion', centro_de_costos.'cco_descripcion' FROM (((('subsubcuenta' subsubcuenta INNER JOIN 'rep_aux_conta' rep_aux_conta ON subsubcuenta.'ssc_scu_cco_cma_ccl_eve' = rep_aux_conta.'rco_ccl' AND subsubcuenta.'ssc_scu_cco_cma_eve' = rep_aux_conta.'rco_cma' AND subsubcuenta.'ssc_scu_cco_eve' = rep_aux_conta.'rco_cco' AND subsubcuenta.'ssc_scu_eve' = rep_aux_conta.'rco_scu' AND subsubcuenta.'ssc_eve' = rep_aux_conta.'rco_ssc') LEFT OUTER JOIN 'subcuenta' subcuenta ON subsubcuenta.'ssc_scu_cco_cma_ccl_eve' = subcuenta.'scu_cco_cma_ccl_eve' AND subsubcuenta.'ssc_scu_cco_cma_eve' = subcuenta.'scu_cco_cma_eve' AND subsubcuenta.'ssc_scu_cco_eve' = subcuenta.'scu_cco_eve' AND subsubcuenta.'ssc_scu_eve' = subcuenta.'scu_eve') LEFT OUTER JOIN 'centro_de_costos' centro_de_costos ON subsubcuenta.'ssc_scu_cco_cma_ccl_eve' = centro_de_costos.'cco_cma_ccl_eve' AND subsubcuenta.'ssc_scu_cco_cma_eve' = centro_de_costos.'cco_cma_eve' AND subsubcuenta.'ssc_scu_cco_eve' = centro_de_costos.'cco_eve') LEFT OUTER JOIN 'cuenta_mayor' cuenta_mayor ON subsubcuenta.'ssc_scu_cco_cma_ccl_eve' = cuenta_mayor.'cma_ccl_eve' AND subsubcuenta.'ssc_scu_cco_cma_eve' = cuenta_mayor.'cma_eve' </pre>

**Sistema de Contabilidad
Casas PostProductoras de Video**

Fecha: 13/08/97

Hoja: 1

Nombre:

Dirección:

Balanza de Comprobación del 01/01/97 al 21/07/97

Num.Cuenta	Descripción	Saldo Anterior	Cargos	Abonos	Saldo Actual
01-02-000-00-00	Bancos	\$0.00	\$7,100.00	\$300.00	\$6,800.00
01-02-001-00-00	Banco Central	\$0.00	\$7,100.00	\$300.00	\$6,800.00
01-02-001-01-00	Banco Internacional	\$0.00	\$7,100.00	\$300.00	\$6,800.00
01-02-001-01-01	Cuenta Núm. 12345643	\$0.00	\$7,100.00	\$300.00	\$6,800.00
01-03-000-00-00	Almacén	\$0.00	\$14,488.00	\$7,964.00	\$6,524.00
01-03-001-00-00	Almacén 1	\$0.00	\$13,488.00	\$7,964.00	\$5,524.00
01-03-001-01-00	Materiales	\$0.00	\$13,488.00	\$7,964.00	\$5,524.00
01-03-001-02-00	Papelaría y Suministros	\$0.00	\$4.00	\$0.00	\$4.00
01-03-002-00-00	Almacén 2	\$0.00	\$1,000.00	\$0.00	\$1,000.00
01-03-002-01-00	Alimentos y Bebidas	\$0.00	\$1,000.00	\$0.00	\$1,000.00
01-04-000-00-00	Impuestos anticipados	\$0.00	\$2,013.00	\$0.00	\$2,013.00
01-04-001-00-00	Impuestos Anuales	\$0.00	\$2,013.00	\$0.00	\$2,013.00
01-04-001-01-00	I.V.A. Acreditable	\$0.00	\$2,013.00	\$0.00	\$2,013.00
01-04-001-01-01	I.V.A. Acreditable	\$0.00	\$2,013.00	\$0.00	\$2,013.00
01-06-000-00-00	Cierras	\$0.00	\$31,588.16	\$13,432.48	\$18,155.68
02-01-000-00-00	Mobiliario y Equipo	\$0.00	\$0.00	\$1,000.00	\$(1,000.00)
02-01-001-00-00	Mobiliario y Equipo	\$0.00	\$0.00	\$1,000.00	\$(1,000.00)
02-01-001-03-00	Equipo de Transporte	\$0.00	\$0.00	\$1,000.00	\$(1,000.00)
06-01-000-00-00	Proveedores	\$0.00	\$0.00	\$16,433.00	\$(16,433.00)
06-01-001-00-00	Proveedores	\$0.00	\$0.00	\$16,433.00	\$(16,433.00)
06-02-000-00-00	Acreedores diversos	\$0.00	\$626.96	\$4,077.18	\$(3,250.22)
06-02-001-00-00	Impuestos a Pagar	\$0.00	\$626.96	\$4,077.18	\$(3,250.22)
06-02-001-03-00	I.V.A. al 15%	\$0.00	\$626.96	\$4,077.18	\$(3,250.22)
06-01-000-00-00	Ingresos por Tipo de Servicio	\$0.00	\$6,426.00	\$27,378.80	\$(20,952.80)
06-01-001-00-00	Rent	\$0.00	\$3,780.00	\$3,817.50	\$(37.50)
06-01-001-00-00	Henry	\$0.00	\$600.00	\$3,480.40	\$(2,880.40)
06-01-001-00-00	Paint Box	\$0.00	\$426.00	\$2,425.00	\$(1,999.00)
06-01-001-00-00	Capitales	\$0.00	\$498.00	\$6,875.90	\$(6,377.90)
06-01-001-00-00	Materiales	\$0.00	\$8.00	\$8,810.80	\$(8,802.80)
06-01-001-00-00	Producción	\$0.00	\$0.00	\$2,000.00	\$(2,000.00)
07-01-000-00-00	Gastos por Tipo de Servicio	\$0.00	\$71.60	\$0.00	\$71.60
07-01-001-00-00	Materiales	\$0.00	\$71.60	\$0.00	\$71.60
07-02-000-00-00	Costos de Materiales	\$0.00	\$7,980.00	\$66.00	\$7,914.00
07-02-001-00-00	Costos de Materiales	\$0.00	\$7,980.00	\$66.00	\$7,914.00
07-02-001-01-00	Costos de Materiales	\$0.00	\$7,980.00	\$66.00	\$7,914.00
Totales:		\$0.00	\$69,442.84	\$69,440.54	\$2.30

RepBanco

Nombre del Reporte	<i>RepBanco</i>
Descripción	<i>Reporte de Bancos.</i>
Tablas	<i>BAN</i>
Sentencia(s) SQL	<pre>SELECT BAN.'ban_nombre', BAN.'ban_direccion', BAN.'ban_responsable', BAN.'ban_telefono' FROM 'BAN' BAN</pre>

RepCifrasCtrl

Nombre del Reporte	<i>RepCifrasCtrl</i>
Descripción	<i>Reporte de Cifras de Control.</i>
Tablas	<p><i>pago</i> <i>forma_pago</i> <i>cliente</i> <i>banco</i></p>
Sentencia(s) SQL	<pre>SELECT pago.'pag_cve', pago.'pag_fecha', pago.'pag_monto', pago.'pag_num_docto', cliente.'cli_cve', cliente.'cli_nombre', forma_pago.'fop_desc_larga', banco.'ban_cve', banco.'ban_nombre' FROM (('pago' pago INNER JOIN 'forma_pago' forma_pago ON pago.'pag_fop_cve' = forma_pago.'fop_cve') INNER JOIN 'cliente' cliente ON pago.'pag_cli_cve' = cliente.'cli_cve') LEFT OUTER JOIN 'banco' banco ON pago.'pag_ban_cve' = banco.'ban_cve' ORDER BY pago.'pag_cve' ASC</pre>

Sistema de Contabilidad
Casas PostProductoras de Video

Fecha: 13/08/97

Hoja 1

Nombre:
Dirección:

Reporte de Bancos

Nombre	Dirección	Responsable	Teléfono
Serín	Av. Insurgentes 234	Lic. Raúl Rojas	5646546

Sistema de Contabilidad
Casas PostProductoras de Video

Fecha: 2/07/97

Hoja: 1

Cve Client	Cliente	Núm Recibo	Tipo Pago	Núm Documento	Banco	Moneda
22	Vega	1	Cheque Bancario	chq 123	Serfin	\$300.00
5	Dinámica	2	Pago en Efectivo	2		\$6,800.00
2	Barajas	3	Cheque Bancario	1	Serfin	\$10,000.00
2	Barajas	4	Pago en Efectivo	4		\$500.00
1	Aguilar	5	Cheque Bancario	435345	Serfin	\$5,000.00
1	Aguilar	6	Pago en Efectivo	6		\$6,000.00
1	Aguilar	7	Cheque Bancario	435345	Serfin	\$1,000.00
59	Alejandro Jimen	8	Pago en Efectivo	8		\$200.00
50	Boyer	9	Pago en Efectivo	9		\$300.00
1	Aguilar	10	Cheque Bancario	564	Serfin	\$7,000.00
1	Aguilar	11	Pago en Efectivo	11		\$45,666.00
31	CAMELIA SALGADO	12	Pago en Efectivo	12		\$54,545.00
5	Dinámica	13	Pago en Efectivo	13		\$200.00
5	Dinámica	14	Cheque Bancario	5454	Serfin	\$100.00
5	Dinámica	15	Pago en Efectivo	15		\$100.00
5	Dinámica	16	Pago en Efectivo	16		\$100.00
5	Dinámica	17	Pago en Efectivo	17		\$50.00
5	Dinámica	18	Pago en Efectivo	18		\$30.00
5	Dinámica	19	Pago en Efectivo	19		\$50.00
5	Dinámica	20	Pago en Efectivo	20		\$5.00
5	Dinámica	21	Pago en Efectivo	21		\$5.00
7	Flores	22	Pago en Efectivo	22		\$345.00
7	Flores	23	Pago en Efectivo	23		\$111.00

RepCuenta

Nombre del Reporte	<i>RepCuenta</i>
Descripción	<i>Reporte de Cuentas con Saldo anterior, por periodo.</i>
Tablas	<i>centro_de_costos</i> <i>cuenta_clasificacion</i> <i>subcuenta</i> <i>subsubcuenta</i> <i>cuenta_mayor</i> <i>rep_aux_conta</i>
Sentencia(s) SQL	<pre> SELECT subsubcuenta.'ssc_scu_cco_cma_ccl_cve', subsubcuenta.'ssc_scu_cco_cma_cve', subsubcuenta.'ssc_scu_cco_cve', subsubcuenta.'ssc_scu_cve', subsubcuenta.'ssc_cve', subsubcuenta.'ssc_descripcion', subcuenta.'scu_descripcion', centro_de_costos.'cco_descripcion', cuenta_mayor.'cma_descripcion', rep_aux_conta.'rco_fecha_inicial', rep_aux_conta.'rco_fecha_final', rep_aux_conta.'rco_saldo', rep_aux_conta.'rco_debe', rep_aux_conta.'rco_haber', cuenta_clasificacion.'ccl_descripcion' FROM (((('subsubcuenta' subsubcuenta LEFT OUTER JOIN 'centro_de_costos' centro_de_costos ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = centro_de_costos.'cco_cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = centro_de_costos.'cco_cma_cve' AND subsubcuenta.'ssc_scu_cco_cve' = centro_de_costos.'cco_cve') LEFT OUTER JOIN 'cuenta_mayor' cuenta_mayor ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = cuenta_mayor.'cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = cuenta_mayor.'cma_cve') INNER JOIN 'rep_aux_conta' rep_aux_conta ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = rep_aux_conta.'rco_ccl' AND subsubcuenta.'ssc_scu_cco_cma_cve' = rep_aux_conta.'rco_cma' AND subsubcuenta.'ssc_scu_cco_cve' = rep_aux_conta.'rco_cco' AND subsubcuenta.'ssc_scu_cve' = rep_aux_conta.'rco_scu' AND subsubcuenta.'ssc_cve' = rep_aux_conta.'rco_ssc') LEFT OUTER JOIN 'cuenta_clasificacion' cuenta_clasificacion ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = cuenta_clasificacion.'ccl_cve') LEFT OUTER JOIN 'subcuenta' subcuenta ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = subcuenta.'scu_cco_cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = subcuenta.'scu_cco_cma_cve' AND subsubcuenta.'ssc_scu_cco_cve' = subcuenta.'scu_cco_cve' AND subsubcuenta.'ssc_scu_cve' = subcuenta.'scu_cve' </pre>

Sistema de Contabilidad Casas PostProductoras de Video

Fecha: 13/08/97

Hoja: 1

Nombre:

Dirección:

Catálogo de Cuentas con Saldo Anterior del 01/01/97 al 21/07/97

Cuenta	Concepto	Cargos	Abonos	Saldos
01-00-000-00-00	Activo Circulante	\$66,160.16	\$21,686.48	\$33,473.68
01-02-000-00-00	Bancos	\$7,100.00	\$300.00	\$6,800.00
01-02-001-00-00	Banco Central	\$7,100.00	\$300.00	\$6,800.00
01-02-001-01-00	Banco Internacional	\$7,100.00	\$300.00	\$6,800.00
01-02-001-01-01	Cuenta Núm. 12345643	\$7,100.00	\$300.00	\$6,800.00
01-03-000-00-00	Almacén	\$14,489.00	\$7,954.00	\$6,535.00
01-03-001-00-00	Almacén 1	\$13,489.00	\$7,954.00	\$5,535.00
01-03-001-01-00	Materiales	\$13,485.00	\$7,954.00	\$5,531.00
01-03-001-02-00	Papelaria y Suministros	\$4.00	\$0.00	\$4.00
01-03-002-00-00	Almacén 2	\$1,000.00	\$0.00	\$1,000.00
01-03-002-01-00	Alimentos y Bebidas	\$1,000.00	\$0.00	\$1,000.00
01-04-000-00-00	Impuestos anticipados	\$2,013.00	\$0.00	\$2,013.00
01-04-001-00-00	Impuestos Anuales	\$2,013.00	\$0.00	\$2,013.00
01-04-001-01-00	I.V.A. Acreditable	\$2,013.00	\$0.00	\$2,013.00
01-04-001-01-01	I.V.A. Acreditable	\$2,013.00	\$0.00	\$2,013.00
01-05-000-00-00	Cuentas	\$31,558.16	\$13,432.48	\$18,125.68
02-00-000-00-00	Activo Fijo	\$0.00	\$1,000.00	\$(1,000.00)
02-01-000-00-00	Mobiliario y Equipo	\$0.00	\$1,000.00	\$(1,000.00)
02-01-001-00-00	Mobiliario y Equipo	\$0.00	\$1,000.00	\$(1,000.00)
02-01-001-03-00	Equipo de Transporte	\$0.00	\$1,000.00	\$(1,000.00)
05-00-000-00-00	Pasivo Diferido	\$825.98	\$19,510.16	\$(18,684.18)
05-01-000-00-00	Proveedores	\$0.00	\$15,433.00	\$(15,433.00)
05-01-001-00-00	Proveedores	\$0.00	\$15,433.00	\$(15,433.00)
05-02-000-00-00	Acreedores diversos	\$825.98	\$4,077.16	\$(3,251.18)
05-02-001-00-00	Impuestos a Pagar	\$825.98	\$4,077.16	\$(3,251.18)
05-02-001-03-00	I.V.A. al 15%	\$825.98	\$4,077.16	\$(3,251.18)
06-00-000-00-00	Ingresos	\$5,435.00	\$27,178.90	\$(21,743.90)
06-01-000-00-00	Ingresos por Tipo de Servicio	\$5,435.00	\$27,178.90	\$(21,743.90)
06-01-001-00-00	Rank	\$3,760.00	\$3,917.50	\$(157.50)
06-01-002-00-00	Henry	\$800.00	\$3,460.40	\$(2,660.40)
06-01-003-00-00	Paint Box	\$425.00	\$2,425.00	\$(2,000.00)
06-01-004-00-00	Copiado	\$450.00	\$5,876.00	\$(5,426.00)
06-01-005-00-00	Materiales	\$0.00	\$9,510.00	\$(9,510.00)
06-01-006-00-00	Producción	\$0.00	\$2,000.00	\$(2,000.00)
07-00-000-00-00	Gastos y Productos Financieros	\$8,021.50	\$65.00	\$7,956.50
07-01-000-00-00	Gastos por Tipo de Servicio	\$71.50	\$0.00	\$71.50
07-01-005-00-00	Materiales	\$71.50	\$0.00	\$71.50

RepDiarioGeneral

Nombre del Reporte	<i>RepDiarioGeneral</i>
Descripción	<i>Reporte Diario General a una fecha dada.</i>
Tablas	<i>subsubcuenta tipo_poliza rep_aux_conta poliza partida</i>
Sentencia(s) SQL	<pre> SELECT poliza.'pol_fecha', poliza.'pol_folio', poliza.'pol_concepto', partida.'par_consec', partida.'par_debe', partida.'par_haber', partida.'par_concepto', tipo_poliza.'tpo_desc_corta', subsubcuenta.'ssc_scu_cco_cma_ccl_cve', subsubcuenta.'ssc_scu_cco_cma_cve', subsubcuenta.'ssc_scu_cco_cve', subsubcuenta.'ssc_scu_cve', subsubcuenta.'ssc_cve', subsubcuenta.'ssc_descripcion', rep_aux_conta.'rco_fecha_final' FROM (((('poliza' poliza LEFT OUTER JOIN 'tipo_poliza' tipo_poliza ON poliza.'pol_tpo_cve' = tipo_poliza.'tpo_cve') LEFT OUTER JOIN 'partida' partida ON poliza.'pol_fecha' = partida.'par_pol_fecha' AND poliza.'pol_folio' = partida.'par_pol_folio') LEFT OUTER JOIN 'rep_aux_conta' rep_aux_conta ON tipo_poliza.'tpo_cve' = rep_aux_conta.'rco_cve') LEFT OUTER JOIN 'subsubcuenta' subsubcuenta ON partida.'par_ssc_scu_cco_cma_ccl_cve' = subsubcuenta.'ssc_scu_cco_cma_ccl_cve' AND partida.'par_ssc_scu_cco_cma_cve' = subsubcuenta.'ssc_scu_cco_cma_cve' AND partida.'par_ssc_scu_cco_cve' = subsubcuenta.'ssc_scu_cco_cve' AND partida.'par_ssc_scu_cve' = subsubcuenta.'ssc_scu_cve' AND partida.'par_ssc_cve' = subsubcuenta.'ssc_cve') ORDER BY poliza.'pol_fecha' ASC, poliza.'pol_folio' ASC, partida.'par_consec' ASC </pre>

Sistema de Contabilidad Casas PostProductoras de Video

Fecha 13/08/97

Hoja 1

Nombre:

Dirección:

Diario General al 21/07/97

Num. Cuenta	Descripción de la Cuenta		Concepto del Movimiento	Debe	Haber
De Diario	No.	1	Fecha: 27/06/97	Concepto Factura Serie: A no. 1	
01-05-000-00-00	Cuentas		Detalle de la Factura Serie: A No. 1	\$920.00	\$0.00
05-02-001-03-00	I.V.A. al 15%		Detalle de la Factura Serie: A No. 1	\$0.00	\$120.00
06-01-003-00-00	Pant. Box		Detalle de la Factura Serie: A No. 1	\$0.00	\$800.00
Total de la Póliza				\$920.00	\$920.00
De Diario	No.	2	Fecha: 27/06/97	Concepto Factura Serie: A no. 2	
01-05-000-00-00	Cuentas		Detalle de la Factura Serie: A No. 2	\$1,713.50	\$0.00
05-02-001-03-00	I.V.A. al 15%		Detalle de la Factura Serie: A No. 2	\$0.00	\$223.60
06-01-004-00-00	Copado		Detalle de la Factura Serie: A No. 2	\$0.00	\$450.00
06-01-002-00-00	Henry		Detalle de la Factura Serie: A No. 2	\$0.00	\$800.00
06-01-001-00-00	Rank		Detalle de la Factura Serie: A No. 2	\$0.00	\$240.00
Total de la Póliza				\$1,713.50	\$1,713.50
De Diario	No.	3	Fecha: 27/06/97	Concepto Entrada de Materiales. Orden: 1.	
01-03-001-01-00	Materiales		Detalle de la Orden de Materiales Num: 1	\$11,600.00	\$0.00
01-04-001-01-01	I.V.A. Acreditable		Detalle de la Orden de Materiales Num: 1	\$1,740.00	\$0.00
05-01-001-00-00	Proveedores		Detalle de la Orden de Materiales Num: 1	\$0.00	\$13,340.00
Total de la Póliza				\$13,340.00	\$13,340.00
De Diario	No.	4	Fecha: 27/06/97	Concepto Factura Serie: A no. 4	
01-05-000-00-00	Cuentas		Detalle de la Factura Serie: A No. 4	\$3,335.00	\$0.00
05-02-001-03-00	I.V.A. al 15%		Detalle de la Factura Serie: A No. 4	\$0.00	\$435.00
06-01-003-00-00	Pant. Bcv		Detalle de la Factura Serie: A No. 4	\$0.00	\$600.00
06-01-004-00-00	Copado		Detalle de la Factura Serie: A No. 4	\$0.00	\$400.00
06-01-006-00-00	Produccion		Detalle de la Factura Serie: A No. 4	\$0.00	\$1,000.00
06-01-005-00-00	Materiales		Detalle de la Factura Serie: A No. 4	\$0.00	\$900.00
Total de la Póliza				\$3,335.00	\$3,335.00
De Diario	No.	5	Fecha: 27/06/97	Concepto Materiales de la Factura Serie: A no. 4	
07-02-001-01-00	Costos de Materiales		Detalle de la Factura Serie: A No. 4	\$750.00	\$0.00
01-03-001-01-00	Materiales		Detalle de la Factura Serie: A No. 4	\$0.00	\$750.00
Total de la Póliza				\$750.00	\$750.00

RepPolCheque

Nombre del Reporte	<i>RepPolCheque</i>
Descripción	<i>Reporte de Polizas de Cheque.</i>
Tablas	<i>poliza</i> <i>partida</i> <i>subsubcuenta</i> <i>tipo_poliza</i>
Sentencia(s) SQL	<pre> SELECT poliza.'pol_fecha', poliza.'pol_folio', poliza.'pol_concepto', partida.'par_ssc_scu_cco_cma_ccl_cve', partida.'par_ssc_scu_cco_cma_cve', partida.'par_ssc_scu_cve', partida.'par_ssc_scu_cve', partida.'par_ssc_cve', partida.'par_debehaber', partida.'par_debe', partida.'par_haber', partida.'par_concepto', tipo_poliza.'tpo_desc_corta', subsubcuenta.'ssc_descripcion' FROM (('poliza' poliza INNER JOIN 'tipo_poliza' tipo_poliza ON poliza.'pol_tpo_cve' = tipo_poliza.'tpo_cve') INNER JOIN 'partida' partida ON poliza.'pol_fecha' = partida.'par_pol_fecha' AND poliza.'pol_folio' = partida.'par_pol_folio') INNER JOIN 'subsubcuenta' subsubcuenta ON partida.'par_ssc_scu_cco_cma_ccl_cve' = subsubcuenta.'ssc_scu_cco_cma_ccl_cve' AND partida.'par_ssc_scu_cco_cma_cve' = subsubcuenta.'ssc_scu_cco_cma_cve' AND partida.'par_ssc_scu_cco_cve' = subsubcuenta.'ssc_scu_cco_cve' AND partida.'par_ssc_scu_cve' = subsubcuenta.'ssc_scu_cve' AND partida.'par_ssc_cve' = subsubcuenta.'ssc_cve' </pre>

Concepto
Factura Serie: A no. 1

Firma Cheque Recibido

06/27/9

Cuent	Sub-Cuenta	Nombre	Parcial	Debe	Haber
1 5 0 0 0		Detalle de la Factura Serie: A No. 1 Clientes		\$ 920.00	
5 2 1 3 0		Detalle de la Factura Serie: A No. 1 I.V.A. al 15%			\$ 120.00
6 1 3 0 0		Detalle de la Factura Serie: A No. 1 Paint Box			\$ 800.00
Hecho por	Revisado por	Autorizado por	Diario	Auxiliares	Poliza No. 1 de De Diario

Concepto
Factura Serie: A no. 2

Firma Cheque Recibido

06/27/9

Cuent	Sub-Cuenta	Nombre	Parcial	Debe	Haber
1 5 0 0 0		Detalle de la Factura Serie: A No. 2 Clientes		\$ 1,713.50	
5 2 1 3 0		Detalle de la Factura Serie: A No. 2 I.V.A. al 15%			\$ 223.50
6 1 4 0 0		Detalle de la Factura Serie: A No. 2 Copiado			\$ 450.00
6 1 2 0 0		Detalle de la Factura Serie: A No. 2 Henry			\$ 800.00
6 1 1 0 0		Detalle de la Factura Serie: A No. 2 Rank			\$ 240.00
Hecho por	Revisado por	Autorizado por	Diario	Auxiliares	Poliza No. 2 de De Diario

RepPolDiario

Nombre del Reporte	<i>RepPolDiario</i>
Descripción	<i>Reporte de Polizas de Diario.</i>
Tablas	<i>poliza partida subsubcuenta tipo_poliza</i>
Sentencia(s) SQL	<pre> SELECT poliza.'pol_fecha', poliza.'pol_folio', poliza.'pol_concepto', partida.'par_consec', partida.'par_ssc_scu_cco_cma_ccl_cve', partida.'par_ssc_scu_cco_cma_cve', partida.'par_ssc_scu_cco_cve', partida.'par_ssc_scu_cve', partida.'par_ssc_cve', partida.'par_debehaber', partida.'par_debe', partida.'par_haber', partida.'par_concepto', Tipo_Poliza.'tpo_desc_corta', subsubcuenta.'ssc_descripcion' FROM (('poliza' poliza INNER JOIN 'Tipo_Poliza' Tipo_Poliza ON poliza.'pol_tpo_cve' = Tipo_Poliza.'tpo_cve') LEFT OUTER JOIN 'partida' partida ON poliza.'pol_fecha' = partida.'par_pol_fecha' AND poliza.'pol_folio' = partida.'par_pol_folio') LEFT OUTER JOIN 'subsubcuenta' subsubcuenta ON partida.'par_ssc_scu_cco_cma_ccl_cve' = subsubcuenta.'ssc_scu_cco_cma_ccl_cve' AND partida.'par_ssc_scu_cco_cma_cve' = subsubcuenta.'ssc_scu_cco_cma_cve' AND partida.'par_ssc_scu_cco_cve' = subsubcuenta.'ssc_scu_cco_cve' AND partida.'par_ssc_scu_cve' = subsubcuenta.'ssc_scu_cve' AND partida.'par_ssc_cve' = subsubcuenta.'ssc_cve' </pre>

**Sistema de Contabilidad
Casas PostProductoras de Video**

Fecha: 27/06/97

Hoja: 1

Concepto					
Factura Serie: A no. 1					

Cont	Sub-Cont	Nombre	Parcial	Debe	Haber
1 5	0 0 0	Detalle de la Factura Serie: A No. 1 Clientes		\$ 920.00	
5 2	1 3 0	Detalle de la Factura Serie: A No. 1 I.V.A. al 15%			\$ 120.00
6 1	3 0 0	Detalle de la Factura Serie: A No. 1 Paint Box			\$ 800.00

**Sistema de Contabilidad
Casas PostProductoras de Video**

Fecha: 27/06/97

Hoja: 1

Concepto					
Factura Serie: A no. 2					

Cont	Sub-Cont	Nombre	Parcial	Debe	Haber
1 5	0 0 0	Detalle de la Factura Serie: A No. 2 Clientes		\$ 1,713.50	
5 2	1 3 0	Detalle de la Factura Serie: A No. 2 I.V.A. al 15%			\$ 223.50
6 1	4 0 0	Detalle de la Factura Serie: A No. 2 Copiado			\$ 450.00
6 1	2 0 0	Detalle de la Factura Serie: A No. 2 Henry			\$ 800.00
6 1	1 0 0	Detalle de la Factura Serie: A No. 2 Rank			\$ 240.00

Hecho por	Revisado por	Autorizado por	Diario	Auxiliares	Poliza No	2	de	De Diario
-----------	--------------	----------------	--------	------------	-----------	---	----	-----------

**Sistema de Contabilidad
Casas PostProductoras de Video**

Fecha: 27/06/97

Hoja: 2

Concepto	
Entrada de Materiales. Orden: 1.	

Cuent	Sub-Cuenta	Nombre	Parcial	Debe	Haber
1 3	1 1 0	Detalle de la Orden de Materiales Núm:1 Materiales		\$ 11,600.00	
1 4	1 1 1	Detalle de la Orden de Materiales Núm:1 I.V.A Acreditable		\$ 1,740.00	
5 1	1 0 0	Detalle de la Orden de Materiales Núm:1 Proveedores			\$ 3,340.00
Sistema de Contabilidad Casas PostProductoras de Video				Fecha: 27/06/97	
				Hoja: 2	

Concepto	
Factura Serie: A no. 4	

Cuent	Sub-Cuenta	Nombre	Parcial	Debe	Haber	
1 5	0 0 0	Detalle de la Factura Serie: A No. 4 Clientes		\$ 3,335.00		
5 2	1 3 0	Detalle de la Factura Serie: A No. 4 I.V.A. al 15%			\$ 435.00	
6 1	3 0 0	Detalle de la Factura Serie: A No. 4 Paint Box			\$ 600.00	
6 1	4 0 0	Detalle de la Factura Serie: A No. 4 Copiado			\$ 400.00	
6 1	6 0 0	Detalle de la Factura Serie: A No. 4 Producción			\$ 1,000.00	
6 1	5 0 0	Detalle de la Factura Serie: A No. 4 Materiales			\$ 900.00	
Hecho por		Revisado por	Autorizado por	Diario	Auxiliares	Poliza No 4 de De Diario

RepPosibleCobranza

Nombre del Reporte	<i>RepPosibleCobranza</i>
Descripción	<i>Reporte de la Posible Cobranza a la fecha de ejecución.</i>
Tablas	<i>factura cliente rep_aux_cxc</i>
Sentencia(s) SQL	<pre> SELECT factura.`fac_serie`, factura.`fac_num`, factura.`fac_fecha`, factura.`fac_fecha_prob_pago`, factura.`fac_pago_pendiente`, cliente.`cli_nombre`, rep_aux_cxc.`rac_fecha` FROM ('factura' factura INNER JOIN 'rep_aux_cxc' rep_aux_cxc ON factura.`fac_fecha_prob_pago` <= rep_aux_cxc.`rac_fecha`) INNER JOIN 'cliente' cliente ON factura.`fac_cli_cve` = cliente.`cli_cve` ORDER BY factura.`fac_fecha_prob_pago` ASC, factura.`fac_serie` ASC, factura.`fac_num` ASC </pre>

RepRevision

Nombre del Reporte	<i>RepRevision</i>
Descripción	<i>Reporte de Facturas a Revision a una fecha dada.</i>
Tablas	<i>factura cliente rep_aux_cxc</i>
Sentencia(s) SQL	<pre> SELECT factura.`fac_serie`, factura.`fac_num`, factura.`fac_fecha`, factura.`fac_fecha_revision`, factura.`fac_fecha_prob_pago`, factura.`fac_pago_pendiente`, cliente.`cli_nombre`, rep_aux_cxc.`rac_fecha` FROM ('factura' factura INNER JOIN 'rep_aux_cxc' rep_aux_cxc ON factura.`fac_fecha_revision` = rep_aux_cxc.`rac_fecha`) INNER JOIN 'cliente' cliente ON factura.`fac_cli_cve` = cliente.`cli_cve` ORDER BY factura.`fac_fecha_prob_pago` ASC, factura.`fac_serie` ASC, factura.`fac_num` ASC </pre>

Sistema de Contabilidad
Casas PostProductoras de Video
 Posible Cobranza hasta el día: 25/11/9

Fecha: 14/08/9
 Hoja 1

Fecha PP	Factura	Fecha	Cliente	Monto
30/06/97	5	28/06/9	Flores	\$7,129 40
03/07/97	6	30/06/9	Super Alaska	\$4,048 00
10/07/97	13	10/07/9	CAMELIA SALGADO	\$54,555 00
27/07/97	1	27/06/9	Vega	\$0 00
27/07/97	2	27/06/9	Vega	\$1,713 50
28/07/97	3	28/06/9	Vega	\$570 98
29/07/97	4	27/06/9	Dinámica	\$100 00
08/08/97	7	7/07/97	Ibarrola	\$8,899 28
19/08/97	20	17/07/9	Dinámica	\$107,511 15
21/08/97	21	21/07/9	Barajas	\$0 00
21/08/97	22	21/07/9	Aguilar	\$0 00

Total : \$184,527 31

Sistema de Contabilidad
Casas PostProductoras de Video
Facturación a Revisión el día:

Fecha: 14/08/9

Hoja 1

Fecha Rev	Factura	Fecha	Cliente	Monto
	8	8/07/97	Aguilar	
	9	8/07/97	Dinámica	
	10	8/07/97	Dinámica	
	11	9/07/97	Urrieta	
	12	10/07/9	Garduño	
	14	11/07/9	Arturo Trejo	
	15	11/07/9	Arturo Trejo	
	16	11/07/9	Ibarrola	
	17	17/07/9	Rufino Lobeto	
	18	17/07/9	Vega	
	19	17/07/9	Boyer	
28/06/97	5	28/06/9	Flores	\$7,129.40
02/07/97	6	30/06/9	Super Alaska	\$4,048.00
10/07/97	13	10/07/9	CAMELIA SALGADO	\$54,555.00
27/06/97	1	27/06/9	Vega	\$0.00
27/06/97	2	27/06/9	Vega	\$1,713.50
28/06/97	3	28/06/9	Vega	\$570.98
27/06/97	4	27/06/9	Dinámica	\$100.00
07/07/97	7	7/07/97	Ibarrola	\$8,899.28
17/07/97	20	17/07/9	Dinámica	\$107,511.15
21/07/97	21	21/07/9	Barajas	\$0.00

Sistema de Contabilidad
Casas PostProductoras de Video
Facturación a Revisión el día:

Fecha: 14/08/9
Hoja 2

Fecha Rev	Factura	Fecha	Cliente	Monto
21/07/97	22	21/07/9	Agular	\$0 00

Total : \$184,527 31

RepSubSubCuenta

Nombre del Reporte	<i>RepSubSubCuenta</i>
Descripción	<i>Reporte de cuentas detallado a nivel de subsubcuenta..</i>
Tablas	<i>subsubcuenta partida cuenta_clasificacion subcuenta cuenta_mayor centro_de_costos</i>
Sentencia(s) SQL	<pre> SELECT subsubcuenta.'ssc_scu_cco_cma_ccl_cve', subsubcuenta.'ssc_scu_cco_cma_cve', subsubcuenta.'ssc_scu_cco_cve', subsubcuenta.'ssc_scu_cve', subsubcuenta.'ssc_cve', subsubcuenta.'ssc_tipo_cuenta', subsubcuenta.'ssc_descripcion', centro_de_costos.'cco_descripcion', cuenta_mayor.'cma_descripcion', subcuenta.'scu_descripcion', cuenta_clasificacion.'ccl_descripcion' FROM (((('subsubcuenta' subsubcuenta LEFT OUTER JOIN 'cuenta_mayor' cuenta_mayor ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = cuenta_mayor.'cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = cuenta_mayor.'cma_cve') LEFT OUTER JOIN 'subcuenta' subcuenta ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = subcuenta.'scu_cco_cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = subcuenta.'scu_cco_cma_cve' AND subsubcuenta.'ssc_scu_cco_cve' = subcuenta.'scu_cco_cve' AND subsubcuenta.'ssc_scu_cve' = subcuenta.'scu_cve') LEFT OUTER JOIN 'cuenta_clasificacion' cuenta_clasificacion ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = cuenta_clasificacion.'ccl_cve') LEFT OUTER JOIN 'centro_de_costos' centro_de_costos ON subsubcuenta.'ssc_scu_cco_cma_ccl_cve' = centro_de_costos.'cco_cma_ccl_cve' AND subsubcuenta.'ssc_scu_cco_cma_cve' = centro_de_costos.'cco_cma_cve' AND subsubcuenta.'ssc_scu_cco_cve' = centro_de_costos.'cco_cve' </pre>

Sistema de Contabilidad Casas PostProductoras de Video

Fecha: 14.08/97

Hoja: 1

Nombre:

Dirección:

Catálogo de SubSubCuenta

Cuenta	Descripción
01-00-000-00-00	T Activo Circulante
01-01-000-00-00	T Caja
01-02-000-00-00	T Bancos
01-02-001-00-00	T Banco Central
01-02-001-01-00	T Banco Internacional
01-02-001-01-01	D Cuenta Núm. 12345543
01-03-000-00-00	T Almacén
01-03-001-00-00	T Almacén 1
01-03-001-01-00	D Materiales
01-03-001-02-00	D Papelera y Suministros
01-03-002-00-00	T Almacén 2
01-03-002-01-00	D Alimentos y Bebidas
01-04-000-00-00	T Impuestos anticipados
01-04-001-00-00	T Impuestos Anuales
01-04-001-01-00	T I V A. Acreditable
01-04-001-01-01	D I V A Acreditable
01-05-000-00-00	D Clientes
02-00-000-00-00	T Activo Fijo
02-01-000-00-00	T Mobiliario y Equipo
02-01-001-00-00	T Mobiliario y Equipo
02-01-001-01-00	D Mobiliario y Equipo de Oficina
02-01-001-02-00	D Maquinaria y Equipo de Grabación
02-01-001-03-00	D Equipo de Transporte
02-01-001-04-00	D Equipo de Computo
02-01-001-05-00	D Herramientas y Accesorios de Trabajo
03-00-000-00-00	T Activo Diferido
04-00-000-00-00	T Pasivo Fijo
05-00-000-00-00	T Pasivo Diferido
05-01-000-00-00	T Proveedores
05-01-001-00-00	D Proveedores
05-02-000-00-00	T Acreedores diversos
05-02-001-00-00	T Impuestos a Pagar
05-02-001-01-00	D I S R
05-02-001-02-00	D 2% S A R
05-02-001-03-00	D I V A al 15%
05-03-000-00-00	T Documentos por pagar a corto plazo
05-04-000-00-00	T Anticipos
05-05-000-00-00	T Impuestos por pagar
05-06-000-00-00	T IVA por pagar
06-00-000-00-00	T Ingresos
06-01-000-00-00	T Ingresos por Tipo de Servicio
06-01-001-00-00	D Rank
06-01-002-00-00	D Henry
06-01-003-00-00	D Paint Box
06-01-004-00-00	D Copiado
06-01-005-00-00	D Materiales
06-01-006-00-00	D Producción
06-02-000-00-00	T Ingresos por Notas de Crédito
06-02-001-00-00	T Ingresos con Notas de crédito
06-02-001-01-00	D Nota de Crédito

**Sistema de Contabilidad
Casas PostProductoras de Video**

Fecha: 14/08/97

Hoja: 2

Nombre:

Dirección:

Catálogo de SubSubCuenta

Cuenta	Descripción
07-00-000-00-00 T	Gastos y Productos Financieros
07-01-000-00-00 T	Gastos por Tipo de Servicio
07-01-001-00-00 D	Rank
07-01-002-00-00 D	Henry
07-01-003-00-00 D	Paint Box
07-01-004-00-00 D	Copiado
07-01-005-00-00 D	Materiales
07-01-006-00-00 D	Producción
07-02-000-00-00 T	Costos de Materiales
07-02-001-00-00 T	Costos de Materiales
07-02-001-01-00 D	Costos de Materiales

RepCliente

Nombre del Reporte	<i>RepCliente</i>
Descripción	<i>Reporte del Catálogo de Clientes del Sistema.</i>
Tablas	<i>cliente</i> <i>entidad</i>
Sentencia(s) SQL	<pre> SELECT cliente.'cli_cve', cliente.'cli_rfc', cliente.'cli_razon_social', cliente.'cli_direccion', cliente.'cli_colonia', cliente.'cli_cp', cliente.'cli_telefono', entidad.'ent_desc_larga' FROM 'cliente' cliente INNER JOIN 'entidad' entidad ON cliente.'cli_ent_cve' = entidad.'ent_cve' ORDER BY cliente.'cli_cve' ASC </pre>

RepClienteAna

Nombre del Reporte	<i>RepClienteAna</i>
Descripción	<i>Reporte de Clientes del Sistema a detalle.</i>
Tablas	<i>cliente</i> <i>entidad</i> <i>día_semana</i>
Sentencia(s) SQL	<pre> SELECT cliente.'cli_cve', cliente.'cli_rfc', cliente.'cli_razon_social', cliente.'cli_nombre', cliente.'cli_direccion', cliente.'cli_colonia', cliente.'cli_delegacion', cliente.'cli_cp', cliente.'cli_telefono', cliente.'cli_fax', cliente.'cli_clasificacion', cliente.'cli_dia_pago', cliente.'cli_limite_credito', cliente.'cli_dias_credito', cliente.'cli_descuento', cliente.'cli_dia_revision', cliente.'cli_saldo_anticipado', cliente.'cli_saldo_pendiente', cliente.'cli_saldo_real', entidad.'ent_desc_larga' FROM 'cliente' cliente INNER JOIN 'entidad' entidad ON cliente.'cli_ent_cve' = entidad.'ent_cve' ORDER BY cliente.'cli_cve' ASC </pre>

**SISTEMA DE RESERVACIONES
CASAS POSTPRODUCTORAS DE VIDEO**

Catálogo de Clientes

Fecha: 19/05/97
Hoja: 1

Clave	Razón Social	R.F.C.	Dirección	Ciudad	Teléfono	C.P.	Entidad
1	Agua Nueva Felipe	AAA 123456 P	Laguna Nóm 127	Casa Blanca	5 07 64 56	06500	Chihuahua
2	Barreras Amezquita María	BBB 234567 M	Tehuacanopc 251	Doctores	5 42 3796	03456	Chihuahua
3	Carnes Seleadas Rocky	CCC 345678 N	Leona Vicario 10	Doc de Marzo	5 82 31 45	04400	Chihuahua
4	Chavez Martinez Marcelino	DDD 456789 O	Monte Alegre 45-A	Caracol	6 45 12 57	09989	Chihuahua
5	Dinamica Almenaca SA de CV	EEE 567890 Q	Honorio 530-34	El Globo	5 05 50 23	09970	Chihuahua
6	El Faro de la Colonia	FFF 678901 R	Central 12 Loc 11	Estrada del Sur	6 03 51 55	03700	Chihuahua
7	Flores Chavez Arturo	GGG 789012 S	Lomas Verdes 382	Flora Magón	5 11 53 66	06140	Chihuahua
8	Garduno Ruiz Dulce	HHH 890123 T	Galés 33 Int 34	Jalisco	6 35 90 11	01090	Chihuahua
9	Huerta Andrade Angel	JJJ 901234 UV	Playa Horizontes 31-3	Jardines Estrella	2 51 29 22	02400	Chihuahua
10	Huanda Malden Rocio	KKK 012345 V	Bosque de Duraznos 34	Jardines de Casa Nueva	6 44 21 33	03410	Chihuahua
11	Jurado Merza Antonia	LLL 102938 W	Navegantes y Fundadores 123	Joyas del Pedregal	2 51 29 44	02400	Chihuahua
12	La Cadena de Oro	MMM 209347	Av. Las Palmas 756	Juventud Unida	2 56 54 56	14050	Chihuahua
13	Mancera Barrera Eugenia	NNN 304756 Y	Sra. Ana No. 86	Lago Norte	6 06 94 66	01010	Chihuahua
14	Nicolás Campos Raúl	OOO 768594 Z	Península 2291	Lomas de Tarango	7 19 70 77	14310	Chihuahua
15	Ocampo Carmen Jasie	PPP 850403 A	Rancho del Arco Est 98	Micozac	6 28 23 88	06784	Chihuahua
16	Pachaza Mendoza Rogelio	QQQ 135791 B	Indagables Sur 1234	Morales	6 20 35 99	01080	Chihuahua
17	Quintana Contreras Armando	RRR 248802 C	Amsterdam 365-9	Nueva Anzures	5 87 34 77	02300	Chihuahua
18	Royes Telez Agustín	SSS 192837 D	Monterrey 566	San Rafael	6 44 68 00	14210	Chihuahua
19	Super Alasca	TTT 029384 E	Perifoneo 45	Parras	5 12 15 45	04326	Chihuahua
20	Torres Campos Hilario	UUU 186593 F	C. Deportiva Mabeuca	Pedregal de Tezompa	6 26 96 96	01029	Chihuahua
21	Uribe Chavez Marco	VVV 222999 G	Miguel Jacinto No. 366	Observatorio	2 99 91 23	11800	Chihuahua
22	Vega Chimal Roque	XXX 777444 H	Miguel Laurent 678	Plantilla	5 55 21 33	06760	Chihuahua
23	ETV PROD	mmmmmmmm	puerto 313-b	Roma			
24	Televisión Azteca	wwwwww					
25	Alasca Y ATI	ddddddddd					
26	Dorana Video						
27	Canal 7 de Televisión						
28	FERNANDO VILLEGAS BAHEÑA		Uruapan, Mich. Cupatitzio Núm 94	AMPULACION	4201319		
29	ALTEC ELECTRONICA CHIHUAMUA		PLAN DE AYALA Y RIVERA GRESPO LOCAL G	CHAPULTEPEC	2015000	31020	Chihuahua
30	Asociación Oaxaqueña de Televisión	distrofox	PERIFERICO DE LA ADVENTURO SM Cajalza Madero SIN seq. Av. Tecnológico	PARQUE INDUSTRIAL EL SAUCITO Centro Cultural Oaxaca	7610416 480-50 497-53 FX 44887 180942	14 292475	Oaxaca

**SISTEMA DE RESERVACIONES
CASAS POSTPRODUCTORAS DE VIDEO**
Catálogo de Clientes
Detallado

Fecha: 16/05/97
Hoja: 1

Clave	Razón Social	R.F.C.	Dirección	Colonia	Teléfono	C.P.	Entidad				
Nombre	Delegación	Fax	Clasificación	Día	Revisión	Día	Pago				
			Lim.	Credito	M %	S. Anticipos	S. Pendiente	S. Real			
1	Aguilar Anaya Felipe	AAA 123456 P	Legaria Nlm 127	Casa Blanca	5 07 64 58	06500	Chihuahua				
Aguilar	Avaro Oregón	5 07 64 58	Buena	Cuauquera	\$2,000,000.00	0	15	\$46,666.00	\$0.00	\$-46,666.00	
2	Barajas Amazquilla María	BBB 234567 M	Tehuantepec 251	Doctores	5 42 3796	03456	Chihuahua				
Barajas	Benito Juárez	5 42 3796	Buena	Lunes	Miércoles	\$3,000,000.00	0	0	\$0.00	\$10,000.00	\$805.00
3	Carnes Selecciones Rocky	CCC 345678 N	Leona Vicario 10	Dos de Marzo	5 82 31 45	04400	Chihuahua				
Carnes	Cuauhtemoc	5 82 31 45	Regular	Cuauquera	Cuauquera	\$4,000,000.00	0	0	\$0.00		
4	Chavez Martinez Marcelino	DDD 456789 O	Monte Alegre 45-A	Caracol	6 45 12 57	05889	Chihuahua				
Chavez	Coyacahn	6 45 12 57	Escalante	Martes	Viernes	\$5,000,000.00	0	0	\$0.00		
5	Dinámica Alimenticia SA de CV	EEE 567890 Q	Homero 330-34	El Globo	5 05 50 23	09970	Chihuahua				
Dinámica	Coahuila	5 05 50 23	Escalante	Cuauhtera	Jueves	\$6,000,000.00	0	0	\$0.00	\$107,611.15	\$107,611.15
6	El Faro de la Colonia	FFF 678901 R	Central 12 Loc 11	Estrella del Sur	6 03 51 55	03100	Chihuahua				
El Faro	Gustavo A. Madero	6 03 51 55	Escalante	Miércoles	Viernes	\$7,000,000.00	0	0	\$0.00		
7	Flores Chavez Arturo	GGG 789012 S	Lomas Verdes 302	Flores Margon	5 11 53 66	06140	Chihuahua				
Flores	Hidalgo	5 11 53 66	Regular	Viernes	Miércoles	\$8,000,000.00	0	10	\$0.00	\$7,129.40	\$7,129.40
8	Garduño Ruiz Dulce	HHH 890123 T	Gelali 33 Int 34	Jalapa	6 35 90 11	01090	Chihuahua				
Garduño	Tlalpan	6 35 90 11	Buena	Jueves	Jueves	\$9,000,000.00	0	0	\$0.00		
9	Huerta Andrade Angel	JJJ 901234 U	Playa Homillos 31-3	Granjas Estrella	2 51 29 22	02400	Chihuahua				
Huerta	Tehuacan	2 51 29 22	Escalante	Jueves	Jueves	\$500,000.00	0	0	\$0.00		
10	Ibarrola Milian Rocio	KKK 012345 V	Boaque de Duraznos 34	Jardines de Casa Nueva	6 44 21 33	02410	Chihuahua				
Ibarrola	Veracruz	6 44 21 33	Buena	Lunes	Viernes	\$600,000.00	0	0	\$0.00	\$8,864.78	\$8,864.78

RepMaquina

Nombre del Reporte	<i>RepMaquina</i>
Descripción	<i>Reporte Del inventario de Máquinas.</i>
Tablas	<i>maquina tipo_maquina</i>
Sentencia(s) SQL	<pre> SELECT maquina.'maq_cve', maquina.'maq_descripcion', tipo_maquina.'tma_descripcion' FROM 'maquina' maquina INNER JOIN 'tipo_maquina' tipo_maquina ON maquina.'maq_tma_cve' = tipo_maquina.'tma_cve' ORDER BY tipo_maquina.'tma_descripcion' ASC </pre>

RepMaquinaServicio

Nombre del Reporte	<i>RepMaquinaServicio</i>
Descripción	<i>Reporte Del inventario de Máquinas y servicios asociados.</i>
Tablas	<i>maquina maquina_servicio servicio</i>
Sentencia(s) SQL	<pre> SELECT servicio.'ser_cve', servicio.'ser_descripcion', servicio.'ser_precio', maquina.'maq_descripcion' FROM ('maquina_servicio' maquina_servicio INNER JOIN 'maquina' maquina ON maquina_servicio.'mas_maq_cve' = maquina.'maq_cve') INNER JOIN 'servicio' servicio ON maquina_servicio.'mas_ser_cve' = servicio.'ser_cve' ORDER BY maquina.'maq_descripcion' ASC </pre>

SISTEMA DE RESERVACIONES CASAS POSTPRODUCTORAS DE VIDEO

Fecha: 13/05/9

Hoja: 1

Catálogo de Máquinas

Tipo de Máquina: Abekas

Clave	Descripción
21	Máq Abekas 1

Tipo de Máquina: D1

Clave	Descripción
8	Máq. D1 4
5	Máq. D1 1
6	Máq. D1 2
7	Máq. D1 3

Tipo de Máquina: D2

Clave	Descripción
14	Máq. D2 2
15	Máq. D2 3
16	Máq. D2 4
13	Máq. D2 1

Tipo de Máquina: D3

Clave	Descripción
20	Máq. D3 4
17	Máq. D3 1
18	Máq. D3 2
19	Máq. D3 3

Tipo de Máquina: Lentes

Clave	Descripción
28	Zoom B
27	Zoom A

Tipo de Máquina: Luces

Clave	Descripción
34	Kit de luces portátil
33	Kit de luces foro
24	Luces No serie 1

Tipo de Máquina: Paint Box

Clave	Descripción
22	Máq Paint Box 1

Tipo de Máquina: Riel

Clave	Descripción
30	Riel curvo
29	Grua Viten Merlin

Tipo de Máquina: VHS

Clave	Descripción
1	Máq. VHS 1

**SISTEMA DE RESERVACIONES
CASAS POSTPRODUCTORAS DE VIDEO**

Fecha: 13-08/9

Hoja: 1

Catálogo de Máquinas-Servicios

Máquina: Betacam Sony

Clave	Descripción	Precio
61	Máquina adicional	567.657.00
59	Renta de foro	87.673.00
58	Paquete B	677.00
58	Paquete AAA	98.763.00

Máquina: Cámara Portátil 1

Clave	Descripción	Precio
53	Renta de Trípé	200.00
50	Renta de Cámara Portátil	2.500.00
58	Paquete AAA	98.763.00
54	Unidad móvil	9.876.00
51	Renta de Camioneta	250.00

Máquina: CCD

Clave	Descripción	Precio
58	Paquete AAA	98.763.00
59	Renta de foro	87.673.00
55	Paquete A	1.234.00
57	Paquete AA	8.765.00

Máquina: Grúa Viren Marin

Clave	Descripción	Precio
58	Paquete B	677.00
58	Paquete AAA	98.763.00
59	Renta de foro	87.673.00

Máquina: Kit de luces foro

Clave	Descripción	Precio
55	Paquete A	1.234.00
56	Paquete B	677.00
57	Paquete AA	8.765.00
58	Paquete AAA	98.763.00
59	Renta de foro	87.673.00

Máquina: Kit de luces portátil

Clave	Descripción	Precio
58	Paquete AAA	98.763.00

Máquina: Luces No serie 1

Clave	Descripción	Precio
54	Unidad móvil	9.876.00
53	Renta de Trípé	200.00
52	Renta de Equipo de Iluminación	100.00
51	Renta de Camioneta	250.00
58	Paquete AAA	98.763.00

RepOrden

Nombre del Reporte	<i>RepOrden</i>
Descripción	<i>Reporte de Ordenes de Trabajo.</i>
Tablas	<i>orden orden_servicio orden_servicio_maquina maquina servicio sala origen_material cliente tipo_trabajo operador proyecto</i>
Sentencia(s) SQL	<pre> SELECT orden.'ord_num', orden.'ord_res_num', orden.'ord_fecha', orden.'ord_solicitada_por', orden.'ord_orden_cliente', orden.'ord_descripcion', orden.'ord_autoriza', cliente.'cli_rfc', cliente.'cli_razon_social', cliente.'cli_direccion', cliente.'cli_colonia', cliente.'cli_delegacion', cliente.'cli_telefono', proyecto.'pry_desc_larga', tipo_trabajo.'tr_desc_larga', operador.'ope_nombre', orden_servicio.'ors_consec', orden_servicio.'ors_ser_eve', orden_servicio.'ors_no_eve', orden_servicio.'ors_fecha', orden_servicio.'ors_hr_ini_res', orden_servicio.'ors_hr_fin_res', orden_servicio.'ors_hr_ini_util', orden_servicio.'ors_hr_fin_util', orden_servicio.'ors_num_comerciales', orden_servicio.'ors_contenido', orden_servicio.'ors_num_copias', servicio.'ser_descripcion', servicio.'ser_precio', sala.'sal_desc_corta', orden_servicio_maquina.'osm_hr_ini_res', orden_servicio_maquina.'osm_hr_fin_res', orden_servicio_maquina.'osm_hr_ini_util', orden_servicio_maquina.'osm_hr_fin_util', maquina.'maq_descripcion' FROM (((((((orden' orden LEFT OUTER JOIN 'proyecto' proyecto ON orden.'ord_pry_eve' = proyecto.'pry_eve') LEFT OUTER JOIN 'tipo_trabajo' tipo_trabajo ON orden.'ord_tr_eve' = tipo_trabajo.'tr_eve') LEFT OUTER JOIN 'operador' operador ON orden.'ord_ope_num' = operador.'ope_num') INNER JOIN 'orden_servicio' orden_servicio ON orden.'ord_num' = orden_servicio.'ors_ord_num') LEFT OUTER JOIN 'cliente' cliente ON orden.'ord_cli_eve' = cliente.'cli_eve') INNER JOIN 'servicio' servicio ON orden_servicio.'ors_ser_eve' = servicio.'ser_eve') LEFT OUTER JOIN 'orden_servicio_maquina' orden_servicio_maquina ON orden_servicio.'ors_ord_num' = orden_servicio_maquina.'osm_ors_ord_num' AND orden_servicio.'ors_consec' = orden_servicio_maquina.'osm_ors_consec') LEFT OUTER JOIN 'sala' sala ON orden_servicio.'ors_sal_eve' = sala.'sal_eve') LEFT OUTER JOIN 'maquina' maquina ON orden_servicio_maquina.'osm_maq_eve' = maquina.'maq_eve' </pre>

Orden: 1
 Fecha orden: 26-06-97
 Fecha emisión: 13-08-97

CLIENTE : Vega Chimal Roque	PROYECTO : Comercial Coca-Cola
RFC : XXX 777444 HHH	TIPO TRABAJO : Copados
DIRECCIÓN : Miguel Laurent 67B	SOLICITADA POR :
COLONIA : Plenitud	ORDEN CLIENTE :
DELEGACIÓN : Ixtaro Obregón	DESCRIPCIÓN :
TELÉFONO : 5 55 21 33	AUTORIZADA POR :
RESERVACIÓN : 1	OPERADOR :

Salas

Fecha	Sala	Servicio	Inicio	Fin	Inicio	Fin	1/2 Hrs	Precio	Total
26/06/97	Sala de Paint	Servicio PaintBox	09:00	13:00	09:00	13:00	8	\$100.00	\$800.00
Máquinas:									
		Máq. Paint Box 1	09:00	13:00	09:00	13:00			

Recibimos y aceptamos a nuestra entera satisfacción los servicios y materiales aquí descritos, estando de acuerdo con los términos y condiciones de ambos lados de esta orden de trabajo. Asimismo, debemos y pagaremos incondicionalmente por este pagaré a la orden de Telerecibo producciones, S.A. de C.V., la cantidad de _____ por los servicios antes descritos.

Nombre: _____
 Firma: _____

Orden: 2
 Fecha orden: 26-06-97
 Fecha emisión: 13-08-97

CLIENTE : Vega Chimal Roque	PROYECTO :
RFC : XXX 777444 HHH	TIPO TRABAJO : Postproduccion
DIRECCION : Miguel Laurent 678	SOLICITADA POR :
COLONIA : Plentud	ORDEN CLIENTE :
DELEGACION : Ixtaro Obregon	DESCRIPCION :
TELEFONO : 5 55 21 33	AUTORIZADA POR :
RESERVACION : 6	OPERADOR :

Salas

Fecha	Sala	Servicio	Reservado		Utilizado		1/2 Hrs	Precio	Total
			Hr Inl	Hr Fin	Hr Inl	Hr Fin			
26-06-97	Sala Rank	Cine-Video Sin Corrector D1	15:30	16:30	15:30	16:30	2	\$120.00	\$240.00

Maquinas:

Maq. D1 1	15:30	16:30	15:30	16:30
Maq. D1 2	15:30	16:30	15:30	16:30

Fecha	Sala	Servicio	Reservado		Utilizado		1/2 Hrs	Precio	Total
			Hr Inl	Hr Fin	Hr Inl	Hr Fin			
26-06-97	Sala Copiado	Servicio Multiformato	16:30	19:00	16:30	19:00	5	\$90.00	\$450.00

Maquinas:

Maq. VHS 1	16:30	19:00	16:30	19:00
Maq. VHS 4	16:30	19:00	16:30	19:00

Fecha	Sala	Servicio	Reservado		Utilizado		1/2 Hrs	Precio	Total
			Hr Inl	Hr Fin	Hr Inl	Hr Fin			
26-06-97	Sala Henry	Sala digital con 3 VTRs D3	18:00	20:00	18:00	20:00	4	\$200.00	\$800.00

Maquinas:

Maq. D3 1	18:00	20:00	18:00	20:00
Maq. D3 3	18:00	20:00	18:00	20:00

Recibimos y aceptamos a nuestra entera satisfacción los servicios y materiales aquí descritos, estando de acuerdo con los términos y condiciones de ambos lados de esta orden de trabajo. Asimismo, debemos y pagaremos incondicionalmente por este pagaré a la orden de Telemidia producciones, S.A. de C.V. la cantidad de por los servicios antes descritos.

Nombre: _____
 Firma: _____

RepOperador

Nombre del Reporte	<i>RepOperador</i>
Descripción	<i>Reporte de Operadores.</i>
Tablas	<i>operador</i>
Sentencia(s) SQL	<pre>SELECT operador.'ope_num', operador.'ope_nombre', operador.'ope_rfc', operador.'ope_direccion', operador.'ope_colonia', operador.'ope_delegacion', operador.'ope_telefono' FROM 'operador' operador</pre>

RepOrdenEntradaMaterial

Nombre del Reporte	<i>RepOrdenEntradaMaterial</i>
Descripción	<i>Reporte de Ordenes de Entrada de Material.</i>
Tablas	<i>orden_entrada</i> <i>orden_entrada_proveedor</i> <i>proveedor</i> <i>material</i> <i>orden_entrada_material</i> <i>tipo_cambio</i> <i>tipo_moneda</i>
Sentencia(s) SQL	<pre>SELECT orden_entrada.'ore_num', orden_entrada.'ore_fecha', orden_entrada.'ore_recibido_por', orden_entrada.'ore_observaciones', orden_entrada_material.'oem_cantidad', orden_entrada_material.'oem_precio', orden_entrada_material.'oem_descuento', proveedor.'pro_razon_social', material.'mat_desclargas' FROM (((('orden_entrada' orden_entrada INNER JOIN 'orden_entrada_proveedor' orden_entrada_proveedor ON orden_entrada.'ore_num' = orden_entrada_proveedor.'oep_ore_num') INNER JOIN 'orden_entrada_material' orden_entrada_material ON orden_entrada.'ore_num' = orden_entrada_material.'oem_ore_num') INNER JOIN 'material' material ON orden_entrada_material.'oem_mat_cve' = material.'mat_cve') INNER JOIN 'proveedor' proveedor ON orden_entrada_proveedor.'oep_pro_cve' = proveedor.'pro_cve' ORDER BY orden_entrada.'ore_num' ASC</pre>

SISTEMA DE RESERVACIONES CASAS POSTPRODUCTORAS DE VIDEO

Fecha: 14/08/97
Hoja: 1

Catálogo de Operadores

Número| Nombre
1 | Reymundo Vazquez Rojo
2 | José Luis Martínez
3 | Miguel Solo Soils

R.F.C. | Dirección | Colonia
AAA123123SSS | Av. Obispo Mundial 234 | Del Valle
RVR321654PPP | Calle 24 Sur 23 | Roma
MSS567543MM | Cayco 123 Int 56 | Lindavista

Delegación | Teléfono
Benito Juárez | 6543487
Cuahutemoc | 5432345
Guillermo A. Madero | 3456789

ORDEN DE ENTRADA

Orden 1
 Fecha orden: 27/06/97
 Fecha emisión: 14/08/97

Proveedor: Proveedor de Productos de Arte S.A. de C.V.
 Recibido Por: Miguel Ramirez
 Observaciones:

Material	Cantidad	Precio	Descuento	Subtotal
VIDEOCASSETTE BETACAM 90 MIN.	30	\$ 70.00	0.00	\$ 2,100.00
VIDEOCASSETTE DIGITAL D2 120 MIN	35	\$ 180.00	0.00	\$ 6,300.00
VIDEOCASSETTE VHS 90 MIN	40	\$ 60.00	0.00	\$ 2,400.00
VIDEOCASSETTE BETACAM 120 MIN	10	\$ 80.00	0.00	\$ 800.00

Proveedor: Proveedor de Productos de Arte S.A. de C.V.
 Recibido Por: Martin Robles
 Observaciones:

Material	Cantidad	Precio	Descuento	Subtotal
VIDEOCASSETTE VHS 120 MIN	3	\$ 65.00	0.00	\$ 195.00
VIDEOCASSETTE BETACAM 120 MIN	1	\$ 80.00	0.00	\$ 80.00
VIDEOCASSETTE BETACAM 60 MIN.	10	\$ 60.00	0.00	\$ 600.00
VIDEOCASSETTE BETACAM 90 MIN.	5	\$ 70.00	0.00	\$ 350.00
VIDEOCASSETTE DIGITAL D3 90 MIN	32	\$ 340.00	0.00	\$ 10,880.00
VIDEOCASSETTE DIGITAL D2 120 MIN	4	\$ 180.00	10.00	\$ 648.00
VIDEOCASSETTE VHS 90 MIN	7	\$ 60.00	25.00	\$ 315.00
VIDEOCASSETTE VHS 60 MIN	34	\$ 55.00	0.00	\$ 1,870.00

Proveedor: Almacenes Soria S.A. de C.V.
 Recibido Por: rrrr
 Observaciones:

Material	Cantidad	Precio	Descuento	Subtotal
VIDEOCASSETTE BETACAM 120 MIN	1	\$ 80.00	0.00	\$ 80.00

RepOrdenSalidaMaterial

Nombre del Reporte	<i>RepOrdenSalidaMaterial</i>
Descripción	<i>Reporte de Ordenes de Salida de Material del Almacén.</i>
Tablas	<i>orden_salida_almacen orden_salida_material material cliente</i>
Sentencia(s) SQL	<pre> SELECT orden_salida_almacen.`osa_num`, orden_salida_almacen.`osa_ord_num`, orden_salida_almacen.`osa_fecha`, orden_salida_almacen.`osa_recibido_por`, orden_salida_almacen.`osa_observaciones`, orden_salida_material.`osl_cantidad`, orden_salida_material.`osl_precio`, orden_salida_material.`osl_descuento`, cliente.`cli_razon_social`, material.`mat_desclarga` FROM ((orden_salida_almacen`orden_salida_almacen` INNER JOIN `cliente` cliente ON orden_salida_almacen.`osa_cli_cve` = cliente.`cli_cve`) INNER JOIN `orden_salida_material` orden_salida_material ON orden_salida_almacen.`osa_num` = orden_salida_material.`osl_osa_num`) INNER JOIN `material` material ON orden_salida_material.`osl_mat_cve` = material.`mat_cve` ORDER BY orden_salida_almacen.`osa_num` ASC </pre>

ORDEN DE SALIDA

Orden 1
 Fecha orden: 27/06/97
 Fecha emisión: 14/08/97

Orden: 3
 Cliente: Dinámica Alimenticia SA de CV
 Recibido Por: Luis Rojo
 Observaciones:

Material	Cantidad	Precio	Descuento	Subtotal
VIDEOCASSETTE BETACAM 120 MIN	2	\$ 96.00	0.00	\$ 192.00
VIDEOCASSETTE VHS 90 MIN	4	\$ 72.00	0.00	\$ 288.00
VIDEOCASSETTE BETACAM 90 MIN.	5	\$ 84.00	0.00	\$ 420.00

Orden: 4
 Cliente: Flores Chávez Arturo
 Recibido Por: Ing. Gil
 Observaciones: Material en buen estado.

Material	Cantidad	Precio	Descuento	Subtotal
VIDEOCASSETTE BETACAM 120 MIN	4	\$ 96.00	0.00	\$ 384.00
VIDEOCASSETTE BETACAM 90 MIN.	5	\$ 84.00	0.00	\$ 420.00
VIDEOCASSETTE VHS 90 MIN	8	\$ 72.00	0.00	\$ 576.00

Orden: 7
 Cliente: Chavez Martínez Marcelino
 Recibido Por: mtf
 Observaciones:

Material	Cantidad	Precio	Descuento	Subtotal
VIDEOCASSETTE BETACAM 120 MIN	2	\$ 96.00	0.00	\$ 192.00
VIDEOCASSETTE BETACAM 30 MIN.	99	\$ 60.00	0.00	\$ 5,940.00

Orden: 9
 Cliente: Ibarrola Millán Rocio
 Recibido Por: ertert
 Observaciones:

Material	Cantidad	Precio	Descuento	Subtotal
VIDEOCASSETTE VHS 60 MIN	1	\$ 60.50	0.00	\$ 60.50

Recibimos y aceptamos a nuestra entera satisfacción los materiales aquí descritos, estando de acuerdo con los términos y condiciones de ambos lados de esta orden de trabajo. Asimismo, debemos y pagaremos incondicionalmente por este pagaré a la orden de Telemeda a producción, S.A. de C.V., la cantidad de _____ por los servicios antes descritos.

Nombre: _____
 Firma: _____

RepProveedor

Nombre del Reporte	<i>RepProveedor</i>
Descripción	<i>Reporte de Proveedores del Sistema de Almacén.</i>
Tablas	<i>proveedor</i>
Sentencia(s) SQL	<pre>SELECT proveedor.'pro_cve', proveedor.'pro_rfc', proveedor.'pro_razon_social', proveedor.'pro_nombre', proveedor.'pro_direccion', proveedor.'pro_colonia' FROM 'proveedor' proveedor</pre>

RepReservacion

Nombre del Reporte	<i>RepReservacion</i>
Descripción	<i>Reporte de Reservaciones de Salas y Máquinas.</i>
Tablas	<i>reservacion solicitud maquina_solicitud maquina servicio sala cliente status proyecto tipo_trabajo</i>
Sentencia(s) SQL	<pre> SELECT reservacion.`res_num`, reservacion.`res_fecha`, reservacion.`res_solicitada_por`, reservacion.`res_orden_cliente`, reservacion.`res_descripcion`, cliente.`cli_rfc`, cliente.`cli_razon_social`, cliente.`cli_direccion`, cliente.`cli_colonia`, cliente.`cli_delegacion`, cliente.`cli_telefono`, tipo_trabajo.`tr_desc_corta`, proyecto.`pry_desc_corta`, status.`sta_desc_corta`, solicitud.`sol_num`, solicitud.`sol_tso_cve`, solicitud.`sol_fecha`, solicitud.`sol_hr_ini`, solicitud.`sol_hr_fin`, solicitud.`sol_descripcion`, solicitud.`sol_num_copias`, solicitud.`sol_num_comerciales`, sala.`sal_desc_corta`, servicio.`ser_descripcion`, servicio.`ser_precio`, maquina_solicitud.`mso_hr_ini`, maquina_solicitud.`mso_hr_fin`, maquina.`maq_descripcion` FROM (((((((('reservacion' reservacion LEFT OUTER JOIN 'tipo_trabajo' tipo_trabajo ON reservacion.`res_tr_cve` = tipo_trabajo.`tr_cve`) LEFT OUTER JOIN 'proyecto' proyecto ON reservacion.`res_pry_cve` = proyecto.`pry_cve`) LEFT OUTER JOIN 'status' status ON reservacion.`res_sta_cve` = status.`sta_cve`) INNER JOIN 'solicitud' solicitud ON reservacion.`res_num` = solicitud.`sol_res_num`) LEFT OUTER JOIN 'cliente' cliente ON reservacion.`res_cli_cve` = cliente.`cli_cve`) INNER JOIN 'servicio' servicio ON solicitud.`sol_ser_cve` = servicio.`ser_cve`) LEFT OUTER JOIN 'maquina_solicitud' maquina_solicitud ON solicitud.`sol_num` = maquina_solicitud.`mso_sol_num`) LEFT OUTER JOIN 'sala' sala ON solicitud.`sol_sal_cve` = sala.`sal_cve`) LEFT OUTER JOIN 'maquina' maquina ON maquina_solicitud.`mso_maq_cve` = maquina.`maq_cve` ORDER BY </pre>

	reservacion.`res_num` ASC, solicitud.`sol_tso_cve` ASC, solicitud.`sol_num` ASC
--	---

Reservación: 1
 Fecha: 06/28/97
 Fecha emisión: 02/04/97

CLIENTE : Vega Chimal Roque
 RFC : XXX 777444 HHH
 DIRECCIÓN : Miguel Laurent 678
 COLONIA : Planitud
 DELEGACIÓN : Alvaro Obregon
 TELÉFONO : 5 55 21 33

PROYECTO : Coca-Cola
 TIPO TRABAJO : Copiado
 SOLICITADA POR :
 ORDEN CLIENTE :
 DESCRIPCIÓN :
 STATUS : Confirmada

Salas

Fecha	Sala	Servicio solicitado	De	A	1/2 Hr.	Precio	Total	Descripción
26/06/97	Sala de Paint	Servicio Paint Box	09:00	13:00	4	200.00	800.00	

MÁQUINAS.

Máq Paint Box 1	09:00	13:00
-----------------	-------	-------

Por medio de la presente solicito sean confirmados los servicios antes señalados en las fechas descritas.
ATENTAMENTE

Reservación: 2
 Fecha: 06/26/97
 Fecha emisión: 02/04/97

CLIENTE : Flores Chávez Arturo
 RFC : GGG 789012 SS
 DIRECCIÓN : Lomas Verdes 382
 COLONIA : Flores Magón
 DELEGACIÓN : Hidalgo
 TELÉFONO : 5 11 53 66

PROYECTO :
 TIPO TRABAJO : PostPro
 SOLICITADA POR :
 ORDEN CLIENTE :
 DESCRIPCIÓN :
 STATUS : Confirmada

Sq/os

Fecha	Sala	Servicio solicitado	De	A	1/2 Hr.	Precio	Total	Descripción
26/06/97	Sala Copiado	Servicio Multiformato	09:30	12:30	6	90.00	540.00	

MÁQUINAS:

Máq. VHS 1	09:30	12:30
Máq. VHS 4	09:30	12:30

Fecha	Sala	Servicio solicitado	De	A	1/2 Hr.	Precio	Total	Descripción
26/06/97	Sala							

MÁQUINAS:

Máq. Abekas 1	11:00	11:30
---------------	-------	-------

Fecha	Sala	Servicio solicitado	De	A	1/2 Hr.	Precio	Total	Descripción
26/06/97	Sala							

MÁQUINAS:

Máq. D3 4	14:00	15:30
Máq. D3 2	14:00	15:30

Por medio de la presente solicito sean confirmados los servicios antes señalados en las fechas descritas.
ATENTAMENTE

3.3.3. DEPURACIÓN, RESPALDO Y RECUPERACIÓN DE INFORMACIÓN.

Introducción.

El proceso de depuración de información consiste en eliminar de la base de datos de trabajo la información que no es utilizada para la operación diaria la cual abarca desde catálogos hasta información resultante de la utilización del sistema, como derivaciones contables, reportes grabados a disco, reservaciones ya operadas en su totalidad (reservado, operado, facturado, contabilizado, cobrado) y que para efectos de continuar la operación no es necesario seguir manteniendo esta información.

En el caso de respaldos, este procedimiento es necesario a fin de establecer planes de contingencias para la recuperación de la información (datos y programas). Siendo este proceso incremental o selectivo.

En el caso incremental se refiere a respaldar periódicamente la información que se pudiera acumular en un lapso de tiempo predeterminado: diario, semanal, mensual, trimestral, etc.

En el caso de respaldos selectivos, como su nombre lo indica, puede realizarse a través de elegir de la totalidad del sistema, qué información es vital que se respalde en algún medio (cinta, disco, CD, etc). Este proceso es mucho más complicado que el incremental puesto que requiere que haga un vaciado selectivo de cada una de las tablas importantes del sistema.

Adicionalmente, se pueden combinar estos dos procesos (incremental y selectivo) para agilizar los procesos de recuperación: en caso de solo dañarse un archivo o tabla, es posible elegir sólo la información respaldada a través del respaldo selectivo, o bien, si el daño es mayor, quizá sea conveniente hacer una recuperación total del sistema.

Situación Actual.

Como ya se mencionó en puntos anteriores, este sistema es parte de un sistema mayor y que por lo tanto, al establecer un esquema de respaldos y recuperación de información deberá contemplar cada uno de los módulos del sistema global a fin de preservar la integridad de la información en su totalidad, esto es, existe información que es derivada de un módulo a otro: facturación con cobranza, reservaciones con almacén, almacén con contabilidad, etc.

En este momento no existe un proceso de respaldos y recuperación selectiva de información y para efectos de contingencia, se propone realizar un respaldo semanal total de información y mensual en el caso de los programas de la aplicación.

La periodicidad está dada por los ciclos mismos del sistema: cierres mensuales. Adicionalmente, este proceso se basa en la derivación misma de los documentos legales exigibles por ley: la conservación de documentos (pólizas contables, fichas, estados financieros, etc) por un periodo determinado y que en el peor de los casos, es posible reconstruir la historia a través de estos documentos.

Procedimiento propuesto.

Como ya se mencionó anteriormente, no existe un procedimiento "puro" que resuelva todas las posibles contingencias, sino que es necesario realizar una mezcla de ambos.

También es necesario hacer un planteamiento global del proceso de respaldo y recuperación, contemplando cada uno de los módulos y subsistemas, llevando un registro adecuado y ágil de qué se respalda, cuándo se respalda, dónde se respalda, dónde se almacenan los medios utilizados para respaldo, cuál es la vida útil de los respaldos, quién respalda, etc.

Como se puede observar, este punto es vital en cualquier sistema y en cualquier empresa de la cual dependa la calidad y agilidad de la información y como tal deberá considerarse este proceso: como un proyecto que se sugiere iniciar en paralelo al sistema de producción, o bien sentar las bases durante el proceso de análisis y diseño para un proyecto futuro.

En el caso concreto que nos ocupa, el proyecto de Depuración, Respaldo y Recuperación de Información, será iniciado una vez concluidos los módulos operativos a fin de establecer una visión global del sistema. Es necesario recordar, de acuerdo a lo mencionado en puntos anteriores, que este sistema es parte de otros procesos, de los cuales recibe y deriva información, por lo que es necesario una revisión general para establecer la mejor política para depuración, respaldo y recuperación de información.

CAPITULO 4
PRUEBAS DEL SISTEMA

4. PRUEBAS DEL SISTEMA

4.1 EL PROCESO DE PRUEBAS

4.1.1 PRUEBAS UNITARIAS

Son las pruebas técnicas y funcionales que se realizan a los programas conforme se terminan de codificar. Estas finalizan cuando se ha probado la totalidad de las funciones de la aplicación y no se detectan más defectos.

Objetivo:

- Confirmar que el módulo o programa fue codificado correctamente.
- Determinar la funcionalidad de los programas o módulos desarrollados, es decir, que cada uno de ellos haga lo que tiene que hacer.
- Encontrar y corregir defectos de cada módulo o programa.

Entradas :

- Plan de trabajo
- Programa o módulo depurado
- Guía de planeación de pruebas
- Estándares
- Reglas de negocio
- Especificaciones funcionales
- Formato de pruebas
- Catálogo de pruebas técnicas

Salidas :

- Reporte de errores
- Código fuente y ejecutable libre de defectos.
- Boletín de liberación para pruebas de integración

Actividades de las pruebas unitarias

En la figura 4.1.1.A se muestra el flujo de las actividades que se deben realizar para poder concluir las pruebas.

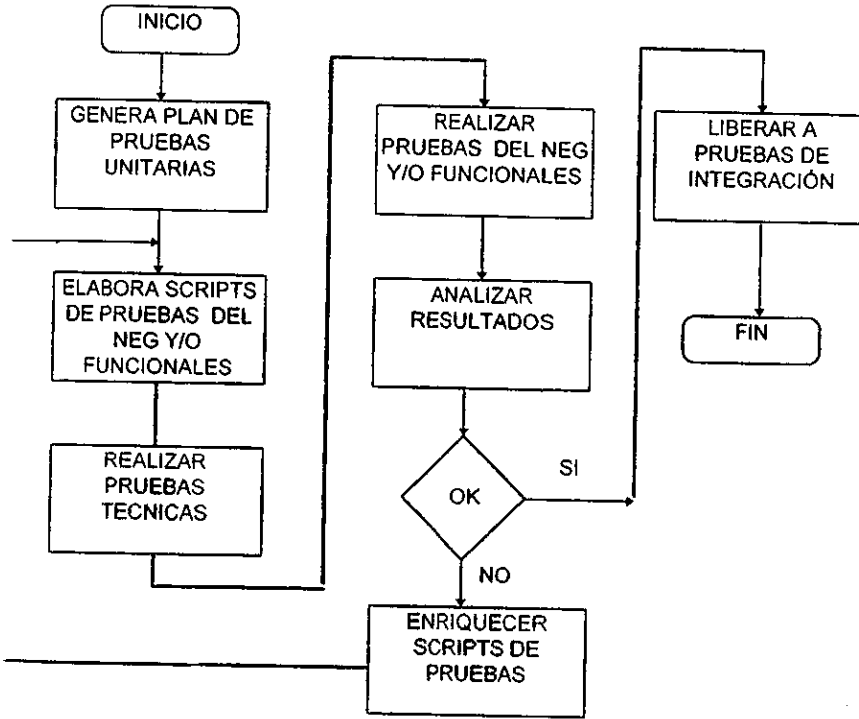


FIG. 4.1.1.A FLUJO DE ACTIVIDADES DEL PROCESO DE PRUEBAS UNITARIAS

Plan de pruebas unitarias

El plan de pruebas unitarias es un documento formal en el cual se especifican y calendarizan todas las actividades para el desarrollo de pruebas unitarias.

Objetivo

- Identificar piezas funcionales a probar.
- Dimensionar y calendarizar actividades.
- Identificar los recursos necesarios.

Entradas:

- Guía de planeación de la metodología
- Plan de trabajo del proyecto
- Formatos de plan de pruebas

Salida:

- Plan de pruebas

Elaborar scripts de pruebas de negocio y/o funcionales

Un script de prueba es un documento en el cual se tienen por escrito todos los casos de pruebas que contemplan las reglas del negocio y especificaciones funcionales que deberá de cumplir el módulo o programa.

Objetivo:

- Garantizar que los programas o módulos cumplan correcta y totalmente los requerimientos y especificaciones del usuario.

Entradas:

- Formatos de scripts
- Especificaciones funcionales
- Reglas del negocio.

Salida:

- Scripts de pruebas

Aplicar pruebas técnicas

Pruebas para garantizar la funcionalidad del módulo o programa sin considerar las especificaciones del usuario.

Objetivo:

- Asegurar el correcto funcionamiento de:
- Interfaces gráficas
- Transacciones de campo, registro, archivo, stress, control
- Recuperación y respaldo de información
- Reportes
- Caída y recuperación del equipo
- Procedimiento de instalación

Entradas:

- Código ejecutable.
- Catálogo de pruebas técnicas.
- Formato de reporte de errores.
- Estándares.

Salidas:

- Reporte de errores o
- Resultado satisfactorio

Pruebas del negocio/funcionales

Ejecutar las pruebas del negocio y funcionales guiadas por los scripts de pruebas previamente elaborados.

Objetivo:

- Asegurarse que el módulo o programa maneje adecuadamente las reglas del negocio

Entradas:

- Código compilado
- Scripts de prueba.
- Formato de reporte de errores.

Salidas:

- Resultado satisfactorio de pruebas del negocio ó
- Reporte de errores
- Reporte de herramienta de análisis dinámico

Liberar a pruebas de integración

Consiste en reunir la documentación, reportes de las pruebas realizadas y piezas de software.

Objetivo:

- Cumplir con los requisitos establecidos para concluir pruebas unitarias y comenzar pruebas de integración.

Entradas:

- Documentación y reportes de pruebas realizadas.
- Formato de solicitud de ambiente de pruebas

Salidas:

- Descripción de los programas que se liberan.
- Procedimientos de instalación.
- Procedimiento de operación.
- Manual técnico
- Manual de usuario
- Manual de operación
- Documento de especificaciones funcionales y reglas del negocio
- Scripts de pruebas

4.1.2 PRUEBAS DE INTEGRACIÓN

Es la prueba de ensamble de cada uno de los módulos de un sistema, es la etapa donde se integran todos los componentes los cuales interactúan en un medio ambiente controlado (laboratorio de pruebas). La prueba esta diseñada para descubrir las fallas e incompatibilidades entre componentes, además de detectar cualquier implementación incorrecta de las especificaciones. Para realizar estas pruebas se utilizará toda la documentación elaborada en la pruebas unitarias. En la fig. 4.1.2.A se muestra el flujo de actividades de estas pruebas.

Objetivo:

- Probar la integración de los módulos o programas
- Verificar las interfaces entre componentes y equipos
- Evaluar desempeño del sistema
- Asegurar que cada componente y su complemento cumplan con los requerimientos y especificaciones de diseño.

Entrada:

- Archivo de resultados de PU (Pruebas Unitarias)
- Documentación del Sistema (manuales)
- Documentación del diseño/requerimientos y especificaciones del usuario
- Scripts de prueba

Salida:

- Reporte de resultados de PI (Pruebas de Integración)
- Reporte de herramientas

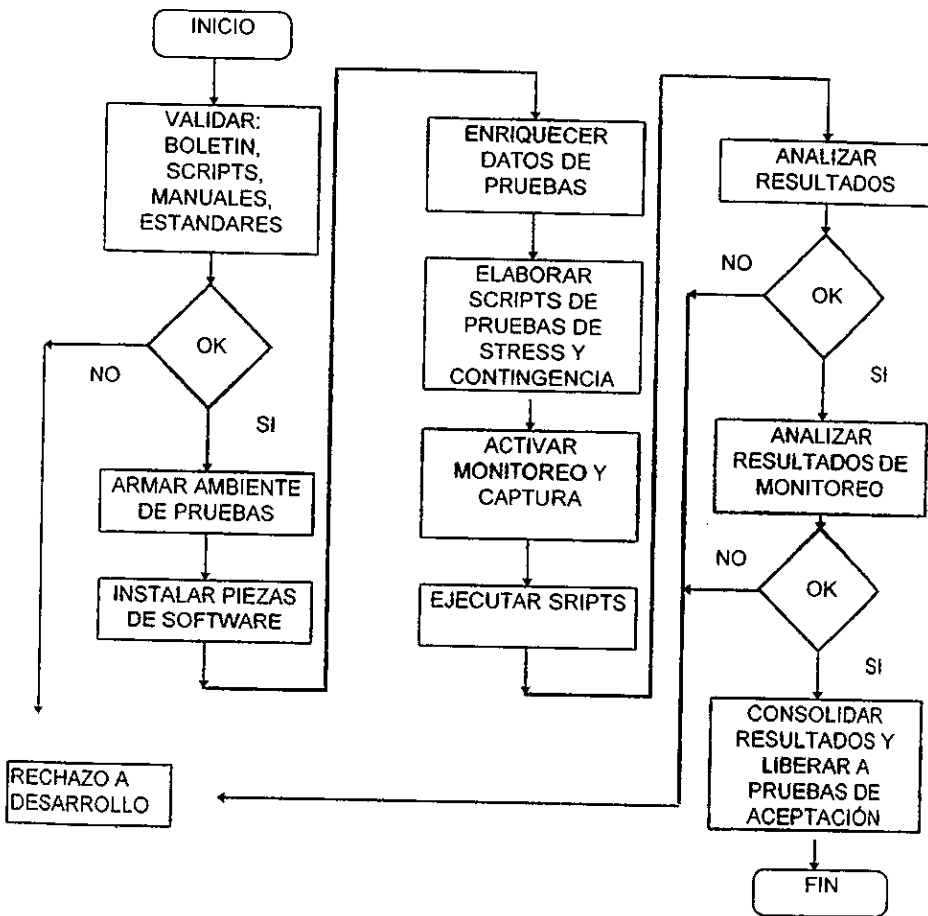


FIG. 4.1.2.A FLUJO DE ACTIVIDADES DEL PROCESO DE PRUEBAS DE INTEGRACIÓN.

Actividades de las pruebas de integración

Validar documentación

Actividad mediante la cual se revisan documentos como requisito previo a la etapa de integración.

Objetivo:

- Asegurarse que las actividades de pruebas unitarias se hayan realizado mediante la:
 - Validación de scripts para confirmar que se hayan realizado las pruebas unitarias que posteriormente se enriquecerán en esta etapa.
 - Validación de manuales: usuario, operación y técnico
 - Validación de estándares

Entrada:

- Scripts de pruebas del negocio
- Manuales de usuario, operación y técnico
- Check List de manuales
- Check List de estándares

Salida:

- Visto bueno para iniciar pruebas de integración.

Armar ambiente de pruebas

En el laboratorio de pruebas se instalan y configuran tanto hardware como software. El armado de ambientes contempla aplicaciones en PC y en servidor.

Objetivo:

- Reproducir el medio ambiente de producción a fin de detectar fallas potenciales en las aplicaciones anticipadamente.

Entradas:

- Solicitud de ambientes de pruebas (si es necesario)
- Boletín de liberación

Salida:

- Ambiente para la prueba

Instalar piezas de software

Instalar las piezas que se probarán siguiendo los procedimientos definidos en el manual de instalación.

Objetivo:

- Detectar las posibles fallas y problemas que puedan surgir en la instalación de las piezas de software antes de su distribución.

Entradas:

- Manual de instalación.
- Medio magnético de instalación (diskettes)

Salida:

- Reporte de instalación.

Generar y capturar datos para pruebas

Obtención de datos a utilizar para la realización de la prueba en base a documentos fuentes con apego a los scripts del negocio, una vez obtenidos los datos, estos se capturan.

Objetivo:

- Generar y capturar las transacciones con apego a los scripts del negocio

Entradas:

- Documentos fuentes
- Scripts de Pruebas del Negocio

Salida:

- Archivos de scripts para la realización de la prueba

Elaborar scripts de pruebas de stress y de contingencia

Un script es un documento en el cual se tienen por escrito todos los casos de pruebas que contemplan las transacciones de stress como situaciones imprevistas, que deberá cumplir el módulo o programa.

Objetivo:

- Provocar situaciones de carga de trabajo y/o anormales (aleatorias) para observar el comportamiento de la aplicación.

Entradas:

- Formatos de scripts
- Considerar los siguientes casos de prueba:
 - Copia de archivos
 - Generación de respaldos
 - Interpretación de mensajes
 - Generación de productos (archivos y reportes)
 - Reinicios y recuperaciones
 - Validar casos técnicos de control
 - Validar casos técnicos de stress
 - Validar casos técnicos de comunicaciones

Salida:

- Los scripts de pruebas

Activar monitoreo

Consiste básicamente en cargar los componentes de software con las herramientas de monitoreo para analizar el comportamiento de la aplicación.

Objetivo:

- Activar herramientas y aplicación con la finalidad de simular procesos reales de carga de trabajo.

Entradas:

- Manual de operación de la(s) herramienta(s) de monitoreo y de la aplicación
- Datos de pruebas

Salida:

- Reporte de herramienta(s)

Pruebas de stress - contingencia

Ejecutar las pruebas de stress y de contingencia guiadas por los scripts de pruebas previamente elaborados

Objetivo:

- Asegurarse que la aplicación maneje adecuadamente los volúmenes de información, y las situaciones anormales que se pudieran presentar en producción.
- Validar las medidas y planes de contingencias.

Entradas:

- Aplicación activada
- Monitor
- Transacciones (de acuerdo a los scripts elaborados)

Salidas:

- Archivos generados por la herramienta de monitoreo y de Captura y Repetición.
- Archivos generados por la aplicación
- Resultado satisfactorio
- Reporte de defectos (si existen)

Validar Transacciones

En base al archivo de respuesta de transacciones determinaremos si las respuestas generadas por la aplicación son las adecuadas.

Objetivo:

- Validar que se cumpla y se conserve la consistencia de la información bajo circunstancias anormales y de excesiva carga de trabajo.

Entradas:

- Archivo(s) de respuesta de transacciones

Salida:

- Vo. Bo. de las pruebas transaccionales
- Reporte de defectos (si existen)

Analizar monitoreo

Se hace uso de las herramientas de monitoreo para analizar los resultados de la ejecución de la aplicación.

Objetivos:

- Cuantificar el desempeño y consumo de recursos ante situaciones anormales y de carga excesiva de trabajo.

Entradas:

- Resultados de la herramienta monitoreo

Salidas:

- Listados generados por la herramienta
- Vo. Bo. de Análisis de PI

Consolidar resultados y liberar a pruebas de aceptación

Una vez siendo satisfactorias las pruebas de integración, se procede a recapitular información para consolidar los resultados de todas las actividades y así después promover el módulo o programa a la siguiente etapa.

Objetivo:

- Promover el módulo o programa al siguiente nivel de prueba

- Registrar el Boletín a la siguiente fase de pruebas
- Consolidar resultados para la certificación de sistemas

Entrada:

- Vo. Bos. generados durante la prueba
- Reportes de las herramientas utilizadas
- Formato de resultado de PI

Salida:

Reporte consolidado de resultados

4.1.3 PRUEBAS DE ACEPTACIÓN

Son las pruebas que realiza el usuario donde verifica que la aplicación fue desarrollada conforme a sus requerimientos y para validar que ésta se comportará adecuadamente cuando sea instalada en producción.

Objetivos :

- Que el usuario conozca y valide funcionalmente el sistema que solicitó, utilizando para ello el manual de usuario.
- Obtener el Visto Bueno del usuario
- Delimitar alcance de versión y formulación de nuevos requerimientos

Entradas:

- Aplicaciones con un nivel óptimo de prueba
- Manual de Usuario
- Manual de Operación

Salidas:

- Vo. Bo. de Usuario donde acepta la aplicación

Actividades de las pruebas de aceptación

Activar aplicación y ejecutar scripts de pruebas del negocio

Proceso mediante el cual se activa la aplicación y se comienzan a validar el diseño y los scripts de pruebas del negocio de acuerdo al requerimiento inicial.

Objetivos:

- Validación de cada una de las reglas del negocio
- Validación de la interface gráfica

Entrada:

- Manual de usuario
- Formatos de scripts
- Scripts de pruebas del negocio
- Documento de requerimiento de usuario

Salida:

- Reporte de errores (si existen).
- Resultado satisfactorio

Validar cifras y reportes

En este proceso se generan transacciones sobre la base de datos , por primera vez o por modificación para posteriormente ver reflejadas las transacciones en un medio impreso o de consulta.

Objetivo:

- Validar los resultados obtenidos después de algún evento de alta o modificación de la información (consistencia) y los cálculos originados por una transacción.

Entrada:

- Catálogo de casos técnicos

Salida:

- Vo. Bo. del usuario donde acepta la aplicación
- *Check-List* de cifras con resultados aprobados

4.2 PLAN DE PRUEBAS DEL SISTEMA

El sistema integral de contabilidad es una herramienta amigable para automatizar todo el proceso de administración y contabilidad de una compañía postproductora de cine y vídeo. Sus principales características son :

- Amigable
- Ambiente gráfico Windows
- Puede funcionar en una sola estación de trabajo o en varias estaciones conectadas en red compartiendo datos a través de un servidor.

El sistema consta de dos módulos: el módulo de contabilidad y el de cuentas por cobrar.

Este sistema se puede integrarse a otros dos sistemas que se encuentran trabajando actualmente . Estos sistemas son los sistemas de reservación de salas y el de almacén.

4.2.1 OBJETIVOS DE LAS PRUEBAS

1. De una manera general revisar minuciosamente el sistema en todos sus niveles para corregir posibles fallas antes de su salida a producción.
2. Encontrar y corregir todos los errores, tanto técnicos como funcionales.
3. Poder entregar un producto de alta calidad.

4.2.2 ALCANCES

Contabilidad

Este módulo contará con las siguientes características:

1. Control de acceso al sistema a través de una clave y una contraseña.
2. Contemplará altas, bajas, modificaciones y consultas de los siguientes catálogos
 - Clasificación de cuentas
 - Cuentas de mayor
 - Centros de costos
 - Subcuentas
 - Subsubcuentas
 - Areas de trabajo
 - Grupos
 - Usuarios
2. Realizará altas, bajas, modificaciones y consultas de las siguientes pólizas:
 - Tipo de póliza
 - Póliza modelo
 - Póliza modelo cuenta
3. Contemplará ajustes de pólizas
4. Se emitirán los siguientes reportes
 - Auxiliares por nivel de cuenta
 - Diario general
 - Balanza de comprobación
 - Balanza de comprobación por cuenta de mayor
 - Auxiliares por cuenta
 - Cuentas con saldo
5. Contará con cierre cada mes.
6. Tendrá captura y emisión de reportes de estados financieros

7. Tendrá captura y emisión de reportes sobre balance general.

Este módulo no contará con las siguientes características:

1. Emisión de reportes con gráficas para toma de decisiones.
2. Depuración automática de la información en cada cierre..
3. Respaldo y restauración automática de información en cada cierre.

4. Emisión de reportes de los catalogos :

Clasificación de cuentas

Centros de costos

Subcuentas

Areas de trabajo

Tipos de pólizas

Usuarios

Grupos

Cuentas por cobrar

Este módulo contará con las siguientes características:

1. Control de acceso al sistema a través de una clave y una contraseña.
2. Contemplará altas, bajas, modificaciones y consultas de los siguientes catálogos

Cientes

Bancos

3. Realizará control de pagos tanto normales como anticipados

4. Hará ajustes a diferentes movimientos como :

Cancelación de facturas

Pagos no considerados

Cancelación de pago mal aplicado por cheque devuelto o por error

5. Se emitirán los siguientes reportes

Cifras de control
Reportes
Estado de cuenta general
Pronostico de cobranza
Pronostico de revisión
Saldo vencido
Antigüedad de saldos
Pagos realizados
Facturas

Este módulo no contará con las siguientes características:

1. Emisión de reportes con gráficas para toma de decisiones.
2. Depuración automática de la información en cada cierre..
3. Respaldo y restauración automática de información en cada cierre.
4. Emisión de reportes de los siguientes catálogos :
 - Clientes
 - Bancos

4.2.3 TABLA DE ROLES Y RESPONSABILIDADES

Nombre	Roles
Blanca Gil Castellanos	Desarrollador y probador
Mario Carrasco Carrasco	Desarrollador y probador
Edgar García Adame	Desarrollador y probador

4.2.4 LISTA DE PROGRAMAS

No.	Nombre
1.	<p>Módulo de contabilidad</p> <p>Contabilidad.exe</p>
2.	<p>Módulo de cuentas por cobrar</p> <p>CtaXCobrar.exe</p>
3.	<p>Librerías Dinámicas miscelaneas</p> <p>MSVBVM50.dll, StdOle2.tlb,OleAut32.dll,OlePro32.dll,AsycFilt.dll, Ctl3d32.dll,ComCat.dll, DBLIST32.OCX, DAO350.DLL, MSJtEr35.dll, MSJInt35.dll, MSVCRT40.dll, THREED32.OCX, MFC40.dll, TDBG32.OCX, MSMASK32.OCX, VCF132.OCX:</p>
4.	<p>Librerías dinámicas para realizar reportes.</p> <p>CRYSTL32.OCX, CRPE32.DLL, IMplode.DLL, PG32.DLL, CO2C40EN.DLL, MSVCRT20.DLL, CRXLAT32.DLL, P2BBND.DLL, P2BDAO.DLL, P2CTDAO.DLL, P2IRDAO.DLL, P2SODBC.DLL, U2DDISK.DLL, U2DMAPI.DLL, U2FCR.DLL, U2FDIF.DLL, U2FHTML.DLL, U2FREC.DLL, U2FRTF.DLL, U2FSEPV.DLL, U2FTEXT.DLL, U2FWKS.DLL, U2FWORDW.DLL, U2FXLS.DLL</p>
5.	<p>Librerías dinámicas para realizar queries hacia la base de datos</p> <p>MSJet35.dll, VBAJet32.dll, VB5DB.dll,MSRD2x35.dll, MsRep135.dll, ODBCJt32.dll, ODBCJl32.dll, ODBCtl32.dll</p>

4.2.5 LISTA DE PANTALLAS

Contabilidad

No.	Nombre
1.	Pantalla principal MDI (Multiple document interface - Interfaz de multiples documentos) la cual da el acceso al sistema y contiene el menu principal del módulo.
2.	Pantalla para actualizaciones de clasificación de cuentas
3.	Pantalla para actualizaciones de cuentas de mayor
4.	Pantalla para actualizaciones de centros de costos
5.	Pantalla para actualizaciones de subcuentas
6.	Pantalla para actualizaciones de áreas de trabajo
7.	Pantalla para actualizaciones de grupos
8.	Pantalla para actualizaciones de usuarios
9.	Pantalla para realizar actualizaciones de pólizas
10.	Pantalla para realizar ajustes de pólizas
11.	Pantalla para realizar cierre contable
12.	Pantalla para captura de estados financieros
13.	Pantalla para captura de balance general

Cuentas por cobrar

No.	Nombre
1.	Pantalla principal MDI (Multiple document interface - Interfaz de multiples documentos) la cual da el acceso al sistema y contiene el menu principal del módulo.
2.	Pantalla para actualizaciones de clientes
3.	Pantalla para actualizaciones de bancos
4.	Pantalla para control de pagos tanto normales como anticipados
5.	Pantalla para realizar cancelación de facturas
6.	Pantalla para realizar ajustes a pagos no considerados
7.	Pantalla para realizar ajustes a pagos mal aplicados por cheques devueltos o por error de sistema

4.2.6 LISTA DE REPORTES

Contabilidad

No.	Nombre
1.	Reporte de subsubcuentas : RepSubsubcuenta.rpt
2.	Reportes de auxiliares por nivel de cuenta : RepAuxNivelCCL.rpt, RepAuxNivelCCO.rpt, RepAuxNivelCMA.rpt, RepAuxNivelSCU.rpt, RepAuxNivel.SSC.rpt
3.	Reporte diario general : RepDiarioGeneral.rpt
4.	Reporte de balanza de comprobación : RepBalanzaCompr.rpt
5.	Reporte de balanza de comprobación por cuenta de mayor : RepBalanzaCompM.rpt
6.	Reportes Auxiliares por cuenta : RepAuxCCL.rpt, RepAuxCCO.rpt, RepAuxCMA.rpt, RepAuxSCU.rpt, RepAux.SSC.rpt
7.	Reporte de cuentas con saldo : RepEdoCtasTodos.rpt
8.	Reporte de estados financieros : RepEdoFin.rpt
9.	Reporte de balance general : RepBalanceGeneral.rpt

Cuentas por cobrar

No.	Nombre
1.	Reporte de cifras de control : RepCifrasCtrl.rpt
2.	Reportes de estado de cuenta general : RepEdoCta.rpt
3.	Reporte de pronostico de cobranza : RepPosibleCobanza.rpt
4.	Reporte de pronostico de revisión : RepRevision.rpt
5.	Reporte de saldo vencido : RepSaldoVencido.rpt
6.	Reporte de antigüedad de saldos : RepAntiSaldo.rpt
7.	Reporte de pagos realizados : RepPagosRealizados.rpt
8.	Reporte de factura : RepCxCFacturas.rpt

4.2.7 CRITERIOS DE TERMINACIÓN, RECHAZO Y REANUDACIÓN DE LAS PRUEBAS

Terminación	Rechazo	Reanudación

4.2.8 MECANISMO DE SEGUIMIENTO DE ERRORES

Responsable	Función
Blanca Gil Castellanos (bcc)	Probar módulos y corregir posibles fallas.
Mario Carrasco Carrasco (mcc)	Probar módulos y corregir posibles fallas.
Edgar García Adame (ega)	Probar módulos y corregir posibles fallas.

4.2.9 CALENDARIO DE ACTIVIDADES DE PRUEBAS

Contabilidad

No.	Actividades	Inicio	Termino	Responsable /Participan	Metas
1.	Instalación del software	14-07-97	14-07-97	bcc,mcc,ega	Instalación correcta del software
2.	Actualizaciones a los diferentes catálogos: Clasificación de ctas Cuentas de mayor Centros de costos Subcuentas Subsubcuentas Areas de trabajo	14-07-97	15-07-97	bcc,mcc,ega	Actualización correcta de los datos en los diferentes catálogos. En todas las actualizaciones de los catálogos haya validaciones a nivel campo para que la información sea única e integra.
3.	Generación de varios movimientos contables a través de la facturación en el módulo de reservación y ctas por cobrar.	15-07-97	15-07-97		Haya suficientes movimientos contables para poder probar el sistema de integral de contabilidad
4.	Accesar al sistema (módulos de contabilidad y cuentas	16-07-97	16-07-97	bcc,mcc,ega	Funcionamiento correcto de los diferentes niveles

	por cobrar) con diferentes niveles de facultades a través de dar de alta a diferentes usuarios y grupos				de seguridad. Funcionamiento correcto de actualizaciones en los catálogos de usuarios y grupos.
5.	Generar varias pólizas modelo y probar su afectación en la contabilidad.	17-07-97	17-07-97	bcc,mcc,ega	Afectación correcta en la contabilidad general.
6.	Generar varias pólizas modelo cuenta y probar su afectación en la contabilidad.	17-07-97	17-07-97	bcc,mcc,ega	Afectación correcta en la contabilidad general.
7.	Realizar varios movimientos a través de pólizas manuales	18-07-97	18-07-97	bcc,mcc,ega	Afectación correcta en la contabilidad general.
8.	Realizar ajustes a pólizas	18-07-07	18-07-97	bcc,mcc,ega	Afectación correcta en la contabilidad general.
9.	Emisión de diferentes reportes : -Auxs por nivel de cta -Diario general -Balanza de comprobación -Balanza de comprobación por cuenta de mayor	18-07-97	21-07-97	bcc,mcc,ega	Funcionamiento correcto de los reportes. Toda la información emitida sea la correcta.

	-Auxiliares por cuenta -Cuentas con saldo				
10.	Captura y emisión de reportes de estados financieros.	21-07-97	21-07-97	bcc,mcc,ega	Funcionamiento correcto de la captura y de los reportes emitidos.
11.	Captura y emisión de reportes de balance general	21-07-97	21-07-97	bcc,mcc,ega	Funcionamiento correcto de la captura y de los reportes emitidos.
12.	Realizar cierre contable mensual	25-07-97	25-07-97	bcc,mcc,ega	Todos las cifras del cierre sean las correctas.

Cuentas por cobrar

No.	Actividades	Inicio	Termino	Responsable /Participan	Metas
1.	Actualizaciones a los diferentes catálogos: clientes bancos	22-07-97	22-07-97	bcc,mcc,ega	Actualización correcta de los datos en los diferentes catálogos. En todas las actualizaciones de los catálogos haya validaciones a nivel campo para que la información sea única e integra.
2.	Realizar pagos normales y anticipados	22-07-97	22-07-97	bcc,mcc,ega	Correcta afectación contable de saldos, así como correcta emisión de los recibos de pagos.
3.	Realizar ajustes de diferentes movimientos: Cancelación de facturas Pagos no considerados Cancelación de pagos	23-07-97	23-07-97	bcc,mcc,ega	Correcta afectación en la contabilidad general.
4.	Emisión de diferentes reportes : Cifras de control	24-07-97	24-07-97	bcc,mcc,ega	Funcionamiento correcto de los reportes. Toda la

Reportes				información emitida
Estado de cta general				sea la correcta.
Pronostico de cobranza				
Pronostico de revisión				
Saldo vencido				
Antigüedad de saldos				
Pagos realizados				
Facturas				

4.2.10 SCRIPTS DE PRUEBAS

Scripts de Pruebas Sistema Integral de Contabilidad

Parte del sistema a probar:

Módulo de contabilidad

Caso	Caso de prueba	Resultado esperado	OK
1.	Instalación del software	Funcionamiento correcto	ok
2.	Actualización al catálogo de clasificación de cuentas	Funcionamiento correcto	ok
3.	Actualización al catálogo de cuentas de mayor	Funcionamiento correcto	ok
4.	Actualización al catálogo de centros de costos	Funcionamiento correcto	Error OK
5.	Actualización al catálogo de subcuentas	Funcionamiento correcto	ok
6.	Actualización al catálogo de subsubcuentas	Funcionamiento correcto	ok
7.	Actualización al catálogo de áreas de trabajo	Funcionamiento correcto	ok
8.	Accesar al sistema (módulos de contabilidad y cuentas por cobrar) con diferentes niveles de facultades a través de dar de alta a diferentes usuarios y grupos	Funcionamiento correcto	Error ok

9.	Generar varias pólizas modelo y probar su afectación en la contabilidad.	Funcionamiento correcto	Error ok
10.	Generar varias pólizas modelo cuenta y probar su afectación en la contabilidad.	Funcionamiento correcto	ok
11.	Realizar varios movimientos a través de pólizas manuales	Funcionamiento correcto	ok
12.	Realizar ajustes a pólizas	Funcionamiento correcto	ok
13.	Reporte de Aux por nivel de cta	Funcionamiento correcto	ok
14.	Reporte diario general	Funcionamiento correcto	ok
15.	Reporte de Balanza de comprobación	Funcionamiento correcto	ok
16.	Reporte de Balanza de comprobación por cuenta de mayor	Funcionamiento correcto	Error ok
17.	Reporte de auxiliares por cuenta	Funcionamiento correcto	ok
18.	Reporte de cuentas con saldo	Funcionamiento correcto	ok
19.	Captura y emisión de reportes de estados financieros.	Funcionamiento correcto	ok
20.	Captura y emisión de reportes de balance general	Funcionamiento correcto	ok
21.	Realizar cierre contable mensual	Funcionamiento correcto	ok

Scripts de Pruebas Sistema Integral de Contabilidad

Parte del sistema a probar:

Módulo de cuentas por cobrar

Caso	Caso de prueba	Resultado esperado	OK
1.	Actualizaciones al catálogo de clientes	Funcionamiento correcto	ok
2.	Actualizaciones al catálogo de bancos	Funcionamiento correcto	ok
3.	Pagos normales y anticipados	Funcionamiento correcto	ok
4.	Cancelación de facturas	Funcionamiento correcto	ok
5.	Pagos no considerados	Funcionamiento correcto	ok
6.	Cancelación de pagos	Funcionamiento correcto	ok
7.	Reporte de cifras de control	Funcionamiento correcto	ok
8.	Reporte de pronóstico de cobranza	Funcionamiento correcto	ok
9.	Reporte de pronóstico de revisión	Funcionamiento correcto	ok
10.	Reporte de saldo vencido	Funcionamiento correcto	ok
11.	Reporte de antigüedad de saldos	Funcionamiento correcto	ok
12.	Reporte de pagos realizados	Funcionamiento correcto	ok
13.	Reporte de facturas	Funcionamiento correcto	ok

4.2.11 REPORTE DE ERRORES

Formato de Reporte de Errores

Nombre del sistema: Sistema Integral de Contabilidad

Versión: 1.0

Módulo de contabilidad

- Responsable del proyecto:

Blanca Gil Castellanos, Mario Carrasco Carrasco, Edgar García Adame

- Responsable de prueba:

Blanca Gil Castellanos, Mario Carrasco Carrasco, Edgar García Adame

Fecha: 14 de julio de 1997.

Ser	Ord	Descripción	Sev	Ord	Acción	Status
1.	1	Error de tiempo de corrida a la hora de actualizar datos en el catalogo de centros de costos	1	1	Revisar instrucción de actualización a la base de datos. Tal vez falte alguna llave	Corregido
2.	2	El acceso al sistema a través de varios usuarios dados de alta con diferentes facultades no es el correcto de acuerdo a lo que desea el usuario.	1	1	Revisar las facultades con el usuario, para que nos indique que funciones puede realizar cada usuario. Se realice un perfil de cada usuario.	Corregido

3.	1	Las pólizas modelo no funcionan adecuadamente, ya que están afectando la contabilidad de manera inversa, es decir realizar cargos en lugar de abonos y viceversa.	1	1	Revisar rutinas donde se realiza la afectación a cuentas. Seguramente solo hay que invertir la lógica.	Corregido
4.	1	El Reporte de balanza de comprobación por cuenta de mayor tiene muy encimada la información.	3	3	Existe mucho espacio para esparcir los datos sobre el reporte. Solo hay que mover los datos.	Corregido.
5.	1	A la hora de salirse de cualquier pantalla no se esta regresando a la pantalla principal. Se regresa a otra aplicación	3	3	Revisar si todas las pantallas están definidas como formas hijas.	Siguiente versión.
6.	1	Falta forma ancla. Esta forma es un estándar que se presenta a la hora de inicializar aplicación	3	3	Crear forma ancla y desplegarla mientras se esta cargando aplicación.	Siguiente versión.
7.						
8.						
9.						
10.						
11.						
12.						

Formato de Reporte de Errores

Nombre del sistema: Sistema Integral de Contabilidad

Versión: 1.0

Módulo de cuentas por cobrar

- Responsable del proyecto:

Blanca Gil Castellanos, Mario Carrasco Carrasco, Edgar García Adame

- Responsable de prueba:

Blanca Gil Castellanos, Mario Carrasco Carrasco, Edgar García Adame

Fecha: 14 de julio de 1997.

Sec.					Síntoma	Siatus
1.	1	Error de tiempo de corrida a la hora de actualizar datos en el catalogo de clientes	1	1	Revisar instrucción de actualización a la base de datos. Tal vez falte alguna llave	Corregido
2.	1	A la hora de realizar una cancelación de una factura no se están actualizando los saldos de las cuentas afectadas	1	1	Revisar instrucción de actualización a la base de datos. Tal vez falte alguna llave. Tal vez no se están realizando las operaciones de resta y sumas para cancelar dichos movimientos.	Corregido

3.	1	El Reporte de saldos vencidos tiene muy encimada la información.	3	3	Existe mucho espacio para esparcir los datos sobre el reporte. Solo hay que mover los datos.	Corregido.
4.	1	A la hora de salirse de cualquier pantalla no se esta regresando a la pantalla principal. Se regresa a otra aplicación	3	3	Revisar si todas las pantallas están definidas como formas hijas.	Siguiente versión.
5.	1	Falta forma ancla. Esta forma es un estándar que se presenta a la hora de inicializar aplicación	3	3	Crear forma ancla y desplegarla mientras se esta cargando aplicación.	Siguiente versión.
6.						
7.						
8.						
9.						

Tipo de casos

1	Casos del negocio
2	Casos Técnicos

Categoría del error

1	Equivocado
2	Omisión
3	Extra

Severidad del error

1	Alta
2	Media
3	Baja

4.2.12 DOCUMENTO DE ACEPTACIÓN DEL USUARIO

Proyecto: Sistema Integral de contabilidad para una casa postproductora de cine y vídeo.
Compañía de sistemas responsable:
Compañía usuaria responsable:

Lista de puntos a validar:

RESPONSABLE	RESULTADOS	
	CORRECTO	INCORRECTO
Satisfacción de los requerimientos	X	
Apego al análisis y diseño autorizado	X	
Validación de pantallas	X	
Validación de reportes	X	
Validación de calculos	X	
Validación de la integridad de los datos introducidos	X	
Validación del desempeño de la aplicación	X	
Validación de afectación a otros sistemas	X	

LOS QUE FIRMAN AL CALCE HACEN CONSTAR QUE SE REALIZO LA PRUEBA DE ACEPTACION CORRESPONDIENTE AL PROYECTO Y QUE SE AUTORIZA QUE SEA INSTALADO EN PRODUCCION.

Usuario responsable (Nombre y Firma):	Grupo de Desarrolladores Blanca Gil Castellanos Mario Carrasco Carrasco Edgar García Adame
---------------------------------------	---

CONCLUSIONES

CONCLUSIONES

Como se pudo observar, el seguimiento de una metodología de desarrollo, aunado al conocimiento de las reglas del negocio, hizo posible la realización de un sistema que cubriera todo el ciclo de vida. La existencia de diferentes metodologías, a lo largo de la historia de la computación, refleja el interés de la comunidad de ofrecer mecanismos, técnicas, reglas y herramientas que ayuden a hacer cada vez mejores sistemas. Estas metodologías, al igual que los sistemas, están en proceso constante de desarrollo y cada día ofrecen nuevas alternativas que facilitan este proceso.

El haber planteado el objetivo de aprovechar al máximo los recursos del cliente sin sacrificar la flexibilidad requerida del sistema, nos llevó a crear una solución a bajo costo y con las características visuales a las que está acostumbrado el usuario. Este aprovechamiento de la tecnología del cliente abarca tanto sus sistemas actuales como sus equipos de operación. La única adición fue la referente a la adición de los elementos de red (concentrador, tarjetas, software de red).

Una de las conclusiones básicas de este trabajo es haber logrado el objetivo planteado, entregando al usuario final una herramienta de trabajo que facilite, agilice y ayude a las tareas diarias inherentes a la Contabilidad y al cobro de las Cuentas vivas; de una manera adecuada, amigable, clara y fácil que permita el trabajo integral de las áreas involucradas sin problemas de duplicación de información, retrasos en el conocimiento de movimientos contables e integridad en los datos.

Como equipo de trabajo, en esta tesis pudimos aplicar los conceptos vistos en materias de esta carrera como lo son: Base de Datos, Programación Estructurada, Estructura de Datos, Administración de Centros de Cómputo e Ingeniería de Sistemas; basándonos en una metodología de desarrollo tal que permitió el trabajo en equipo, tanto por parte del usuario como de nosotros, responsables de entregar el sistema, en todas las etapas de análisis, diseño, construcción, pruebas y liberación.

Durante el desarrollo de esta tesis se pudo observar que el uso de nuevas tecnologías (lenguajes visuales, redes de computadoras más rápidas, bases de datos, sistemas cliente servidor, case, etc.) en los sistemas actuales hacen mas rápido su desarrollo, así como más confiable y sencilla en su operación. Además de dar mas flexibilidad para crecimientos futuros.

También se constato que al usar estándares (interface gráfica bajo Windows) se facilita el aprendizaje y operación de sistemas. Esto debido a que en la actualidad la mayoría de compañías cuentan con el sistema operativo Windows, y la mayoría de sus empleados saben como se maneja un procesador de palabras (word), una hoja de calculo (excel) bajo esta plataforma, los cuales siguen un mismo estándar.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Chen, M.; Norman, R.J. **Integrated computer-aided software engineering (CASE): adoption, implementation, and impacts Conf.** Title: Proceedings of the Twenty-Fifth Hawaii International Conference on System Science (Cat. No. 91TH0394-7) p. 362-373. Vol. 3.

White Paper on CASE Tools and Research. **A WHITE PAPER: CASE.**
<http://www.eecs.usma.edu/usma/academic/eecs/instruct/jansen/CASE.HTM>
1997. p. 1-5

Tesis: Sistema de Comercialización de Productos Automotrices en base a la Metodología CASE de Oracle.

Facultad de Ingeniería. UNAM. México 1995. p. 57-74.

Tesis: Diseño e implementación de un sistema de monitoreo de claves de acceso a Internet para la validación del propietario a través de la red UNAM.

Facultad de Ingeniería. UNAM. México 1996. p. 11-62.

Tesis: Implementación de un sistema de prestadores de bienes y servicios en Ixtapa-Zihuatanejo.

Facultad de Ingeniería. UNAM. México 1995. p. 164-183.

The CASE Tool Home Page.

Universidad de Sunderland. 1996. U.K.

<http://osiris.sunderland.ac.uk/sst/case2/welcome.html>

Software Engineering Questions and Answers.

Queens University. Canada. p. 1-7.

<http://www.qcis.queensu.ca/Software-Engineering/questions.html>

CASE*Method Task and Deliverables.

The Relational Database Management System.

Versión 1.0

Richard Barker, Dai Clegg, Jeremy Davis.

Oracle Corporation UK Limited, Chertsey, Surrey.

England - Dec. 1988.

CASE*Method Entity Relationship Modelling

The Relational Database Management System.

Versión 1.0

Richard Barker, Dai Clegg, Jeremy Davis.

Oracle Corporation UK Limited, Chertsey, Surrey.

England - Dec. 1988.

CASE*Method Function and Process Modelling

The Relational Database Management System.

Versión 1.0

Richard Barker, Cliff Longman, Barbara Barker

Oracle Corporation UK Limited, Chertsey, Surrey.

U.S.A. 1992.

SQL Spoken here.

Jonathan Sayles. Qued Information Massachusetts. 1989.

Elementos y herramientas en el desarrollo de sistemas de información: una visión actual de la tecnología CASE .

Coordinación Mario G. Plattini, Velthuis, Sunil N. Daryanani Daryanani.
Editorial Ra-Ma Madrid. 1995.

Software Engineering

Jones Gregory W. Editorial J.Wiley New York. 1990.

An Introduction to Database Systems.

C.J. Date. Addison Wesley. 6a. ed. 1995.

Microsoft Visual Basic Home Page.

<http://www.microsoft.com/vbasic/>

Microsoft Corporation

One Microsoft Way Redmond, WA 98052-6399 USA.

Eliás Lara Flores. **Primer Curso de Contabilidad.** Trillas. 9a. ed.

Juan Carlos Torres Tovar. **Contabilidad I.** Introducción a la Contabilidad. Ed. Diana. p.
15-169.

Fundamentos de Contabilidad.

David Himmelblau. Unión Tipográfica Editorial Hispano Americana. 1965.

Client/Server Architecture

Alex Berson

1994

Mc Graw-Hill

Essential Client/Server Survival Guide

Ortali/Dan Hankey, Jeri Edwards

1995

Van Nostrand Reinhold

TCP/IP Companion

Martin R. Arick

1992

Wiley-Quad

Tutorial "Fundamentos de TCP/IP"

Uniform 1993 (Foro Internacional sobre el sistema operativo UNIX)

Apuntes de diplomado en sistemas LANIA

(Laboratorio Nacional de Informática Avanzada)

1996

"Arquitectura cliente/servidor"

"Redes de computadoras"

Data Communications, Networks and Systems

Bartee

1990

SAMS

APÉNDICE A
SISTEMAS MANEJADORES DE
BASES DE DATOS PARA PC'S Y
ELECCIÓN DE ALGUNA

APÉNDICE A.

SISTEMAS MANEJADORES DE BASE DE DATOS PARA PC'S Y ELECCION DE ALGUNA (CARACTERISTICAS, VENTAJAS Y DESVENTAJAS)

El objetivo de un sistema de base de datos es simplificar y facilitar el acceso a los datos.

Las vistas de alto nivel ayudan a lograrlo. Si el tiempo de respuesta para una consulta es demasiado largo, el valor del sistema se reduce. El funcionamiento del sistema depende de la eficiencia de las estructuras de datos utilizados para representar los datos de la base de datos y de qué tan eficientemente pueda operar el sistema con esas estructuras. Como sucede en muchos otros aspectos de los sistemas de cómputo, deben hacerse concesiones, no sólo entre el espacio y el tiempo, sino también entre la eficiencia de un tipo de operación y la de otro.

La aparición de la computadora personal en el mercado, y su rápida evolución tecnológica, así como el abatimiento del costo, ha propiciado su uso en gran escala. En consecuencia ha surgido una gran cantidad de software de Base de Datos.

Por la facilidad de implantación del modelo relacional, el desarrollo de los manejadores de Base de Datos se ha orientado a éste. El poder de la facilidad de uso son los aspectos fundamentales para los usuarios de bases de datos relacionales.

El manejador de base de datos es un módulo de programas que constituye la interface entre los datos de bajo nivel almacenados en la base de datos y los programas de

aplicaciones y las consultas hechas al sistema. El manejador de base de datos es responsable de las siguientes tareas:

- **Interacción con el manejador de archivos.** Los datos sin procesar se almacenan en el disco mediante el sistema de archivos proporcionando normalmente por un sistema operativo convencional. El manejador de base de datos traduce las diferencias proposiciones en DML, a comandos del sistema de archivos de bajo nivel. Así, el manejador de base de datos se encarga realmente del almacenamiento, recuperación y actualización de los datos en la base de datos.
- **Implantación de la integridad.** Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de limitaciones de consistencia. El administrador de la base de datos debe especificar estas limitaciones en forma explícita. Si se especifican estas limitantes, entonces el manejador de la base de datos puede verificar si las actualizaciones a la base de datos resultan en la violación cualquiera de estas limitantes, y si es así, se podrá realizar la acción apropiada.
- **Puesta en práctica de la seguridad.** Es preciso que todos los usuarios de la base de datos tengan acceso a todo su contenido. Es labor del manejador de la base de datos hacer que se cumplan estos requisitos de seguridad.
- **Respaldo y recuperación.** Un sistema de cómputo, como cualquier otro dispositivo mecánico a eléctrico, esta sujeto a fallas. Es responsabilidad del manejador de la base de datos detectar estas fallas y restaurar la base de datos al estado que existía antes de presentarse la falla.
- **Control de concurrencia.** Cuando varios usuarios actualizan la base de datos en forma concurrente, es posible que no se conserve la consistencia se los datos. Es necesario que el sistema controle la interacción entre los usuarios concurrentes; lograr dicho control es una de las tareas del manejador de la base de datos.

Los sistemas manejadores de base de datos relacionales (RDBMS) han sido altamente aceptados por la forma en que se manejan los datos, los sistemas relacionales ofrecen los siguientes beneficios:

- Permiten un sencillo acceso a los datos.
- Flexibilidad en el modelado de los datos.
- Disminuye la redundancia de datos y el nivel de almacenaje.
- Existe independencia entre el almacenaje físico y el diseño lógico de datos.
- Nivel alto en la manipulación de datos.

CLIPPER

Esta paquete para el desarrollo de base de datos es, definitivamente para los programadores. Si bien carece de algunas de las excelentes funciones de generación de código que tienen los demás paquetes. Clipper ofrece una riqueza de armas y capacidad que los programadores necesitan.

Dos de sus armas son un generador de reportes (RL) y función (DBU) para la creación y manejo de los archivos de la base de datos, escritos en el propio lenguaje de programación Clipper, y se incluye el código fuente, el cual se utilizar como referencia, o modificarlo para añadirlo a las aplicaciones de la base de datos.

La pantalla de DBU enlista las opciones a través de su parte superior, junto con las teclas de funciones asignadas. El resto de la pantalla está dedicado a una representación visual del panorama de la base de datos activa, la cual consiste en una columna partida en tres secciones. El primer grupo exhibe el nombre de la base de datos activa. Si al invocar el programa DBU se invoca en argumento, aparecerá la base de datos o la vista específica. En la siguiente sección aparecerán los índices activos

relacionados con la base de datos activa, y el grupo al fondo muestra los nombres de los campos para la base de datos activa.

El generador de reportes, RL, no es tan flexible. No soporta un bosquejo WYSIWYG del reporte y tampoco permite ver la salida en forma preliminar conforme se vaya trabajando. La verdadera potencia de Clipper se basa en su codificación.

El compilador del programa es muy rápido además de tener algunas funciones muy interesantes, dentro de estas se destacan las siguientes: soporta los llamados bloques de código, pedacitos de código ejecutable que se pueden almacenar como variables, o pasar a otros programas como argumentos, para ejecutar un bloque de código, se utiliza una función EVAL(); otra función útil es el uso hecho por Clipper de los archivos cerrados para el seguimiento, a fin de determinar cuales archivos en un programa depende de otros para operar correctamente y quedar al corriente.

Estando instalado este sistema, se puede invocar la función MiMake para llevar a cabo aquellas operaciones de compilación y enlace que se necesiten y mantener todos los archivos sincronizados. Clipper también soporta las funciones para leer los archivos binarios de DOS y escribir en ellos. También se aprecia el depurador de Clipper, el cual permite analizar el funcionamiento de código del programa, ejecutar comandos y revisar el estado de algunas variables, nombre de campo o expresiones en particular. Una ventanilla de estado enlista las bases de datos abiertas en todas las áreas del trabajo activas, así como los valores de todos los comandos Set.

FOXPRO

La interface de foxPro incluye menús presentados junto con una ventanilla de comandos conveniente, para utilizar los menús o escribir los comandos sin que ninguno de los elementos de la interface interfieran con otro. Su soporte al ratón es el mejor de todos los paquetes basados en caracteres. Así mismo, la ventana de comandos

mantiene una historia corrida de las instrucciones (al igual que Dbase III plus), esto facilita la repetición de los comandos utilizados a través de una sesión. También se puede seleccionar parte de la historia de los comandos, y anexarla a sus aplicaciones.

La ventana presenta gráficamente todas las áreas de trabajo disponible; se puede seleccionar un área de trabajo disponible, y abrir en ellas una base de datos, accionando un botón de comando. FoxPro ofrece una caja de diálogo con una lista de los campos principales.

El constructor de pantalla comienza como pantalla en blanco, en la cual se puede capturar el texto y colocar los campos, también se puede crear botones de comando, casillas, marcar con "palomas", botones de "radio" y lista de extraer. Se pueden agregar pedacitos de código a cualquier objeto, incluso a los campos.

Desde la pantalla de establecimiento, se puede ejecutar código antes y después del programa de la pantalla de captura, cuando se genere el código para la pantalla diseñada, se puede anexar otras pantallas a ellas, lo cual ahorra tiempo una vez formulada una biblioteca de pantalla genéricas.

El funcionamiento de FoxPro es magnífico gracias, en parte a su tecnología exclusiva de Rushmore, siendo su único inconveniente su voluminosa documentación, ya que es un problema localizar la información rápidamente, lo cual no obstante disminuye gracias a su fuerte función de ayuda en línea.

INFORMIX-SQL

Informix SQL es cien por ciento una base de datos con lenguaje de consulta estructurado (SQL), el aspecto del programa es muy austero: no se encontrará con la pantalla de colores múltiples con menús y ventanillas a la vista. Las pantallas del paquete, estilo Lotus, impulsadas por menús, autorizan las operaciones más

significativas de la base de datos, tales como la creación de tablas, la definición y modificaciones de los campos, etc. Informix también incluye un generador de reportes, y un sistema para la ejecución de archivos de la definición de formas diseñadas.

Dicho archivo comprende cinco partes: una sección de tablas serán accedidas por la forma; una tabla de atributos, la cual describe cada campo exhibido por la forma, y una sección opcional de instrucciones, que define las operaciones que habrán de ser llevadas a cabo sobre los campos dentro de la forma.

Las secciones y tablas informan al sistema que debe presentar; la sección de pantalla indica en donde presentarlo, la sección de atribuciones indica como presentarlo y la sección de instrucciones le dice al sistema que hace antes, mientras y después de presentarlo.

Se incluye con Informix-SQL varios paquetes de servicios: BECHECK verifica la integridad de los índices, si encuentra una discrepancia entre un archivo de datos y unos de sus índices, le permite reformar el índice; DRLINK y DBLOAD son de utilidad para el traslado de los datos entre Informix y el mundo exterior de los archivos de Lotus 1-2-3 Dbase, o ASCII; con DBSCHEMA, se pueden elaborar las instrucciones de SQL requeridas para crear una tabla o una base de datos.

PARADOX

Una de las características más interesantes de PARADOX es su velocidad, destacándose su rapidísimo tiempo de respuesta para la lectura y edición de las tablas.

Paradox soporta los formatos de importación y exportación más importantes, pero existen problemas en la importación de archivos almacenadas en el formato ASCII. Con Paradox, es necesario planear con cuidado antes de precipitarse a las construcción de una forma, aplicando esto principalmente con las formas que accesan

a múltiples tablas. El diseño de formas con tablas múltiples exige la creación de una forma maestra, después, existe también la introducción, en dicha forma maestra de las formas incluidas en las otras tablas. Así que para crear la forma maestra completa, debe diseñarse las formas que se incluirán primero.

El lenguaje para el manejo de la base de datos PARADOX es PAL, aunque es erróneo representar a PAL como simplemente un DML. Lo que dificulta la programación con PAL es que maneja la transmisión de comandos como si el programador fuera un robot sentado en un teclado operando PARADOX; sin embargo, si simplemente no se quiere tener nada que ver con PAL, podrá instalarse el programador personal. Este programa es, esencialmente un constructor de aplicaciones que a través de la creación de una aplicación completa con menús y formas, ofreciendo el código PAL como su producto final.

ACCESS

Access ha llegado a ser uno de los más populares manejadores de bases de datos relacional lanzados en el medio informático, debido a todas las posibilidades que integra. Es el producto más escalable en su clase, implanta un impresionante grupo de características nuevas y mejoras a la facilidad de uso, incluyendo varios Wizards nuevos.

Tiene el mejor rendimiento de Búsqueda, impone reglas de integridad de datos a nivel de base de datos que le permite determinar como los registros en tablas asociadas se ven afectados cuando se modifica una matriz. Reduciendo así la cantidad de código necesario para construir aplicaciones.

Cuando se crea una base de datos en Access todos sus componentes se guardan en un archivo único .MDB. Pulsado sobre el botón New se comienza el proceso de crear un objeto del tipo seleccionado y es ahí donde los Wizard entran en acción, facilitando

la creación de diseños funcionales. De los asistentes más potentes está el Table Wizard, que provee más de cuarenta modelos de tablas personales y de negocios, cada una con sus propios juegos de campos con formato para usar su propia base de datos.

Los Wizard de botones de comandos frecuentes determinan de antemano las respuestas de los eventos del ratón. Contiene características que permiten generar rápidamente reportes y formularios. Posee apoyo para OLE 2.0, que se puede incluir en formulario y reportes, así como para BOLB's.

Los Wizard son módulos orientados a ciertas tareas, que hacen una serie de preguntas y luego genera los objetos apropiados, según las respuestas. También posee un lenguaje de programación ObjetPAL.

Tiene una ventana Gráfica de Relationship que muestra todas las tablas en la base de datos con líneas entre ellas para representar las relaciones activas. Simplemente se arrastra de un campo en una tabla a otro en otra tabla para definir una relación y luego se personaliza el enlace.

Puede usar datos de varias bases de datos como: dBase III y IV, Foxpro 2.0 y 2.5, Paradox 3.x y 4.x así como Microsoft, Oracle, Sybase y el lenguaje SQL.

Access es una aplicación que se desarrollo bajo Windows, de esta forma facilita construir tareas rápidas por medio de ventanas y sus elementos.

Access nos ayuda a gestionar nuestra base de datos proporcionando una infraestructura eficiente para almacenar y recuperar la información, además genera automáticamente un formulario para complementar todos nuestros datos. Después de que hayamos introducido algunos datos, podemos pedir a Access que visualice una parte seleccionada de información utilizando unos procedimientos: búsqueda,

ordenación y consulta. También nos ayuda a imprimir solamente la parte de la información que deseamos ver en un informe. En cualquier momento, podemos personalizar la apariencia o editar el contenido de cualquier parte de nuestra base de datos.

CARACTERISTICAS DE ACCESS

- Cambiar el nombre de la columna para que se ajuste mejor.
- Cortar el texto fácilmente utilizando un menú abreviado.
- Crear automáticamente una nueva tabla con el asistente para tablas.
- Filtrar una tabla para presentar solamente los registros que deseamos ver.
- Exportar solo de una tabla a una hoja de cálculo de Excel.
- Esquematizar el diseño de nuestra base de datos o ver todas las relaciones a la vez utilizando las ventanas de relaciones.
- Establecer la propiedad de un campo fácilmente utilizando el botón general que nos ayuda a seleccionar las opciones.
- Ajustar la posición de los controles de un formulario desplazándolo un solo punto en la cuadrícula en un momento.
- Pero su mayor desventaja es que no se pueden crear archivos ejecutables (hasta el momento).

ELECCION DE UN MANEJADOR DE BASE DE DATOS PARA PC's

Uno de los aspectos importantes de los manejadores de base de datos más recientes es que entre la característica de poder que ofrecen la mayoría de los fabricantes se cuenta la posibilidad integrada de enlace en redes, la posibilidad de colocar un programa nuevo de Software en la red, para ofrecer acceso compartido a datos y lenguajes de programación que se puedan utilizar para desarrollar aplicaciones adecuadas a las necesidades de cada uno de los usuarios, incluyendo menús y reportes y que ese poder sea más accesible con una serie de menús, editores de toda

la pantalla y características automáticas que vuelvan relativamente sencilla la relación de operaciones relacionales. Productos pioneros como Dbase por ejemplo, tenía que aprender un gran número de comandos y procedimientos. Esto ya no sucede con los productos nuevos, tal es el caso de Access. La demanda del usuario de acceso a la base de datos ha apresurado a los productores de software a diseñar interfaces realista. Recientes adelantos técnicos (implantación de gráficos en la Base de Datos), como se mencionó anteriormente con Access se puede trabajar con la implementación de los avances OLE(Object link Embededd) en la versión 2.0. incorpora datos de Word 6.0, Excel 5.0, Power Point 3.0 y Microsoft Mail. Access es una aplicación que se desarrollo bajo Windows, de esta forma facilita construir tareas rápidas por medio de ventanas y sus elementos.

Para poder elegir la base de datos que se acerca a nuestras necesidades, fue necesario realizar un criterio de selección el cual se muestra en las figuras de la A.1 a la A.12.

PRODUCTO	COMPANIA
Clipper	Nantucket
FoxPro	Fox Software, Inc
Informix-SQL	Informix
Paradox	Borland International
Access	Microsoft

Fig. A.1. Tabla de productos las compañías que los producen.

Para fines prácticos. Se abreviarán con las siguientes letras a los productos

Clipper = C

FoxPro = F

Informix-SQL = I

Paradox = P

Access = A

Tiempo de procesamiento en Batch	C	F	I	P	A
bueno	X				
regular		X		X	X
malo			X		

Fig. A.2. Evaluación de Tiempo de Procesamiento en Batch

Tiempo importación y generación de índices	C	F	I	P	A
bueno		X		X	
regular	X				X
malo			X		

Fig. A.3. Evaluación de Tiempo Importación y generación de índices

Soporte de Núm Máximo de archivos abiertos	C	F	I	P	A
bueno			X		
regular	X	X		X	X
malo					

Fig. A.4. Evaluación de Soporte de Núm Máximo de archivos abiertos

Facilidad de Uso	C	F	I	P	A
bueno		X		X	X
regular	X		X		
malo					

Fig. A.5. Evaluación de Facilidad de Uso

Proporciona Seguridad	C	F	I	P	A
bueno		X	X	X	X
regular	X				
malo					

Fig. A.6. Evaluación que Proporciona Seguridad

Integridad en los Datos	C	F	I	P	A
bueno	X			X	X
regular		X	X		
malo					

Fig. A.7. Evaluación de la integridad de los Datos

Presentación de Calidad en los Reportes y Formas	C	F	I	P	A
bueno					X
regular	X	X	X	X	
malo					

Fig. A.8. Evaluación de la Presentación de Calidad en los Reportes y Formas.

Facilidad de Programación	C	F	I	P	A
bueno					X
regular	X	X		X	
malo			X		

Fig. A.9. Evaluación en la Facilidad de Programación

Compartición de los datos con otras bases de datos	C	F	I	P	A
bueno					X
regular	X	X	X	X	
malo					

Fig. A.10. Evaluación en la Compartición de los datos con otras bases de datos

Diversidad en Tipos de Datos	C	F	I	P	A
bueno					X
regular	X	X		X	
malo			X		

Fig. A.11. Evaluación en la Diversidad en Tipos de Datos

Manejo de Gráficos para formulario	C	F	I	P	A
bueno					
regular					X
malo	X	X	X	X	

Fig. A.12 Evaluación en el Manejo de Gráficos para formulario

Para establecer la puntuación que obtuvo cada producto se determinó un valor para cada categoría determinada de la siguiente manera:

Bueno = 2 puntos

Regular = 1 punto

Malo = 0 puntos

Obteniendo finalmente las siguientes tablas de puntuaciones:

CARACTERÍSTICAS	C	F	I	P	A
Tiempo de procesamiento en Batch	2	1	0	1	1
Tiempo de importación y generación de índices	1	2	0	2	1
Soporte de Núm Máximo de archivos abiertos	1	1	2	1	1
Facilidad de Uso	1	2	1	2	2
Proporciona Seguridad	1	2	2	2	2
Integridad en los datos	2	1	1	2	2
Presentación de Calidad en los Reportes y Formas	1	1	1	1	2
Facilidad de programación	1	1	0	1	2
Compartición de Datos con otras bases de datos	1	1	1	1	2
Diversidad de tipos de datos	1	1	0	1	2
Manejo de Gráficos para Formularios	0	0	0	0	1
TOTAL	12	13	8	14	18

Fig. A.13 Evaluación para la Puntuación final.

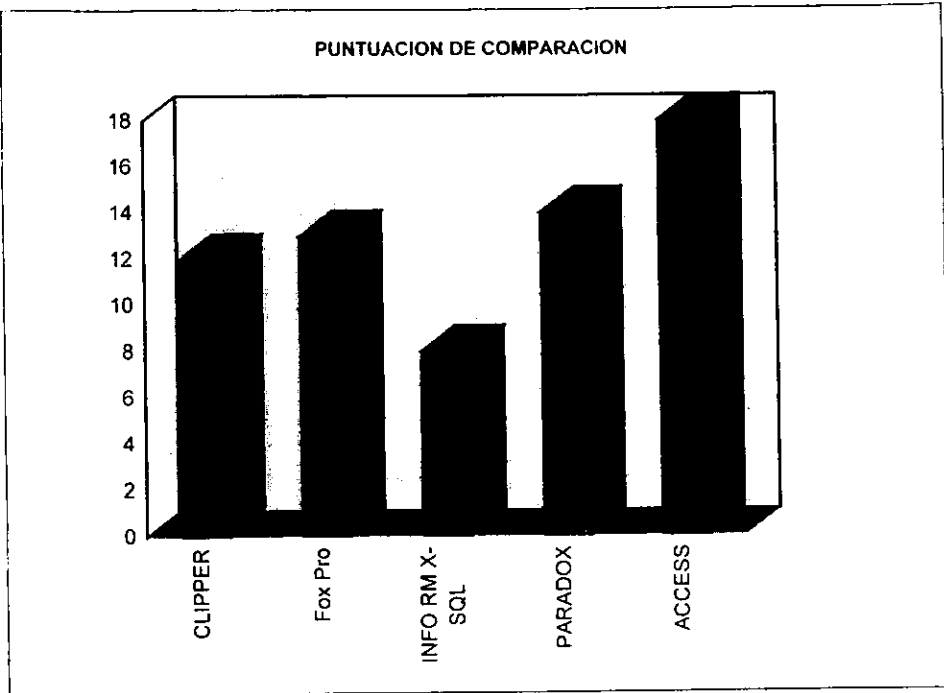


Fig. A.13. Gráfica comparativa de puntuación.

La elección de la base de datos se basó, además de todo lo anterior, en la disponibilidad comercial. Por lo antes mencionado y en relación a las capacidades de manejo, programación y de conocimiento, entre otras, asociadas, se determinó el utilizar Microsoft Access para el desarrollo del sistema.

**APÉNDICE B
CÓDIGO FUENTE DE ALGUNAS
PANTALLAS**

APÉNDICE B

CODIGO FUENTE DE ALGUNAS PANTALLAS

MODULO DE CONTABILIDAD

Pantalla principal

Es la primera pantalla que se despliega cuando es ejecutado el modulo de contabilidad. Esta pantalla pedirá un usuario y una contraseña de usuario para poder acceder el sistema. Una vez validado el usuario se desplegara un menu del cual el usuario podrá escoger una opción para trabajar. Las opciones a las que podrá acceder el operador dependerá del nivel con el que fue dado de alta.

```
Private Sub DefineLogo(LogoPrin)
    PrinLogo.Picture = LoadPicture(LogoPrin)
    TextLogo.Picture = LoadPicture(LogoText)
End Sub

Private Sub CmdAceptar_Click()
    If TxtUsuario = "" Or TxtPass = "" Then
        MsgBox "Teclee los datos del usuario.", 64
        Exit Sub
    End If
    TxtUsuario = UCase(TxtUsuario)
    TxtPass = UCase(TxtPass)
    If TxtUsuario = "ACRATA" Then
        If TxtPass <> PassW Then
            MsgBox "Usuario no reconocido "
            Exit Sub
        Else
            TxtTipoGpo = 1
            TxtArea = 0
        End If
    Else
        strSql = ""
        strSql = strSql + " Select *"
        strSql = strSql + " from usuario, "
        strSql = strSql + " grupo_usuario "
        strSql = strSql + " where usu_cve = " + TxtUsuario + ""
        strSql = strSql + " and usu_password = " + TxtPass + ""
        strSql = strSql + " and usu_gru_cve = gru_cve"
        Set Rds = Db.OpenRecordset(strSql, dbOpenSnapshot)
        If Rds.EOF Then
            MsgBox "Usuario no reconocido."
            Rds.Close
            Exit Sub
        Else
            TxtTipoGpo = Rds("gru_tipo_grupo")
            TxtArea = Rds("usu_are_cve")
        End If
        Rds.Close
    End If
End Sub
```

```
MenuPrin(1).Visible = True
If TxtTipoGpo = 1 Then 'superusuario
    MenuPrin(0).Visible = True
    MenuPrin(2).Visible = True
    MenuPrin(3).Visible = True
    MenuPrin(4).Visible = True
    MenuPrin(5).Visible = False
    MenuPrin(6).Visible = True
Elsel If TxtTipoGpo = 2 Then 'usuario contab
    MenuPrin(0).Visible = True
    MenuPrin(2).Visible = True
    MenuPoliza(0).Visible = True
    MenuCat(5).Visible = False
    MenuCat(8).Visible = False
    MenuPol(2).Visible = False
Elsel If TxtTipoGpo = 3 Then 'usuario general
    MenuPol(0).Visible = False
    MenuPol(2).Visible = False
End If
Pnl(2).Visible = False
End Sub
```

```
Private Sub CmdSalir_Click()
    Unload Me
End Sub
```

```
Private Sub Form_Load()
    Me.WindowState = 2
    If Not CargaParametro() Then
        MsgBox "No fué posible cargar parámetros iniciales " + TitWindow, 64
    Exit Sub
End If
```

```
DefineLogo (TextoPrin)
DefineLogo (LogoPrin)
```

```
If Not AbreBase() Then
    MsgBox "Error en apertura de base de datos " + TitWindow, 64
Exit Sub
End If
```

```
MenuPrin(0).Visible = False
MenuPrin(1).Visible = False
MenuPrin(2).Visible = False
MenuPrin(3).Visible = False
MenuPrin(4).Visible = False
MenuPrin(5).Visible = False
MenuPrin(6).Visible = False
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    x = CierraBase()
End Sub
```

```
Private Sub MenuBG_Click(Index As Integer)
    Select Case Index
        Case Is = 0
            ShowModalForm FRMBalanceGeneral
        Case Is = 1
            ShowModalForm FRMReporteBalanceGeneral
    End Select
End Sub
```

```
Private Sub MenuCat_Click(Index As Integer)
    Select Case Index
        Case Is = 0
            ShowModalForm FRMCatCuentaClasificacion
        Case Is = 1
            ShowModalForm FRMCatCuentaMayor
        Case Is = 2
            ShowModalForm FRMCatCentroCostos
        Case Is = 3
            ShowModalForm FRMCatSubCuenta
        Case Is = 4
            ShowModalForm FRMCatSubSuCuenta
        Case Is = 5
            ShowModalForm FRMCatArea
    End Select
End Sub
```

Pantalla de cuentas de mayor

Esta pantalla se usará para realizar actualizaciones (altas, bajas, modificaciones) al catálogo de cuentas de mayor a través de diversos botones. En un grid (cuadrícula o matriz) se presentarán todos los registros que contiene actualmente el catálogo.

```
Private Function VerificaCierreCuenta(ClaCCL, ClaCMA) As Integer
    VerificaCierreCuenta = False
```

```
    StrSql = ""
    StrSql = StrSql + " Select * from Cierre_Cuenta "
    StrSql = StrSql + " Where CIC_SSC_SCU_CCO_CMA_CCL_Cve=" + ClaCCL
    StrSql = StrSql + " And CIC_SSC_SCU_CCO_CMA_Cve=" + ClaCMA
    Set RdsAux = Db.OpenRecordset(StrSql, dbOpenSnapshot)
    If RdsAux.RecordCount <> 0 Then
        VerificaCierreCuenta = True
    End If
    RdsAux.Close
```

```
End Function
```

```
Private Function VerificaHijos(ClaCCL, ClaCMA) As Integer
```

```
    VerificaHijos = False
    StrSql = ""
    StrSql = StrSql + " Select CCO_CMA_CCL_Cve,CCO_CMA_Cve from Centro_De_Costos"
    StrSql = StrSql + " Where CCO_CMA_CCL_Cve=" + ClaCCL
    StrSql = StrSql + " And CCO_CMA_Cve=" + ClaCMA
    Set RdsAux = Db.OpenRecordset(StrSql, dbOpenSnapshot)
    If RdsAux.RecordCount <> 0 Then
        VerificaHijos = True
    End If
    RdsAux.Close
```

```
End Function
```

```
Private Function Boton(Encender As Boolean)
```

```
    If Encender = True Then
        CmdBaja.Enabled = True
        CmdModificar.Enabled = True
        CmdAlta.Enabled = True
        CmdSalir.Enabled = True
    Else
        CmdBaja.Enabled = False
        CmdModificar.Enabled = False
        CmdAlta.Enabled = False
        CmdSalir.Enabled = False
    End If
```

```
End Function
```

```
Private Function ActualizaDatos() As Integer
```

```
    ActualizaDatos = False
    On Error GoTo ManejaError

    BeginTrans
    StrSql = "Update Cuenta_Mayor Set "
    If OpbCuentaTotalizadora.Value = True Then
```

```

StrSql = StrSql + " CMA_Tipo_Cuenta =1"
Elseif OpbCuentaDetalle.Value = True Then
    StrSql = StrSql + " CMA_Tipo_Cuenta=2"
End If
StrSql = StrSql + ",CMA_Descripcion = " + TxtDescripcion + ""
StrSql = StrSql + " Where CMA_CCL_Cve = " + CmbCuentaClasificacion.BoundsText
StrSql = StrSql + " AND CMA_Cve = " + TxtCuentaMayorClave.Text
Db.Execute StrSql, dbFailOnError

StrSql = "Update subsubcuenta Set "
If OpbCuentaTotalizadora.Value = True Then
    StrSql = StrSql + " SSC_Tipo_Cuenta =1"
Elseif OpbCuentaDetalle.Value = True Then
    StrSql = StrSql + " SSC_Tipo_Cuenta=2"
End If
StrSql = StrSql + ",SSC_Descripcion = " + TxtDescripcion + ""
StrSql = StrSql + " Where SSC_SCU_CCO_CMA_CCL_Cve = " + CmbCuentaClasificacion.BoundsText
StrSql = StrSql + " and SSC_SCU_CCO_CMA_Cve = " + TxtCuentaMayorClave.Text
StrSql = StrSql + " and SSC_SCU_CCO_Cve = 0"
StrSql = StrSql + " and SSC_SCU_Cve = 0"
StrSql = StrSql + " and SSC_Cve = 0"
Db.Execute StrSql, dbFailOnError

```

```

CommitTrans
ActualizaDatos = True

```

```

Exit Function
ManejaError:
    Errores (Err)
    Rollback
End Function

```

```

Private Function VerificaPartida(ClaCCL, ClaCMA) As Integer
    VerificaPartida = False

```

```

StrSql = ""
StrSql = StrSql + " Select * from Partida "
StrSql = StrSql + " Where PAR_SSC_SCU_CCO_CMA_CCL_Cve=" + ClaCCL
StrSql = StrSql + " And PAR_SSC_SCU_CCO_CMA_Cve=" + ClaCMA
Set RdsAux = Db.OpenRecordset(StrSql, dbOpenSnapshot)

```

```

If RdsAux.RecordCount <> 0 Then
    VerificaPartida = True
End If
RdsAux.Close

```

```

End Function

```

```

Private Function VerificaPolizaModeloCuenta(ClaCCL, ClaCMA) As Integer
    VerificaPolizaModeloCuenta = False

```

```

StrSql = ""
StrSql = StrSql + " Select * from Poliza_Modelo_Cuenta "
StrSql = StrSql + " Where PMC_SSC_SCU_CCO_CMA_CCL_Cve=" + ClaCCL
StrSql = StrSql + " And PMC_SSC_SCU_CCO_CMA_Cve=" + ClaCMA
Set RdsAux = Db.OpenRecordset(StrSql, dbOpenSnapshot)
If RdsAux.RecordCount <> 0 Then

```

```
    VerificaPolizaModeloCuenta = True
End If
RdsAux.Close
End Function
```

```
Private Sub CmbCuentaClasificacion_Click(Area As Integer)
If Area = 2 Then
    TxtCuentaDescripcion.Text = ""
    TxtDescripcion.Text = ""
    TxtCuentaMayorClave.Mask = ""
    TxtCuentaMayorClave.Text = ""
    TxtCuentaMayorClave.Mask = "##"
    TxtCuentaCuentaClasificacion.Text = CmbCuentaClasificacion.BoundText
End If
End Sub
```

```
Private Sub CmdAlta_Click()
    Dim x

    Boton (False)
    PnlActualiza.Visible = True
    PnlModifica.Visible = False
    PnlAlta.Visible = True
    PnlActTitulo.Caption = "Alta de Cuenta Mayor"

    If Not LimpiaCampos() Then
        Exit Sub
    End If
End Sub
```

```
Private Sub CmdAltAlta_Click()

    If Not VerificaDatos() Then
        Exit Sub
    End If

    If Not InsertaDatos() Then
        If Not LimpiaCampos Then
            Exit Sub
        End If
        MsgBox "Error al insertar La Cuenta Mayor", 64
        Exit Sub
    End If

    Beep
    MsgBox "La Cuenta Mayor ha sido dado de alta", 64

    If Not LimpiaCampos() Then
        Exit Sub
    End If

    CdGrid.Refresh
    GrdCat.Refresh
End Sub
```

```
Private Sub CmdAltSalir_Click()
    PnlActualiza.Visible = False
    Boton (True)
End Sub
```

```
Private Sub CmdBaja_Click()
    If GrdCat.SelBookmarks.Count > 1 Then
        Beep
        MsgBox "Debe seleccionar solo un registro a dar de baja", 64
        Exit Sub
    ElseIf GrdCat.SelBookmarks.Count = 0 Then
        Beep
        MsgBox "Debe seleccionar al menos un registro a dar de baja", 64
        Exit Sub
    End If
    If MsgBox("Esta seguro de eliminar la Cuenta Mayor seleccionada", vbOKCancel) = vbOK Then
        If Not BorraDatos(GrdCat.Columns("CMA_CCL_Cve").Value, GrdCat.Columns("CMA_Cve").Value)
        Then
            MsgBox "ERROR] No pudo realizarse proceso de borrado"
        End If
    End If
End Sub
```

```
Private Sub CmdModCancelar_Click()
    CmbCuentaClasificacion.Enabled = True
    TxtCuentaMayorClave.Enabled = True
    FraCuentas.Enabled = True
    PnlActualiza.Visible = False
    Boton (True)
End Sub
```

```
Private Sub CmdModificar_Click()
    Dim x

    If GrdCat.SelBookmarks.Count > 1 Then
        Beep
        MsgBox "Debe seleccionar solo un registro a modificar", 64
        Exit Sub
    ElseIf GrdCat.SelBookmarks.Count = 0 Then
        Beep
        MsgBox "Debe seleccionar al menos un registro a modificar", 64
        Exit Sub
    End If

    Boton (False)
    PnlActualiza.Visible = True
    PnlModifica.Visible = True
    PnlAlta.Visible = False
    PnlActTitulo.Caption = "Modificación de Cuenta Mayor"

    If Not LimpiaCampos() Then
        Exit Sub
    End If

    If Not CargaDatos(GrdCat.Columns("CMA_CCL_Cve").Value, GrdCat.Columns("CMA_Cve").Value)
    Then
```

```
Exit Sub
End If

CmbCuentaClasificacion.Enabled = False
TxtCuentaMayorClave.Enabled = False

If OpbCuentaTotalizadora.Value = True Then
    If VerificaHijos(CmbCuentaClasificacion.BoundText, TxtCuentaMayorClave.Text) Or
    VerificaCierreCuenta(CmbCuentaClasificacion.BoundText, TxtCuentaMayorClave.Text) Then
        FraCuentas.Enabled = False
    End If
Elseif OpbCuentaDetalle.Value = True Then
    If VerificaPartida(CmbCuentaClasificacion.BoundText, TxtCuentaMayorClave.Text) Or
    VerificaPolizaModeloCuenta(CmbCuentaClasificacion.BoundText, TxtCuentaMayorClave.Text) Then
        FraCuentas.Enabled = False
    End If
End If
End Sub
```

```
Private Sub CmdModModificar_Click()
    If Not VerificaDatos() Then
        Exit Sub
    End If
    If Not ActualizaDatos() Then
        Exit Sub
    End If
    GrdCat.Refresh
    CdGrid.Refresh
    CmbCuentaClasificacion.Enabled = True
    TxtCuentaMayorClave.Enabled = True
    FraCuentas.Enabled = True
    PnlActualiza.Visible = False
    Boton (True)
End Sub
```

```
Private Sub CmdSalir_Click()
    Unload Me
End Sub
```

```
Private Sub Form_Load()
    'Se define le origen de los datos del grid
    Me.WindowState = 2
    PnlActualiza.Visible = False
    If Not CargaGrid() Then
        Exit Sub
    End If
    strSql = ""
    strSql = strSql + "Select CCL_Cve , "
    strSql = strSql + "    (Str$(CCL_Cve) + '.' + "
    strSql = strSql + "    CCL_Descripcion) as Cadena "
    strSql = strSql + "from Cuenta_Clasificacion "
    strSql = strSql + " Where CCL_Tipo_Cuenta=1"
    strSql = strSql + " order by CCL_Cve"
    If Not DefineDataControl(CdCuentaClasificacion, DbName, DbDir, strSql, vbRSTypeSnapshot) Then
        Exit Sub
    End If
```



```
CdCuentaClasificacion.Refresh
End Sub
```

```
Private Function CargaGrid() As Integer
    CargaGrid = False
```

```
    StrSql = "CMA"
```

```
    If Not DefineDataControl(CdGrid, DbName, DbDir, StrSql, vbRSTypeDynaset) Then
        Exit Function
    End If
```

```
    CargaGrid = True
End Function
```

```
Private Function VerificaDatos() As Integer
    VerificaDatos = False
```

```
    If IsNumeric(CmbCuentaClasificacion) Then
        MsgBox "Debe Elegir del combo"
        CmbCuentaClasificacion.SetFocus
        Exit Function
    End If
```

```
    If IsNumeric(CmbCuentaClasificacion) Then
        MsgBox "Debe Elegir del combo"
        CmbCuentaClasificacion.SetFocus
        Exit Function
    End If
```

```
    If CmbCuentaClasificacion.BoundText = "" Then
        MsgBox "La Cuenta Clasificacion no puede ser nula", 64
        CmbCuentaClasificacion.SetFocus
        Exit Function
    End If
```

```
    If CmbCuentaClasificacion.BoundText <> "" Then
        If Not VerificaCombos(CdCuentaClasificacion, CmbCuentaClasificacion.BoundText) Then
            MsgBox "Debe Seleccionar de la lista", 64
            CmbCuentaClasificacion.SetFocus
            Exit Function
        End If
    End If
```

```
    If TxtCuentaMayorClave = " " Then
        MsgBox "La Clave no puede ser nula", 64
        TxtCuentaMayorClave.SetFocus
        Exit Function
    End If
```

```
    If TxtCuentaMayorClave = 0 Then
        MsgBox "La Clave no puede ser Cero", 64
        TxtCuentaMayorClave.SetFocus
        Exit Function
    End If
```

```
If TxtDescripcion = "" Then
    MsgBox "La Descripción no puede ser nula", 64
    TxtDescripcion.SetFocus
    Exit Function
End If
```

```
If InStr(TxtDescripcion, "'") Then
    MsgBox "El Apostrofé no es caracter válido", 64
    TxtDescripcion.SetFocus
    Exit Function
End If
```

```
VerificaDatos = True
End Function
```

```
Private Function InsertaDatos() As Integer
```

```
    Dim MaxCve As Integer
    InsertaDatos = False
    On Error GoTo ManejaError
```

```
    BeginTrans
```

```
    StrSql = "Insert into Cuenta_Mayor ( CMA_CCL_Cve,CMA_Cve, CMA_Tipo_Cuenta,CMA_Descripcion)
```

```
VALUES ("
```

```
    StrSql = StrSql + Str$(CmbCuentaClasificacion.BoundText)
```

```
    StrSql = StrSql + ","
```

```
    StrSql = StrSql + Str$(TxtCuentaMayorClave)
```

```
    StrSql = StrSql + ","
```

```
    If OpbCuentaTotalizadora.Value = True Then
```

```
        StrSql = StrSql + Str$(1)
```

```
    Else
```

```
        StrSql = StrSql + Str$(2)
```

```
    End If
```

```
    StrSql = StrSql + ","
```

```
    StrSql = StrSql + TxtDescripcion
```

```
    StrSql = StrSql + ")"
```

```
    Db.Execute StrSql, ModeSql
```

```
    StrSql = "Insert into subsubcuenta ( "
```

```
    StrSql = StrSql + "SSC_SCU_CCO_CMA_CCL_Cve. "
```

```
    StrSql = StrSql + "SSC_SCU_CCO_CMA_Cve, "
```

```
    StrSql = StrSql + "SSC_SCU_CCO_Cve, "
```

```
    StrSql = StrSql + "SSC_SCU_Cve, "
```

```
    StrSql = StrSql + "SSC_Cve, "
```

```
    StrSql = StrSql + "SSC_Tipo_Cuenta, "
```

```
    StrSql = StrSql + "SSC_Descripcion) VALUES ("
```

```
    StrSql = StrSql + Str$(CmbCuentaClasificacion.BoundText) + ","
```

```
    StrSql = StrSql + Str$(TxtCuentaMayorClave) + ","
```

```
    StrSql = StrSql + "0,0,0,"
```

```
    If OpbCuentaTotalizadora.Value = True Then
```

```
        StrSql = StrSql + Str$(1)
```

```
    Else
```

```
        StrSql = StrSql + Str$(2)
```

```

End If
StrSql = StrSql + "," + TxtDescripcion
StrSql = StrSql + ")"
Db.Execute StrSql, ModeSql

```

```
CommitTrans
```

```
InsertaDatos = True
```

```
Exit Function
```

```
ManejaError:
    Errores (Err)
End Function
```

```
Private Function LimpiaCampos() As Integer
    LimpiaCampos = False

```

```

CmbCuentaClasificacion.BoundText = "
TxtCuentaMayorClave.Mask = ""
TxtCuentaMayorClave.Text = ""
TxtCuentaMayorClave.Mask = "##"
TxtDescripcion.Text = ""
OpbCuentaTotalizadora.Value = True
CmbCuentaClasificacion.SetFocus

```

```
LimpiaCampos = True
```

```
End Function
```

```
Private Function CargaDatos(ClaCCL, ClaCMA) As Integer
    CargaDatos = False

```

```

Set Query = Db.QueryDefs("CMA_Cve")
Query.Parameters("CmaCclCve") = ClaCCL
Query.Parameters("CmaCve") = ClaCMA
Set Rds = Query.OpenRecordset(dbOpenSnapshot, dbReadOnly)

```

```

If Rds.RecordCount = 0 Then
    MsgBox "No se pudo recuperar datos de la Cuenta Mayor", 64
    Exit Function
End If

```

```

CmbCuentaClasificacion.BoundText = Rds("CMA_CCL_Cve")
TxtCuentaMayorClave.Mask = Rds("CMA_Cve")
TxtCuentaMayorClave.Text = Rds("CMA_Cve")
TxtDescripcion = Rds("CMA_Descripcion")

```

```

If Rds("CMA_Tipo_Cuenta") = 1 Then
    OpbCuentaTotalizadora.Value = True
    OpbCuentaDetalle.Value = False
Elseif Rds("CMA_Tipo_Cuenta") = 2 Then
    OpbCuentaTotalizadora.Value = False
    OpbCuentaDetalle.Value = True

```

```

End If
Rds.Close
CargaDatos = True

```

```
End Function
```

```
Private Function BorraDatos(ClaCCL, ClaCMA) As Integer
    On Error GoTo ManejaErrores
    BorraDatos = False

    If Not ExisteRelacion(2, ClaCCL, ClaCMA, 0, 0, 0) Then
        Exit Function
    End If

    StrSql = "Delete * from Cuenta_Mayor where CMA_CCL_Cve = " + ClaCCL
    StrSql = StrSql + " AND CMA_Cve=" + ClaCMA
    Db.Execute StrSql, dbFailOnError

    StrSql = "Delete * from SubSubCuenta where SSC_SCU_CCO_CMA_CCL_Cve = " + ClaCCL
    StrSql = StrSql + " AND SSC_SCU_CCO_CMA_Cve = " + ClaCMA
    StrSql = StrSql + " AND SSC_SCU_CCO_Cve = 0 "
    StrSql = StrSql + " AND SSC_SCU_Cve = 0 "
    StrSql = StrSql + " AND SSC_Cve = 0 "

    Db.Execute StrSql, dbFailOnError

    CdGrid.Refresh
    BorraDatos = True
    Exit Function
ManejaErrores:
    Errores (Err)
End Function
```

```
Private Sub TxtCuentaMayorClave_Change()
    If CmbCuentaClasificacion = "" And TxtCuentaMayorClave = "" Then
        CmbCuentaClasificacion.SetFocus
        MsgBox "Tiene que seleccionar una Cuenta Clasificacion", 64
    Else
        TxtCuentaClave.Text = TxtCuentaMayorClave
    End If
End Sub
```

```
Private Sub TxtDescripcion_Change()
    TxtCuentaDescripcion.Text = TxtDescripcion
End Sub
```

```
Private Function VerificaCombos(Cdata As Control, Verificar As String) As Integer
    VerificaCombos = False

    If Cdata.Recordset.RecordCount = 0 Then
        MsgBox "No hay datos", 64
        Exit Function
    End If

    If Not IsNumeric(Verificar) Then
        Exit Function
    End If

    VerificaCombos = True
End Function
```

Pantalla para poliza modelo

Esta pantalla se usara para realizar consultas de pólizas modelo existentes y para realizar altas de nuevas pólizas modelo a través de la opción de alta.

Quando se realiza la consulta de pólizas modelos se despliegan dos grids sobre la pantalla. En un 'grid se presentarán todas las pólizas modelos existentes, mientras que en el otro se presentará las 'partidas que forman a cada póliza.

Option Explicit

Private Function EnviaReporte(Fecha As Date, folio, TipoPol) As Integer

 MousePointer = 11

 CrpRep.Destination = 0

 Set Query = Db.QueryDefs("TPOCve")

 Query.Parameters("TPOCve") = TipoPol

 Set Rds = Query.OpenRecordset(dbOpenSnapshot, dbReadOnly)

 CrpRep.Connect = "DSN=ODBCReservav4;UID=Admin;DSQ=" + DbDir + DbName

 StrSql = ""

 If Rds("tpo_cheque") = 1 Then 'con cheque

 CrpRep.ReportFileName = RepPath + "RepPolCheque.Rpt"

 Else

 CrpRep.ReportFileName = RepPath + "RepPolDiario.Rpt"

 End If

 StrSql = "{@Fecha}= date "

 StrSql = StrSql + "(" + Str\$(Year(Format(Fecha, "dd/mm/yyyy")))

 StrSql = StrSql + "," + Str\$(Month(Format(Fecha, "dd/mm/yyyy")))

 StrSql = StrSql + "," + Str\$(Day(Format(Fecha, "dd/mm/yyyy")))

 StrSql = StrSql + ") and {poliza.pol_folio} = " + folio

 CrpRep.SelectionFormula = StrSql

 CrpRep.WindowWidth = 600

 CrpRep.WindowHeight = 400

 CrpRep.WindowTop = 50

 CrpRep.WindowLeft = 10

 On Error GoTo ManejaError

 CrpRep.Action = 1

 EnviaReporte = True

 MousePointer = 0

 Exit Function

ManejaError:

 Errores (Err)

 MousePointer = 0

End Function

```
Sub ListaTipoPoliza()  
    Dim Valor As New Valueltem  
    Dim Valores As Valueltems  
  
    StrSql = "Select tpo_cve, tpo_desc_corta from tpo_poliza "  
    Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)  
  
    Set Valores = GrdPoliza.Columns("POL_tpo_cve").Valueltems  
  
    If Rds.RecordCount > 0 Then  
        Rds.MoveFirst  
        While Not Rds.EOF  
            Valor.Value = Val(Rds!tpo_cve)  
            Valor.DisplayValue = Rds("tpo_desc_corta")  
            Valores.Add Valor  
            Rds.MoveNext  
        Wend  
    End If  
    Rds.Close  
  
    Valores.Translate = True  
    Valores.CycleOnClick = True  
    Valores.Validate = True
```

End Sub

```
Sub BorraGrdDetalle()
```

```
    StrSql = ""  
    StrSql = StrSql + "Delete from detalle "  
    StrSql = StrSql + "where det_sec_consec = " + TxtSecuencia  
    Db.Execute StrSql, dbFailOnError
```

End Sub

```
Sub Totales()
```

```
    Dim Cargo As Double  
    Dim Abono As Double  
    Dim RdsAuxi As Recordset  
  
    TxtCargo = Format(0, "$###,###,##0.00")  
    TxtAbono = Format(0, "$###,###,##0.00")  
  
    StrSql = " Select sum(det_debe) as cargo, sum (det_haber) as abono "  
    StrSql = StrSql + " from detalle "  
    StrSql = StrSql + " where det_sec_consec = " + TxtSecuencia  
    Set RdsAuxi = Db.OpenRecordset(StrSql, dbOpenSnapshot)  
  
    TxtCargo = IIf(IsNull(RdsAuxi("cargo")), 0, RdsAuxi("cargo"))  
    TxtAbono = IIf(IsNull(RdsAuxi("abono")), 0, RdsAuxi("abono"))  
    CdDetalle.Recordset.MoveFirst  
    RdsAuxi.Close
```

End Sub

```

Sub Sumas()
  Dim Cargo, Car As Double
  Dim Abono, Ab As Double

  TxtCargoT = Format(0, "Currency")
  TxtAbonoT = Format(0, "Currency")
  Cargo = 0
  Abono = 0
  Aux = 1

  If CdPartida.Recordset.RecordCount <= 0 Then
    Exit Sub
  End If

  CdPartida.Recordset.MoveFirst

  While Aux <= CdPartida.Recordset.RecordCount
    If GrdPartida.Columns("par_debe").Value = "" Then
      Car = 0
    Else
      Car = GrdPartida.Columns("par_debe").Value
    End If

    If GrdPartida.Columns("par_haber").Value = "" Then
      Ab = 0
    Else
      Ab = GrdPartida.Columns("par_haber").Value
    End If

    If GrdPartida.Columns("par_debehaber") Text = 1 Then
      Cargo = Cargo + Car
    Else
      Abono = Abono + Ab
    End If

    If Aux <> CdPartida.Recordset.RecordCount Then
      CdPartida.Recordset.MoveNext
    End If

    Aux = Aux + 1
  Wend

  TxtCargoT = Cargo
  TxtAbonoT = Abono
End Sub

```

```

Sub BorraDetalle()

  If VerificaSelRenglon(GrdDetalle, 1) Then

    If MsgBox("¿Desea borrar el registro?", vbYesNo) = vbYes Then
      MousePointer = 11
      CdDetalle.Recordset.Delete
      Totales
    End If
  End If

```

```
End If
End If
MousePointer = 0
```

End Sub

Private Function CargaGridCuenta() As Integer

```
CargaGridCuenta = False
```

```
StrSql = ""
StrSql = StrSql + "Select * from subsubcuenta "
StrSql = StrSql + "where ssc_tipo_cuenta = 2 " 'detalle
StrSql = StrSql + "order by ssc_scu_cco_cma_ccl_cve, "
StrSql = StrSql + "    ssc_scu_cco_cma_cve, "
StrSql = StrSql + "    ssc_scu_cco_cve, "
StrSql = StrSql + "    ssc_scu_cve, "
StrSql = StrSql + "    ssc_cve "
```

```
If Not DefineDataControl(CdCuentas, DbName, DbDir, StrSql, vbRSTypeDynaset) Then
Exit Function
End If
```

```
CargaGridCuenta = True
```

End Function

Private Function CargaTipoPoliza() As Integer

```
CargaTipoPoliza = False
```

```
StrSql = "TPO"
```

```
If Not DefineDataControl(CdTpoPoliza, DbName, DbDir, StrSql, vbRSTypeDynaset) Then
Exit Function
End If
```

```
CmbTpoPoliza.BoundText = 1
CargaTipoPoliza = True
```

End Function

Private Function CargaGridPoliza() As Integer

```
CargaGridPoliza = False
StrSql = "select * from poliza order by pol_fecha, pol_folio"
```

```
If Not DefineDataControl(CdPoliza, DbName, DbDir, StrSql, vbRSTypeDynaset) Then
Exit Function
End If
```

```
CdPoliza.Refresh
GrdPoliza.Refresh
CargaGridPoliza = True
```

End Function

```

Private Function InsertaPoliza() As Boolean
    Dim folio As Integer

    InsertaPoliza = False
    On Error GoTo ManejaError

    folio = GeneraFolio(TxtFecha)

    If CmbTipoPoliza.BoundText = "" Then
        MsgBox "Debe seleccionar un Tipo de Póliza.", 64
        CmbTipoPoliza.SetFocus
        Exit Function
    End If

    If InStr(TxtConcepto, "'") Then
        MsgBox " El Apostrofé no es caracter válido"
        TxtConcepto.SetFocus
        Exit Function
    End If

    If folio > 0 Then
        StrSql = ""
        StrSql = StrSql + "Insert into poliza "
        StrSql = StrSql + "    (pol_fecha, "
        StrSql = StrSql + "    pol_folio, "
        StrSql = StrSql + "    pol_tip_cve, "
        StrSql = StrSql + "    pol_tpo_cve, "
        StrSql = StrSql + "    pol_are_cve, "
        StrSql = StrSql + "    pol_concepto)"
        StrSql = StrSql + "values (CvDate('" + Format$(TxtFecha, "dd/mm/yy") + "',"
        StrSql = StrSql + "    " + Str$(folio) + ", "
        StrSql = StrSql + "    1, "
        StrSql = StrSql + "    " + Str$(CmbTipoPoliza.BoundText) + ", "
        StrSql = StrSql + "    " + FRMMenuConta.TxtArea + ", "
        StrSql = StrSql + "    " + TxtConcepto + ")"
        Db.Execute StrSql, dbFailOnError

        TxtFolio = folio
        InsertaPoliza = True

        InsertaPartida

    End If
    Exit Function

ManejaError:
    Errores (Err)
End Function
Sub InsertaPartida()
    Dim Tipo As Integer
    Dim concepto As String

    On Error GoTo ManejaError

```

Aux = 1

```
If CdDetalle.Recordset.RecordCount <= 0 Then
    Exit Sub
End If
```

CdDetalle.Recordset.MoveFirst

```
While Aux <= CdDetalle.Recordset.RecordCount
```

```
    If GrdDetalle.Columns("det_debe").Value <> "" Then
        Tipo = 1
    Else
        Tipo = 2
    End If
```

```
    If GrdDetalle.Columns("det_concepto").Value = "" Then
        concepto = TxtConcepto
    Else
        concepto = GrdDetalle.Columns("det_concepto").Value
    End If
```

```
    StrSql = ""
    StrSql = StrSql + "insert into partida "
    StrSql = StrSql + "    (par_pol_fecha, "
    StrSql = StrSql + "    par_pol_folio, "
    StrSql = StrSql + "    par_consec, "
    StrSql = StrSql + "    par_ssc_scu_cco_cma_ccl_cve, "
    StrSql = StrSql + "    par_ssc_scu_cco_cma_cve, "
    StrSql = StrSql + "    par_ssc_scu_cco_cve, "
    StrSql = StrSql + "    par_ssc_scu_cve, "
    StrSql = StrSql + "    par_ssc_cve, "
    StrSql = StrSql + "    par_debehaber, "
    StrSql = StrSql + "    par_concepto)"
    StrSql = StrSql + "values (CVDate("" + Format$(TxtFecha, "dd/mm/yy") + ""),"
    StrSql = StrSql + "    " + TxtFolio + ","
    StrSql = StrSql + "    " + GrdDetalle.Columns("det_consec").Text + ","
    StrSql = StrSql + "    " + GrdDetalle.Columns("det_ccl").Text + ","
    StrSql = StrSql + "    " + GrdDetalle.Columns("det_cma").Text + ","
    StrSql = StrSql + "    " + GrdDetalle.Columns("det_cco").Text + ","
    StrSql = StrSql + "    " + GrdDetalle.Columns("det_scu").Text + ","
    StrSql = StrSql + "    " + GrdDetalle.Columns("det_ssc").Text + ","
    StrSql = StrSql + "    " + Str$(Tipo) + ","
    StrSql = StrSql + "    "" + concepto + "")"
    Db.Execute StrSql, dbFailOnError
```

```
If Tipo = 1 Then
    StrSql = ""
    StrSql = StrSql + " Update partida "
    StrSql = StrSql + " set par_debe = " + Str$(GrdDetalle.Columns("det_debe").Value)
    StrSql = StrSql + " where par_pol_fecha = CVDate("" + Format(TxtFecha, "dd/mm/yy") + "")"
    StrSql = StrSql + " and par_pol_folio = " + TxtFolio
    StrSql = StrSql + " and par_consec = " + Str$(GrdDetalle.Columns("det_consec").Text)
```

```

Else
  strSql = ""
  strSql = strSql + " Update partida "
  strSql = strSql + " set   par_haber = " + Str$(GrdDetalle.Columns("det_haber").Value)
  strSql = strSql + " where par_pol_fecha = CVDate("" + Format(TxtFecha, "dd/mm/yy") + "")"
  strSql = strSql + " and   par_pol_folio = " + TxtFolio
  strSql = strSql + " and   par_consec = " + Str$(GrdDetalle.Columns("det_consec").Text)
End If

```

```

Db.Execute strSql, dbFailOnError

```

```

If Aux <> CdDetalle.Recordset.RecordCount Then
  CdDetalle.Recordset.MoveNext
End If

```

```

Aux = Aux + 1
Wend

```

```

Exit Sub

```

```

ManejaError:
  Errores (Err)
End Sub

```

```

Private Function CargaGridPartida() As Integer
  CargaGridPartida = False

```

```

  strSql = ""
  strSql = strSql + " select * "
  strSql = strSql + " from   partida "
  strSql = strSql + " where  par_pol_fecha = CVDate("" + GrdPoliza.Columns("pol_fecha").Value + "")"
  strSql = strSql + " and    par_pol_folio = " + GrdPoliza.Columns("pol_folio").Value
  strSql = strSql + " order by par_consec"
  If Not DefineDataControl(CdPartida, DbName, DbDir, strSql, vbRSTypeDynaset) Then
    Exit Function
  End If

```

```

  CdPartida.Refresh
  GrdPartida.Refresh
  CargaGridPartida = True
  Sumas
End Function

```

```

Function ActivaPanel(Pnl As Control, Activar As Boolean)

```

```

  If Activar Then
    Pnl.Visible = True
    Pnl.Enabled = True
  Else
    Pnl.Visible = False
    Pnl.Enabled = False
  End If

```

```

End Function

```

```
Sub SeleccionaCuenta()  
  GrdDetalle.Columns("det_ccl").Value = GrdCuentas.Columns("ssc_scu_cco_cma_ccl_cve").Value  
  GrdDetalle.Columns("det_cma").Value = GrdCuentas.Columns("ssc_scu_cco_cma_cve").Value  
  GrdDetalle.Columns("det_cco").Value = GrdCuentas.Columns("ssc_scu_cco_cve").Value  
  GrdDetalle.Columns("det_scu").Value = GrdCuentas.Columns("ssc_scu_cve").Value  
  GrdDetalle.Columns("det_ssc").Value = GrdCuentas.Columns("ssc_cve").Value  
  GrdDetalle.Columns("det_descripcion").Value = GrdCuentas.Columns("ssc_descripcion").Value  
  Aux = ActivaPanel(PnlGrdCuentas, False)  
End Sub
```

Function VerificaDatos() As Boolean

```
  VerificaDatos = False  
  
  If Not VerificaCuenta Then  
    MsgBox "No existe esta cuenta, o no es de Detalle2.", 64  
    Exit Function  
  End If  
  
  If (GrdDetalle.Columns("det_debe").Value = "" And GrdDetalle.Columns("det_haber").Value = "") Or  
    (GrdDetalle.Columns("det_debe").Value <> "" And GrdDetalle.Columns("det_haber").Value <> "")  
Then  
  MsgBox "Debe teclear o un Cargo o un Abono.", 64  
  Exit Function  
End If  
  
  VerificaDatos = True
```

Function VerificaSelRenglon(Grd As Control, Opcion) As Integer

```
  VerificaSelRenglon = False  
  
  If Opcion = 1 Then 'borrar  
    If Grd.SelBookmarks.Count > 1 Then  
      Beep  
      MsgBox "Debe seleccionar solo un registro a dar de baja", 64  
      Exit Function  
    ElseIf Grd.SelBookmarks.Count = 0 Then  
      Beep  
      MsgBox "Debe seleccionar al menos un registro a dar de baja", 64  
      Exit Function  
    End If  
  ElseIf Opcion = 2 Then 'modificar  
    If Grd.SelBookmarks.Count > 1 Then  
      Beep  
      MsgBox "Debe seleccionar solo un registro para modificar", 64  
      Exit Function  
    ElseIf Grd.SelBookmarks.Count = 0 Then  
      Beep  
      MsgBox "Debe seleccionar al menos un registro para modificar", 64  
      Exit Function  
    End If  
  ElseIf Opcion = 3 Then 'consultar
```

```
If Grd.SelBookmarks.Count > 1 Then
    Beep
    MsgBox "Debe seleccionar solo un registro para consultar", 64
    Exit Function
ElseIf Grd.SelBookmarks.Count = 0 Then
    Beep
    MsgBox "Debe seleccionar al menos un registro para consultar", 64
    Exit Function
End If
Elseif Opcion = 4 Then 'imprimir
    If Grd.SelBookmarks.Count > 1 Then
        Beep
        MsgBox "Debe seleccionar solo un registro para imprimir", 64
        Exit Function
    ElseIf Grd.SelBookmarks.Count = 0 Then
        Beep
        MsgBox "Debe seleccionar al menos un registro para imprimir", 64
        Exit Function
    End If
End If

VerificaSelRenglon = True

End Function
Private Sub CmdDetalleBaja_Click()
    BorraDetalle
End Sub
Private Sub CmdPartidaRep_Click()
    If TxtFolio <> "" Then
        If Not EnviaReporte(TxtFecha, TxtFolio, CmbTipoPoliza BoundText) Then
            Exit Sub
        End If
    Else
        MsgBox "No se ha generado aún la Póliza."
    End If
End Sub
Private Sub CmdPolizaAceptar_Click()

    If (TxtCargo = "" Or TxtAbono = "") Then
        MsgBox "Debe teclear al menos un Cargo y un Abono.", 64
        Exit Sub
    End If

    If Not VerificaDatos Then
        Exit Sub
    End If

    CdDetalle.Recordset.MovePrevious
    CdDetalle.Recordset.MoveNext
    If TxtCargo <> TxtAbono Then
        MsgBox "La suma de Cargos debe ser igual al de abonos.", 64
        Exit Sub
    End If
```

```
If Not InsertaPoliza() Then
    MsgBox "Error al insertar la póliza.", 64
    Exit Sub
End If
xerox 13 marzo 7 pm intercontinental nikko castillo b
MsgBox "La póliza ha sido dado de alta.", 64
```

```
PnlGrdDetalle.Enabled = False
PnlPoliza.Enabled = False
```

```
CmdDetalleBaja.Enabled = False
CmdPolizaAceptar.Enabled = False
CmdPartidaSalir.Enabled = True
```

```
CdPoliza.Refresh
GrdPoliza.Refresh
CdPartida.Refresh
GrdPartida.Refresh
```

End Sub

Private Sub CmdPolizaAlta_Click()

```
If Not VerificaFecha(Date) Then
    Exit Sub
End If
```

```
Aux = ActivaPanel(PnlTrabajoPoliza, True)
Aux = ActivaPanel(PnlGrdPoliza, False)
Aux = ActivaPanel(PnlGrdPartida, False)
Aux = ActivaPanel(PnlGrdDetalle, True)
Aux = ActivaPanel(PnlPoliza, True)
CmdDetalleBaja.Enabled = True
CmdPolizaAceptar.Enabled = True
CmdPartidaSalir.Enabled = True
TxtFecha = Format$(Date, "dd/mm/yy")
TxtSecuencia = GeneraSecuencia
TxtConcepto = ""
TxtFolio = ""
TxtCargo = ""
TxtAbono = ""
```

```
If Not CargaGridDetalle() Then
    Exit Sub
End If
```

End Sub

Private Sub CmdPolizaRep_Click()

```
If VerificaSelRenglon(GrdPoliza, 4) Then
    If Not EnviaReporte(Format(GrdPoliza.Columns("pol_fecha").Value, "dd/mm/yyyy"),
GrdPoliza.Columns("pol_folio").Value, GrdPoliza.Columns("pol_tpo_cve").Value) Then
        Exit Sub
```

```
End If
End If
End Sub
```

```
Private Sub CmdFormaSalir_Click()
    Me.Hide
End Sub
```

```
Private Sub CmdPartidaSalir_Click()
```

```
    Aux = ActivaPanel(PnlTrabajoPoliza, False)
    Aux = ActivaPanel(PnlGrdPoliza, True)
    Aux = ActivaPanel(PnlGrdPartida, True)
    BorraGrdDetalle
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Me.WindowState = 2
    On Error GoTo ManejaErrores
```

```
    Aux = ActivaPanel(PnlTrabajoPoliza, False)
    Aux = ActivaPanel(PnlGrdCuentas, False)
```

```
    If Not CargaGridPoliza() Then
        Exit Sub
    End If
```

```
    GrdPoliza.AllowAddNew = False
```

```
    If Not CargaGridCuenta() Then
        Exit Sub
    End If
```

```
    If Not CargaTipoPoliza() Then
        Exit Sub
    End If
```

```
    ListaTipoPoliza
```

```
    Exit Sub
```

```
ManejaErrores:
```

```
    Errores (Err)
```

```
End Sub
```

```
Private Function CargaGridDetalle() As Integer
```

```
    CargaGridDetalle = False
```

```
    strSql = "Select * from detalle where det_sec_consec =" + TxtSecuencia
```

```
    If Not DefineDataControl(CdDetalle, DbName, DbDir, strSql, vbRSTypeDynaset) Then
        Exit Function
```

```
    End If
    CdDetalle.Refresh
    GrdDetalle.Refresh
    CargaGridDetalle = True
```

```
End Function
```

```
Private Function VerificaCuenta() As Boolean
    Dim Rds1 As Recordset
```

```
    VerificaCuenta = False
```

```
    If GrdDetalle.Columns("det_ccl").Text = "" Then
        GrdDetalle.Columns("det_ccl").Text = "0"
    End If
    If GrdDetalle.Columns("det_cma").Text = "" Then
        GrdDetalle.Columns("det_cma").Text = "0"
    End If
    If GrdDetalle.Columns("det_cco").Text = "" Then
        GrdDetalle.Columns("det_cco").Text = "0"
    End If
    If GrdDetalle.Columns("det_scu").Text = "" Then
        GrdDetalle.Columns("det_scu").Text = "0"
    End If
    If GrdDetalle.Columns("det_ssc").Text = "" Then
        GrdDetalle.Columns("det_ssc").Text = "0"
    End If
```

```
    StrSql = ""
    StrSql = StrSql + " Select ssc_descripcion "
    StrSql = StrSql + " from subsubcuenta "
    StrSql = StrSql + " where ssc_scu_cco_cma_ccl_cve = " + GrdDetalle.Columns("det_ccl").Text
    StrSql = StrSql + " and ssc_scu_cco_cma_cve = " + GrdDetalle.Columns("det_cma").Text
    StrSql = StrSql + " and ssc_scu_cco_cve = " + GrdDetalle.Columns("det_cco").Text
    StrSql = StrSql + " and ssc_scu_cve = " + GrdDetalle.Columns("det_scu").Text
    StrSql = StrSql + " and ssc_cve = " + GrdDetalle.Columns("det_ssc").Text
    StrSql = StrSql + " and ssc_tipo_cuenta = 2" 'detalle
    Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
    If Rds1.EOF Then
        GrdDetalle.Columns("det_descripcion").Text = ""
        Exit Function
    Else
        If GrdDetalle.Columns("det_descripcion").Text <> Rds1("ssc_descripcion") Then
            GrdDetalle.Columns("det_descripcion").Text = Rds1("ssc_descripcion")
        End If
    End If
```

```
    VerificaCuenta = True
```

```
End Function
```

```
Private Sub GrdCuentas_DbClick()
```

```
    SeleccionaCuenta
```

```
End Sub
```

```
Private Sub GrdCuentas_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii = 13 Then
```

```
        SeleccionaCuenta
```

```
    End If
```

```
End Sub
```

```
Private Sub GrdCuentas_LostFocus()
    Aux = ActivaPanel(PnlGrdCuentas, False)
End Sub
```

```
Private Sub GrdDetalle_AfterColUpdate(ByVal ColIndex As Integer)
```

```
    If ColIndex >= 0 And ColIndex <= 4 Then
        VerificaCuenta
    End If
```

```
End Sub
```

```
Private Sub GrdDetalle_AfterUpdate()
```

```
    Totales
```

```
End Sub
```

```
Private Sub GrdDetalle_BeforeColEdit(ByVal ColIndex As Integer, ByVal KeyAscii As Integer, Cancel As Integer)
```

```
    If ColIndex >= 6 And ColIndex <= 8 Then
        If IsNull(GrdDetalle.Columns("det_descripcion").Value) Or
        GrdDetalle.Columns("det_descripcion").Value = "" Then
            MsgBox "Debe seleccionar una Cuenta primero.", 64
            Cancel = True
        End If
    End If
```

```
End Sub
```

```
Private Sub GrdDetalle_BeforeUpdate(Cancel As Integer)
```

```
    If Not VerificaDatos Then
        Cancel = True
        Exit Sub
    End If
```

```
    If GrdDetalle.Columns("det_debe").Value <> "" Then
        GrdDetalle.Columns("det_debehaber").Value = 1
    Else
        GrdDetalle.Columns("det_debehaber").Value = 2
    End If
```

```
    If GrdDetalle.AddNewMode <> 0 Then 'Es nuevo el registro
        GrdDetalle.Columns("det_consec").Value = GeneraConsec
        GrdDetalle.Columns("det_sec_consec").Value = TxtSecuencia
    End If
```

```
End Sub
```

```
Private Sub GrdDetalle_ButtonClick(ByVal ColIndex As Integer)
```

```
    'ok
```

```
    Dim Co As Column
```

```
    Set Co = GrdDetalle.Columns(ColIndex - 1)
```

```
    PnlGrdCuentas.Left = GrdDetalle.Left + Co.Left + Co.Width
    PnlGrdCuentas.Top = GrdDetalle.Top + GrdDetalle.RowTop(GrdDetalle.Row) + 10
    Aux = ActivaPanel(PnlGrdCuentas, True)
    GrdCuentas.SetFocus
```

```
End Sub
```

```
Private Sub GrdDetalle_Error(ByVal DataError As Integer, Response As Integer)
    If DataError = 16389 Then
        Response = 0
    End If
End Sub
```

```
Private Sub GrdDetalle_Scroll(Cancel As Integer)
    Aux = ActivaPanel(PnlGrdCuentas, False)
End Sub
```

```
Private Sub GrdPoliza_SelChange(Cancel As Integer)
    'ok
    If VerificaSelRenglon(GrdPoliza, 3) Then
        If Not CargaGridPartida() Then
            Exit Sub
        End If
    End If

End Sub
```

```
Function GeneraConsec() As Integer
    Dim Rds1 As Recordset

    On Error GoTo ManejaErrores

    StrSql = ""
    StrSql = StrSql + "Select count (*) as cuenta "
    StrSql = StrSql + "from detalle "
    StrSql = StrSql + "where det_sec_consec = " + TxtSecuencia
    Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)

    If Rds1("cuenta") = 0 Then
        GeneraConsec = 1
    Else
        GeneraConsec = Rds1("cuenta") + 1
    End If

    Exit Function
```

```
ManejaErrores:
    Errores (Err)
End Function
```

```
Function GeneraSecuencia() As Integer
    Dim Rds1 As Recordset

    On Error GoTo ManejaErrores

    StrSql = ""
    StrSql = StrSql + "Select max (sec_consec) as consec "
    StrSql = StrSql + "from secuencia "
    Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)

    If IsNull(Rds1("consec")) Then
        GeneraSecuencia = 1
        StrSql = "Insert into secuencia (sec_consec) values (1)"
    Else
```

```
GeneraSecuencia = Rds1("consec") + 1  
StrSql = "Update secuencia set sec_consec = " + Str$(GeneraSecuencia)  
End If
```

```
Db.Execute StrSql, dbFailOnError
```

```
Exit Function
```

```
ManejaErrores:
```

```
  Errores (Err)
```

```
End Function
```

Pantalla para usuarios

'Esta pantalla se usara para realizar actualizaciones (altas, bajas, modificaciones) al catálogo de usuarios que pueden operar el sistema. A través de esta pantalla se darán las facultades a cada uno de los usuarios, dependiendo de las facultades que se le asignen podrán acceder a los diferentes módulos. Las facultades asignadas dependerán del área y grupo seleccionado.

```
Private Function Boton(Encender As Boolean)
```

```
    If Encender = True Then  
        CmdBaja.Enabled = True  
        CmdModificar.Enabled = True  
        CmdAlta.Enabled = True  
        CmdSalir.Enabled = True  
    Else  
        CmdBaja.Enabled = False  
        CmdModificar.Enabled = False  
        CmdAlta.Enabled = False  
        CmdSalir.Enabled = False  
    End If
```

```
End Function
```

```
Private Function ActualizaDatos() As Integer
```

```
    ActualizaDatos = False  
    On Error GoTo ManejaError
```

```
    BeginTrans
```

```
    strSql = "Update Usuario Set "
```

```
    strSql = strSql + " USU_ARE_Cve = " + CmbArea.BoundText
```

```
    strSql = strSql + ", USU_GRU_Cve = " + CmbGrupo.BoundText
```

```
    strSql = strSql + ", USU_Nombre = " + TxtNombres.Text + ""
```

```
    strSql = strSql + ", USU_Password = " + UCase(TxtPassFin.Text) + ""
```

```
    strSql = strSql + " Where USU_Cve = " + TxtClaves.Text + ""
```

```
    Db.Execute strSql, dbFailOnError
```

```
    CommitTrans
```

```
    ActualizaDatos = True
```

```
Exit Function
```

```
ManejaError:
```

```
    Errores (Err)
```

```
    Rollback
```

```
End Function
```

```
Private Sub CmdAlta_Click()
```

```
    Dim x
```

```
    Boton (False)
```

```
    PnlActualiza.Visible = True
```

```
    PnlModifica.Visible = False
```

```
    PnlAlta.Visible = True
```

```
    PnlActTitulo.Caption = "Alta de Usuarios "
```

```
    If Not LimpiaCampos() Then
```

```
        Exit Sub
```

```
    End If
```

```
End Sub
```

```
Private Sub CmdAltAlta_Click()
```

```
    If Not VerificaDatos() Then
        Exit Sub
    End If
```

```
    If Not InsertaDatos() Then
        If Not LimpiaCampos Then
            Exit Sub
        End If
        MsgBox "Error al insertar El Usuario ", 64
        Exit Sub
    End If
```

```
    Beep
    MsgBox "El Usuario ha sido dado de alta", 64
```

```
    If Not LimpiaCampos() Then
        Exit Sub
    End If
```

```
    CdGrid.Refresh
    GrdCat.Refresh
```

```
End Sub
```

```
Private Sub CmdAltSalir_Click()
```

```
    PnlActualiza.Visible = False
    Boton (True)
```

```
End Sub
```

```
Private Sub CmdBaja_Click()
```

```
    If GrdCat.SelBookmarks.Count > 1 Then
        Beep
        MsgBox "Debe seleccionar solo un registro a dar de baja", 64
        Exit Sub
```

```
    ElseIf GrdCat.SelBookmarks.Count = 0 Then
        Beep
        MsgBox "Debe seleccionar al menos un registro a dar de baja", 64
        Exit Sub
```

```
    End If
    If MsgBox("Esta seguro de eliminar El Usuario seleccionado", vbOKCancel) = vbOK Then
        If Not BorraDatos(GrdCat.Columns("USU_Cve").Value) Then
            MsgBox "ERROR] No pudo realizarse proceso de borrado"
        End If
    End If
```

```
End Sub
```

```
Private Sub CmdModCancelar_Click()
```

```
    TxtClaves.Enabled = True
    PnlActualiza.Visible = False
    Boton (True)
```

```
End Sub
```

```
Private Sub CmdModificar_Click()
    Dim x

    If GrdCat.SelBookmarks.Count > 1 Then
        Beep
        MsgBox "Debe seleccionar solo un registro a modificar", 64
        Exit Sub
    ElseIf GrdCat.SelBookmarks.Count = 0 Then
        Beep
        MsgBox "Debe seleccionar al menos un registro a modificar", 64
        Exit Sub
    End If

    Boton (False)
    PnlActualiza.Visible = True
    PnlModifica.Visible = True
    PnlAlta.Visible = False
    PnlActTitulo.Caption = "Modificación de Usuarios "

    If Not LimpiaCampos() Then
        Exit Sub
    End If

    If Not CargaDatos(GrdCat.Columns("USU_Cve").Value) Then
        Exit Sub
    End If

    TxtClaves.Enabled = False

End Sub
```

```
Private Sub CmdModModificar_Click()
```

```
    If Not VerificaDatos() Then
        Exit Sub
    End If
```

```
    If Not ActualizaDatos() Then
        Exit Sub
    End If
```

```
    GrdCat.Refresh
    CdGrid.Refresh
    TxtClaves.Enabled = True
    PnlActualiza.Visible = False
    Boton (True)
```

```
End Sub
```

```
Private Sub CmdSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    'Se define le origen de los datos del grid
    Me.WindowState = 2
```

PnlActualiza.Visible = False

If Not CargaGrid() Then
 Exit Sub
End If

StrSql = "ARE "
 If Not DefineDataControl(CdArea, DbName, DbDir, StrSql, vbRSTypeSnapShot) Then
 Exit Sub
End If
CdArea.Refresh

StrSql = "GRU "
 If Not DefineDataControl(CdGrupo, DbName, DbDir, StrSql, vbRSTypeSnapShot) Then
 Exit Sub
End If
CdGrupo.Refresh

If Not AsignaValuelItem(GrdCat, "USU_ARE_Cve", "ARE", 0, 2) Then
 Exit Sub
End If

If Not AsignaValuelItem(GrdCat, "USU_GRU_Cve", "GRU", 0, 2) Then
 Exit Sub
End If

End Sub

Private Function CargaGrid() As Integer
 CargaGrid = False

StrSql = "USU"
 If Not DefineDataControl(CdGrid, DbName, DbDir, StrSql, vbRSTypeDynaset) Then
 Exit Function
End If

CargaGrid = True
End Function

Private Function VerificaDatos() As Integer
 VerificaDatos = False

If StrComp(TxtPassInicial.Text, TxtPassFin.Text, 1) <> 0 Then
 MsgBox "El Password no es el mismo", 64
 TxtPassFin.Text = ""
 TxtPassFin.SetFocus
 Exit Function
End If

If IsNumeric(CmbArea) Then
 MsgBox "Debe Elegir del combo"
 CmbArea.SetFocus
 Exit Function
End If

```
If CmbArea.BoundText = "" Then
    MsgBox "El Area no puede ser nula", 64
    CmbArea.SetFocus
    Exit Function
End If

If CmbArea.BoundText <> "" Then
    If Not VerificaCombos(CdArea, CmbArea.BoundText) Then
        MsgBox "Debe Seleccionar de la lista", 64
        CmbArea.SetFocus
        Exit Function
    End If
End If

If IsNumeric(CmbGrupo) Then
    MsgBox "Debe Elegir del combo"
    CmbGrupo.SetFocus
    Exit Function
End If

If CmbGrupo.BoundText = "" Then
    MsgBox "El Grupo no puede ser nulo", 64
    CmbGrupo.SetFocus
    Exit Function
End If

If CmbGrupo.BoundText <> "" Then
    If Not VerificaCombos(CdGrupo, CmbGrupo.BoundText) Then
        MsgBox "Debe Seleccionar de la lista", 64
        CmbGrupo.SetFocus
        Exit Function
    End If
End If

If StrComp(TxtClaves, "ACRATA", 1) = 0 Then
    MsgBox "La Clave no puede se ACRATA", 64
    TxtClaves.SetFocus
    Exit Function
End If

If TxtClaves = "" Then
    MsgBox "El Nombre no puede ser nulo", 64
    TxtClaves.SetFocus
    Exit Function
End If

If InStr(TxtClaves, "'") Then
    MsgBox "El Apostrofé no es caracter válido", 64
    TxtClaves.SetFocus
    Exit Function
End If

If TxtNombres = "" Then
```



```
MsgBox "El Nombre no puede ser nulo", 64
TxtNombres.SetFocus
Exit Function
End If

If InStr(TxtNombres, "'") Then
MsgBox "El Apostrofé no es caracter válido", 64
TxtNombres.SetFocus
Exit Function
End If

If TxtPassInicial = "" Then
MsgBox "Debes Escribir un Password", 64
TxtPassInicial.SetFocus
Exit Function
End If

If InStr(TxtPassInicial, "'") Then
MsgBox "El Apostrofé no es caracter válido", 64
TxtPassInicial.SetFocus
Exit Function
End If

If TxtPassFin = "" Then
MsgBox "Debes Escribir un Password", 64
TxtPassInicial.SetFocus
Exit Function
End If

If InStr(TxtPassInicial, "'") Then
MsgBox "El Apostrofé no es caracter válido", 64
TxtPassInicial.SetFocus
Exit Function
End If

VerificaDatos = True
End Function

Private Function InsertaDatos() As Integer
Dim MaxCve As Integer
InsertaDatos = False
On Error GoTo ManejaError

BeginTrans
StrSql = "Insert into Usuario
(USU_Cve,USU_Password,USU_Nombre,USU_ARE_Cve,USU_GRU_Cve) VALUES ("
StrSql = StrSql + """"
StrSql = StrSql + Trim(UCCase(TxtClaves))
StrSql = StrSql + """, ""
StrSql = StrSql + Trim(UCCase(TxtPassFin))
StrSql = StrSql + """, ""
StrSql = StrSql + TxtNombres
StrSql = StrSql + """, ""
```

```
StrSql = StrSql + Str$(CmbArea.BoundsText)
StrSql = StrSql + ","
StrSql = StrSql + Str$(CmbGrupo.BoundsText)
StrSql = StrSql + ")"
Db.Execute StrSql, ModeSql
```

```
CommitTrans
```

```
InsertaDatos = True
```

```
Exit Function
```

```
ManejaError:
```

```
    Errores (Err)
```

```
    Rollback
```

```
End Function
```

```
Private Function LimpiaCampos() As Integer
```

```
    LimpiaCampos = False
```

```
    CmbArea.BoundsText = " "
```

```
    CmbGrupo.BoundsText = " "
```

```
    TxtClaves.Text = ""
```

```
    TxtNombres.Text = ""
```

```
    TxtPassInicial.Text = ""
```

```
    TxtPassFin.Text = ""
```

```
    CmbArea.SetFocus
```

```
    LimpiaCampos = True
```

```
End Function
```

```
Private Function CargaDatos(Clave) As Integer
```

```
    CargaDatos = False
```

```
    Set Query = Db.QueryDefs("USUCve")
```

```
    Query.Parameters("USUCve") = Clave
```

```
    Set Rds = Query.OpenRecordset(dbOpenSnapshot, dbReadOnly)
```

```
    If Rds.RecordCount = 0 Then
```

```
        MsgBox "No se pudo recuperar datos del Usuario", 64
```

```
        Exit Function
```

```
    End If
```

```
    CmbArea.BoundsText = Rds("USU_ ARE_Cve")
```

```
    CmbGrupo.BoundsText = Rds("USU_ GRU_Cve")
```

```
    TxtClaves.Text = Rds("USU_Cve")
```

```
    TxtNombres.Text = Rds("USU_Nombre")
```

```
    TxtPassInicial.Text = Rds("USU_Password")
```

```
    TxtPassFin.Text = Rds("USU_Password")
```

```
    Rds.Close
```

```
    CargaDatos = True
```

```
End Function
```

```
Private Function BorraDatos(Clave) As Integer
    On Error GoTo ManejaErrores
    BorraDatos = False
```

```
    StrSql = "Delete * from Usuario where USU_Cve = " + "" + Clave + ""
    Db.Execute StrSql, dbFailOnError
```

```
    CdGrid.Refresh
    BorraDatos = True
    Exit Function
```

```
ManejaErrores:
    Errores (Err)
End Function
```

```
Private Function VerificaCombos(Cdata As Control, Verificar As String) As Integer
```

```
    VerificaCombos = False
    If Cdata.Recordset.RecordCount = 0 Then
        MsgBox "No hay datos", 64
        Exit Function
    End If
```

```
    If Not IsNumeric(Verificar) Then
        Exit Function
    End If
```

```
    VerificaCombos = True
End Function
```

Pantalla para captura de estados financieros

A través de esta pantalla se podrán capturar los datos requeridos para generar diversos tipos de estados financieros para cada una de las salas. Esta pantalla da la opción de diferentes tipos de 'costos':

- Costos de ingresos
- Costos variables
- Costos fijos
- Gastos de operación
- Productos financieros
- No deducibles.

Option Explicit

Dim Lugar As Long

Dim FechaC As Date

Private Function BorraParametro()

On Error GoTo ManejaErrores

StrSql = ""

StrSql = "Delete * from costo_parametro_mes where cpm_ssc_scu_cco_cma_ccl_cve = " + GrdParametro.Columns(0).Value

StrSql = StrSql + " AND cpm_ssc_scu_cco_cma_cve=" + GrdParametro.Columns(1).Value

StrSql = StrSql + " AND cpm_ssc_scu_cco_cve=" + GrdParametro.Columns(2).Value

StrSql = StrSql + " AND cpm_ssc_scu_cve=" + GrdParametro.Columns(3).Value

StrSql = StrSql + " AND cpm_ssc_cve=" + GrdParametro.Columns(4).Value

StrSql = StrSql + " AND cpm_par_cve=" + GrdParametro.Columns(5).Value

StrSql = StrSql + " AND cpm_tco_cve=" + GrdParametro.Columns(6).Value

StrSql = StrSql + " AND cpm_fecha=CvDate(" + GrdParametro.Columns(7).Value + ")"

Db.Execute StrSql, dbFailOnError

Exit Function

ManejaErrores:

Errores (Err)

End Function

Private Function BorraSala()

On Error GoTo ManejaErrores

StrSql = ""

StrSql = "Delete * from costo_sala_mes"

StrSql = StrSql + " Where csm_ssc_scu_cco_cma_ccl_cve = " + GrdSala.Columns(0).Value

StrSql = StrSql + " AND csm_ssc_scu_cco_cma_cve=" + GrdSala.Columns(1).Value

StrSql = StrSql + " AND csm_ssc_scu_cco_cve=" + GrdSala.Columns(2).Value

StrSql = StrSql + " AND csm_ssc_scu_cve=" + GrdSala.Columns(3).Value

StrSql = StrSql + " AND csm_ssc_cve=" + GrdSala.Columns(4).Value

StrSql = StrSql + " AND csm_sal_cve=" + GrdSala.Columns(5).Value

StrSql = StrSql + " AND csm_tco_cve=" + GrdSala.Columns(6).Value

StrSql = StrSql + " AND csm_fecha=CvDate(" + GrdSala.Columns(7).Value + ")"

Db.Execute StrSql, dbFailOnError

Exit Function

ManejaErrores:

Errores (Err)

End Function

Private Function BorraTodos()

On Error GoTo ManejaErrores

StrSql = ""

StrSql = "Delete * from costo_cuenta_mes where ccm_ssc_scu_cco_cma_ccl_cve = " +
GrdTodos.Columns(0).Value

StrSql = StrSql + " AND ccm_ssc_scu_cco_cma_cve=" + GrdTodos.Columns(1).Value

StrSql = StrSql + " AND ccm_ssc_scu_cco_cve=" + GrdTodos.Columns(2).Value

StrSql = StrSql + " AND ccm_ssc_scu_cve=" + GrdTodos.Columns(3).Value

StrSql = StrSql + " AND ccm_ssc_cve=" + GrdTodos.Columns(4).Value

StrSql = StrSql + " AND ccm_tco_cve=" + GrdTodos.Columns(5).Value

StrSql = StrSql + " AND ccm_fecha=C\Date("" + GrdTodos.Columns(6).Value + "")"

Db.Execute StrSql, dbFailOnError

Exit Function

ManejaErrores:

Errores (Err)

End Function

Private Function InsertaParametro()

On Error GoTo ManejaError

StrSql = ""

StrSql = StrSql + "Insert into costo_parametro_mes("

StrSql = StrSql + " cpm_ssc_scu_cco_cma_ccl_cve, "

StrSql = StrSql + " cpm_ssc_scu_cco_cma_cve, "

StrSql = StrSql + " cpm_ssc_scu_cco_cve, "

StrSql = StrSql + " cpm_ssc_scu_cve, "

StrSql = StrSql + " cpm_ssc_cve, "

StrSql = StrSql + " cpm_par_cve, "

StrSql = StrSql + " cpm_tco_cve, "

StrSql = StrSql + " cpm_fecha, "

StrSql = StrSql + " cpm_presupuesto) "

StrSql = StrSql + "values (" + GrdSubSubcuenta.Columns(0).Text + ", "

StrSql = StrSql + " " + GrdSubSubcuenta.Columns(1).Text + ", "

StrSql = StrSql + " " + GrdSubSubcuenta.Columns(2).Text + ", "

StrSql = StrSql + " " + GrdSubSubcuenta.Columns(3).Text + ", "

StrSql = StrSql + " " + GrdSubSubcuenta.Columns(4).Text + ", "

StrSql = StrSql + " " + CmbParametro.BoundText + ", "

StrSql = StrSql + " " + CmbTipoCosto.BoundText + ", "

StrSql = StrSql + " C\Date("" + Format\$(FechaC, "dd/mm/yy") + ""), "

StrSql = StrSql + " " + Str\$(0) + ")"

Db.Execute StrSql, dbFailOnError

Exit Function

ManejaError:

Errores (Err)

End Function

Private Function InsertaSala()

On Error GoTo ManejaError

```

StrSql = ""
StrSql = StrSql + "Insert into costo_sala_mes( "
StrSql = StrSql + "        csm_ssc_scu_cco_cma_ccl_cve,"
StrSql = StrSql + "        csm_ssc_scu_cco_cma_cve,"
StrSql = StrSql + "        csm_ssc_scu_cco_cve,"
StrSql = StrSql + "        csm_ssc_scu_cve,"
StrSql = StrSql + "        csm_ssc_cve,"
StrSql = StrSql + "        csm_sal_cve,"
StrSql = StrSql + "        csm_tco_cve,"
StrSql = StrSql + "        csm_fecha,"
StrSql = StrSql + "        csm_presupuesto) "
StrSql = StrSql + "values (" + GrdSubSubcuenta.Columns(0).Text + ","
StrSql = StrSql + "        " + GrdSubSubcuenta.Columns(1).Text + ","
StrSql = StrSql + "        " + GrdSubSubcuenta.Columns(2).Text + ","
StrSql = StrSql + "        " + GrdSubSubcuenta.Columns(3).Text + ","
StrSql = StrSql + "        " + GrdSubSubcuenta.Columns(4).Text + ","
StrSql = StrSql + "        " + CmbSala.BoundText + ","
StrSql = StrSql + "        " + CmbTipoCosto.BoundText + ","
StrSql = StrSql + "        CVDate("" + Format$(FechaC, "dd/mm/yy") + ""),"
StrSql = StrSql + "        " + Str$(0) + ")"
```

Db.Execute StrSql, dbFailOnError

Exit Function

ManejaError:

Errores (Err)

End Function

Private Function InsertaTodos()

On Error GoTo ManejaError

```

StrSql = ""
StrSql = StrSql + "Insert into costo_cuenta_mes( "
StrSql = StrSql + "        ccm_ssc_scu_cco_cma_ccl_cve,"
StrSql = StrSql + "        ccm_ssc_scu_cco_cma_cve,"
StrSql = StrSql + "        ccm_ssc_scu_cco_cve,"
StrSql = StrSql + "        ccm_ssc_scu_cve,"
StrSql = StrSql + "        ccm_ssc_cve,"
StrSql = StrSql + "        ccm_tco_cve,"
StrSql = StrSql + "        ccm_fecha,"
StrSql = StrSql + "        ccm_presupuesto) "
StrSql = StrSql + "values (" + GrdSubSubcuenta.Columns(0).Text + ","
StrSql = StrSql + "        " + GrdSubSubcuenta.Columns(1).Text + ","
StrSql = StrSql + "        " + GrdSubSubcuenta.Columns(2).Text + ","
StrSql = StrSql + "        " + GrdSubSubcuenta.Columns(3).Text + ","
StrSql = StrSql + "        " + GrdSubSubcuenta.Columns(4).Text + ","
StrSql = StrSql + "        " + CmbTipoCosto.BoundText + ","
StrSql = StrSql + "        CVDate("" + Format$(FechaC, "dd/mm/yy") + ""),"
StrSql = StrSql + "        " + Str$(0) + ")"
```

Db.Execute StrSql, dbFailOnError

```
Exit Function
ManejaError:
  Errores (Err)
End Function
```

```
Sub Sumas(Lugar)
  Dim Pres As Double
```

```
  TxtTotal = Format(0, "Currency")
  Pres = 0
  Aux = 1
```

```
  If Lugar = 1 Then
```

```
    StrSql = ""
    StrSql = StrSql + "Select csm_presupuesto as Pres from costo_sala_mes "
    StrSql = StrSql + " where csm_sal_cve=" + CmbSala.BoundText
    StrSql = StrSql + " and csm_tco_cve=" + CmbTipoCosto.BoundText
    StrSql = StrSql + " and csm_fecha =CVDate(" + Format(FechaC) + "")"
```

```
    Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
    If Rds.RecordCount <> 0 Then
```

```
      While Not Rds.EOF
```

```
        If IsNull(Rds("Pres")) Then
          Pres = Pres + 0
        Else
          Pres = Pres + Rds("Pres")
        End If
```

```
        Rds.MoveNext
```

```
      Wend
      Rds.Close
```

```
    End If
```

```
  ElseIf Lugar = 2 Then
```

```
    StrSql = ""
    StrSql = StrSql + "Select cpm_presupuesto as Pres from costo_parametro_mes "
    StrSql = StrSql + " where cpm_par_cve=" + CmbParametro.BoundText
    StrSql = StrSql + " and cpm_tco_cve=" + CmbTipoCosto.BoundText
    StrSql = StrSql + " and cpm_fecha =CVDate(" + Format(FechaC) + "")"
```

```
    Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
    If Rds.RecordCount <> 0 Then
```

```
      While Not Rds.EOF
```

```
    If IsNull(Rds("Pres")) Then
        Pres = Pres + 0
    Else
        Pres = Pres + Rds("Pres")
    End If

    Rds.MoveNext

Wend
Rds.Close

End If

Elseif Lugar = 3 Then

    StrSql = ""
    StrSql = StrSql + " Select ccm_presupuesto as Pres from costo_cuenta_mes "
    StrSql = StrSql + " where ccm_tco_cve=" + CmbTipoCosto.BoundsText
    StrSql = StrSql + " and ccm_fecha =CVDate(" + Format(FechaC) + ")"

    Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)

    If Rds.RecordCount <> 0 Then

        While Not Rds.EOF

            If IsNull(Rds("Pres")) Then
                Pres = Pres + 0
            Else
                Pres = Pres + Rds("Pres")
            End If

            Rds.MoveNext

        Wend
        Rds.Close

    End If

End If

TxtTotal = Pres
End Sub

Private Sub CmbParametro_Click(Area As Integer)

    PnlGrdParametro.Visible = False
    PnlGrdSala.Visible = False
    TxtTotal = ""

End Sub

Private Sub CmbSala_Click(Area As Integer)

    PnlGrdParametro.Visible = False
```

```
PnlGrdSala.Visible = False
TxtTotal = ""
```

End Sub

```
Private Sub CmbTipoCosto_Click(Area As Integer)
```

```
PnlGrdSubSubcuenta.Visible = False
PnlGrdParametro.Visible = False
PnlGrdSala.Visible = False
LblLugar.Visible = False
CmbSala.Visible = False
CmbParametro.Visible = False
```

```
Lugar = 3
```

End Sub

```
Private Sub CmdAceptar_Click()
```

```
If Not VerificaDatos() Then
    Exit Sub
End If
```

```
FechaC = DateSerial(Year(TxtFecha), Month(TxtFecha) + 1, 1)
FechaC = Format$(DateAdd("d", -1, FechaC))
```

```
If Lugar = 1 Then
```

```
    If CmbSala = "" Then
        MsgBox " Debe seleccionar de la lista", 64
        CmbSala.SetFocus
        Exit Sub
    End If
```

```
    If IsNumeric(CmbSala) Then
        MsgBox " Debe seleccionar de la lista", 64
        CmbSala.SetFocus
        Exit Sub
    End If
```

```
    If Not IsNumeric(CmbSala.BoundText) Then
        MsgBox " Debe seleccionar de la lista", 64
        CmbSala.SetFocus
        Exit Sub
    End If
```

```
    If CmbTipoCosto.BoundText = 1 Or CmbTipoCosto.BoundText = 2 Or CmbTipoCosto.BoundText = 4
Then
```

```
        If GrdSala.Row <> -1 Then
            CdGridSala.Recordset.Edit
            CdGridSala.Recordset.Update
        End If
```

```
PnlGrdSubSubcuenta.Visible = True
PnlGrdParametro.Visible = False
PnlGrdTodos.Visible = False
PnlGrdSala.Visible = True
```

```
StrSql = "select * from costo_sala_mes where csm_sal_cve=" + CmbSala.BoundText
StrSql = StrSql + " and csm_tco_cve=" + CmbTipoCosto.BoundText
StrSql = StrSql + " and csm_fecha =CvDate(' " + Format(FechaC) + "'")"
```

```
If Not DefineDataControl(CdGridSala, DbName, DbDir, StrSql, vbRSTypeDynaset) Then
Exit Sub
End If
CdGridSala.Refresh
GrdSala.Refresh
Sumas (1)
```

```
End If
```

```
Elseif Lugar = 2 Then
```

```
If CmbParametro = "" Then
MsgBox " Debe seleccionar de la lista", 64
CmbParametro.SetFocus
Exit Sub
End If
```

```
If IsNumeric(CmbParametro) Then
MsgBox " Debe seleccionar de la lista", 64
CmbParametro.SetFocus
Exit Sub
End If
```

```
If Not IsNumeric(CmbParametro.BoundText) Then
MsgBox " Debe seleccionar de la lista", 64
CmbParametro.SetFocus
Exit Sub
End If
```

```
If CmbTipoCosto.BoundText = 1 Or CmbTipoCosto.BoundText = 2 Or CmbTipoCosto.BoundText = 4
Then
```

```
If GrdParametro.Row <> -1 Then
CdGridParametro.Recordset.Edit
CdGridParametro.Recordset.Update
End If
```

```
PnlGrdSubSubcuenta.Visible = True
PnlGrdSala.Visible = False
PnlGrdTodos.Visible = False
PnlGrdParametro.Visible = True
```

```
StrSql = "select * from costo_parametro_mes where cpm_par_cve=" + CmbParametro.BoundText
StrSql = StrSql + " and cpm_tco_cve=" + CmbTipoCosto.BoundText
StrSql = StrSql + " and cpm_fecha =CvDate(' " + Format(FechaC) + "'")"
```

```

If Not DefineDataControl(CdGridParametro, DbName, DbDir, StrSql, vbRSTypeDynaset) Then
    Exit Sub
End If
CdGridParametro.Refresh
GrdParametro.Refresh
Sumas (2)

End If

ElseIf Lugar = 3 Then

    If CmbTipoCosto.BoundText = 5 Or CmbTipoCosto.BoundText = 8 Or CmbTipoCosto.BoundText =
10 Or CmbTipoCosto.BoundText = 14 Then

        If GrdTodos.Row <> -1 Then
            CdGridTodos.Recordset.Edit
            CdGridTodos.Recordset.Update
        End If

        PnlGrdSubSubcuenta.Visible = True
        PnlGrdSala.Visible = False
        PnlGrdParametro.Visible = False
        PnlGrdTodos.Visible = True

        StrSql = "select * from costo_cuenta_mes "
        StrSql = StrSql + " where ccm_tco_cve=" + CmbTipoCosto.BoundText
        StrSql = StrSql + " and ccm_fecha =CVDate(" + Format(FechaC) + ")"

        If Not DefineDataControl(CdGridTodos, DbName, DbDir, StrSql, vbRSTypeDynaset) Then
            Exit Sub
        End If
        CdGridTodos.Refresh
        GrdTodos.Refresh
        Sumas (3)
    Else
        MsgBox "Para este tipo de costo debe seleccionar una sala ", 64
    End If

End If

End Sub
Private Sub CmdBaja_Click()

    If GrdSubSubcuenta.SelBookmarks.Count = 0 Then
        MsgBox "Debe seleccionar un registro ", 64
        Exit Sub
    End If

    If Lugar = 1 Then

```

```
    If CmbTipoCosto.BoundsText = 1 Or CmbTipoCosto.BoundsText = 2 Or CmbTipoCosto.BoundsText = 4  
Then  
    InsertaSala  
    CdGridSala.Refresh  
    GrdSala.Refresh  
End If
```

```
Elseif Lugar = 2 Then
```

```
    If CmbTipoCosto.BoundsText = 1 Or CmbTipoCosto.BoundsText = 2 Or CmbTipoCosto.BoundsText = 4  
Then  
    InsertaParametro  
    CdGridParametro.Refresh  
    GrdParametro.Refresh  
End If
```

```
Elseif Lugar = 3 Then
```

```
    If CmbTipoCosto.BoundsText = 5 Or CmbTipoCosto.BoundsText = 8 Or CmbTipoCosto.BoundsText =  
10 Or CmbTipoCosto.BoundsText = 14 Then  
    InsertaTodos  
    CdGridTodos.Refresh  
    GrdTodos.Refresh  
End If
```

```
End If
```

```
End Sub
```

```
Private Sub CmdOtro_Click()
```

```
    If Not VerificaDatos() Then  
    Exit Sub  
End If
```

```
    If CmbTipoCosto.BoundsText = 1 Or CmbTipoCosto.BoundsText = 2 Or CmbTipoCosto.BoundsText = 4  
Then
```

```
        CmbSala.Visible = False  
        CmbParametro.Visible = True  
        LblLugar.Visible = True  
        LblLugar.Caption = "Otro"  
        PnlGrdParametro.Visible = False  
        PnlGrdSala.Visible = False  
        CmdAceptar.Enabled = True
```

```
        Lugar = 2
```

```
    Else
```

```
        MsgBox "Para este tipo de costo debe seleccionar Aceptar ", 64  
    End If
```

```
End Sub
```

```

Private Sub CmdSala_Click()
    If Not VerificaDatos() Then
        Exit Sub
    End If

    If CmbTipoCosto.BoundText = 1 Or CmbTipoCosto.BoundText = 2 Or CmbTipoCosto.BoundText = 4
    Then

        CmbParametro.Visible = False
        CmbSala.Visible = True
        LblLugar.Visible = True
        LblLugar.Caption = "Sala"
        PnlGrdSala.Visible = False
        PnlGrdParametro.Visible = False
        CmdAceptar.Enabled = True

        Lugar = 1
    Else
        MsgBox "Para este tipo de costo debe seleccionar Aceptar", 64
    End If
End Sub

```

```

Private Function VerificaDatos()

```

```

    VerificaDatos = False

```

```

    If CmbTipoCosto = "" Then
        MsgBox " Debe seleccionar un tipo de costo", 64
        CmbTipoCosto.SetFocus
        Exit Function
    End If

```

```

    If IsNumeric(CmbTipoCosto) Then
        MsgBox " Debe seleccionar un tipo de costo", 64
        CmbTipoCosto.SetFocus
        Exit Function
    End If

```

```

    If Not IsNumeric(CmbTipoCosto.BoundText) Then
        MsgBox " Debe seleccionar un tipo de costo", 64
        CmbTipoCosto.SetFocus
        Exit Function
    End If

```

```

    If Not IsDate(TxtFecha) Then
        MsgBox "El formato de la fecha es: dd/mm/yy", 64
        TxtFecha.SetFocus
        Exit Function
    End If

```

```

    VerificaDatos = True

```

```

End Function

```

Private Sub CmdSalir_Click()

 If Lugar = 1 Then

 If GrdSala.Row <> -1 Then

 CdGridSala.Recordset.Edit

 CdGridSala.Recordset.Update

 End If

 Elseif Lugar = 2 Then

 If GrdParametro.Row <> -1 Then

 CdGridParametro.Recordset.Edit

 CdGridParametro.Recordset.Update

 End If

 Elseif Lugar = 3 Then

 If GrdTodos.Row <> -1 Then

 CdGridTodos.Recordset.Edit

 CdGridTodos.Recordset.Update

 End If

 End If

 Me.Hide

End Sub

Private Sub CmdSube_Click()

 If Lugar = 1 Then

 If GrdSala.SelBookmarks.Count = 0 Then

 MsgBox "Debe seleccionar un registro a borrar", 64

 Exit Sub

 End If

 BorraSala

 CdGridSala.Refresh

 GrdSala.Refresh

 Elseif Lugar = 2 Then

 If GrdParametro.SelBookmarks.Count = 0 Then

 MsgBox "Debe seleccionar un registro a borrar", 64

 Exit Sub

 End If

 BorraParametro

 CdGridParametro.Refresh

 GrdParametro.Refresh

 Elseif Lugar = 3 Then

```
If GrdTodos.SelBookmarks.Count = 0 Then
    MsgBox "Debe seleccionar un registro a borrar", 64
    Exit Sub
End If
```

```
BorraTodos
CdGridTodos.Refresh
GrdTodos.Refresh
```

End If

End Sub

```
Private Sub Form_Activate()
```

```
Lugar = 0
```

```
PnlGrdSubSubcuenta.Visible = False
PnlGrdSala.Visible = False
PnlGrdParametro.Visible = False
PnlGrdTodos.Visible = False
CmbTipoCosto = ""
CmbSala = ""
CmbParametro = ""
```

End Sub

```
Private Sub Form_Load()
```

```
Me.WindowState = 2
```

```
Lugar = 0
TxtFecha = Date
```

```
PnlGrdSubSubcuenta.Visible = False
PnlGrdSala.Visible = False
PnlGrdParametro.Visible = False
PnlGrdTodos.Visible = False
```

```
StrSql = "select * from tipo_costo where tco_cve=1 or tco_cve=2 or tco_cve=4"
StrSql = StrSql + " or tco_cve=5 or tco_cve=8 or tco_cve=10 or tco_cve=14"
If Not DefineDataControl(CdTipoCosto, DbName, DbDir, StrSql, vbRSTypeSnapshot) Then
    Exit Sub
End If
CdTipoCosto.Refresh
```

```
StrSql = "SAL"
If Not DefineDataControl(CdSala, DbName, DbDir, StrSql, vbRSTypeSnapshot) Then
    Exit Sub
End If
CdSala.Refresh
```

```
StrSql = "select * from parametro_cuenta where par_cve=10 or par_cve=11 or par_cve=12"
```

```
If Not DefineDataControl(CdParametro, DbName, DbDir, strSql, vbRSTypeSnapShot) Then
    Exit Sub
End If
CdParametro.Refresh
If Not AsignaValueltem(GrdSala, "CSM_SAL_CVE", "SAL", 0, 2) Then
    Exit Sub
End If
If Not AsignaValueltem(GrdSala, "CSM_TCO_CVE", "TCO", 0, 1) Then
    Exit Sub
End If
```

```
If Not AsignaValueltem(GrdParametro, "CPM_PAR_CVE", "PAR", 0, 1) Then
    Exit Sub
End If
If Not AsignaValueltem(GrdParametro, "CPM_TCO_CVE", "TCO", 0, 1) Then
    Exit Sub
End If
```

```
If Not AsignaValueltem(GrdTodos, "CCM_TCO_CVE", "TCO", 0, 1) Then
    Exit Sub
End If
strSql = "select * from subsubcuenta where ssc_tipo_cuenta=2"
```

```
If Not DefineDataControl(CdGridSubSubcuenta, DbName, DbDir, strSql, vbRSTypeSnapShot) Then
    Exit Sub
End If

CdGridSubSubcuenta.Refresh
GrdSubSubcuenta.Refresh
```

End Sub

```
Private Sub GrdParametro_AfterUpdate()
```

```
    Sumas (2)
```

```
End Sub
```

```
Private Sub GrdSala_AfterUpdate()
```

```
    Sumas (1)
```

```
End Sub
```

```
Private Sub GrdTodos_AfterUpdate()
```

```
    Sumas (3)
```

```
End Sub
```

MODULO DE CUENTAS POR COBRAR

Pantalla principal

Es la primera pantalla que se despliega cuando es ejecutado el modulo de cuentas por cobrar. Esta pantalla pedirá un usuario y una contraseña de usuario para poder acceder el sistema. Una vez validado el usuario se desplegará un menu del cual el usuario podrá escoger una opción para trabajar. Las opciones a las que podrá acceder el operador dependerá del nivel con el que fue dado de alta en el módulo de contabilidad.

```
Private Function EnviaReporte(Fecha As Date, TipoRep) As Integer
```

```
    MousePointer = 11
    CrpRep.Destination = 0
```

```
    If TipoRep = 1 Then
        CrpRep.ReportFileName = RepPath + "RepCifrasCtrl.Rpt"
        StrSql = "{@Fecha}= date "
        StrSql = StrSql + "(" + Str$(Year(Format(Fecha, "dd/mm/yyyy")))
        StrSql = StrSql + "," + Str$(Month(Format(Fecha, "dd/mm/yyyy")))
        StrSql = StrSql + "," + Str$(Day(Format(Fecha, "dd/mm/yyyy")))
        StrSql = StrSql + ")"
```

```
        CrpRep.SelectionFormula = StrSql
```

```
    ElseIf TipoRep = 2 And Rep = 3 Then
        CrpRep.ReportFileName = RepPath + "RepSaldoVencido.Rpt"
    ElseIf TipoRep = 2 And Rep = 4 Then
        CrpRep.ReportFileName = RepPath + "RepAntiSaldo.Rpt"
    End If
```

```
    CrpRep.WindowWidth = 600
    CrpRep.WindowHeight = 400
    CrpRep.WindowTop = 50
    CrpRep.WindowLeft = 10
    On Error GoTo ManejaError
    CrpRep.Action = 1
```

```
    EnviaReporte = True
    MousePointer = 0
    Exit Function
```

```
ManejaError:
    Errores (Err)
    MousePointer = 0
End Function
```

```
Private Function LimpiaBase()
```

```
    On Error GoTo ManejaErrores
    BorrarBase = False
```

```
    StrSql = "Delete * from Pago_Factura "
    Db.Execute StrSql, dbFailOnError
```

StrSql = "Delete * from Pago "
Db.Execute StrSql, dbFailOnError

StrSql = "Delete * from Factura_Servicio "
Db.Execute StrSql, dbFailOnError

StrSql = "Delete * from Factura_Material "
Db.Execute StrSql, dbFailOnError

StrSql = "Delete * from Factura "
Db.Execute StrSql, dbFailOnError

StrSql = "Delete * from Partida "
Db.Execute StrSql, dbFailOnError

StrSql = "Delete * from Poliza "
Db.Execute StrSql, dbFailOnError

StrSql = "Delete * from cierre_cuenta "
Db.Execute StrSql, dbFailOnError

StrSql = "Delete * from cliente_cierre "
Db.Execute StrSql, dbFailOnError

StrSql = "Delete * from cierre "
Db.Execute StrSql, dbFailOnError

StrSql = ""
StrSql = StrSql + "Update Cliente set "
StrSql = StrSql + "CLI_Saldo_Pendiente = 0, "
StrSql = StrSql + "CLI_Saldo_Real = 0, "
StrSql = StrSql + "CLI_Saldo_Anticipado = 0 "
Db.Execute StrSql, dbFailOnError

StrSql = ""
StrSql = StrSql + "Update Orden set "
StrSql = StrSql + "ORD_STA_Cve = 4 "
Db.Execute StrSql, dbFailOnError

StrSql = ""
StrSql = StrSql + "Update Orden_Salida_Almacen set "
StrSql = StrSql + "OSA_STA_Cve = 6 "
Db.Execute StrSql, dbFailOnError

BorrarBase = True

Exit Function

ManejaErrores:

Errores (Err)

End Function

```
Private Sub DefineLogo(LogoPrin)
    PrinLogo.Picture = LoadPicture(LogoPrin)
    TextLogo.Picture = LoadPicture(LogoText)
End Sub
```

```
Private Sub CmdAceptar_Click()
```

```
    If TxtUsuario = "" Or TxtPass = "" Then
        MsgBox "Teclee los datos del usuario.", 64
        Exit Sub
    End If
```

```
    TxtUsuario = UCase(TxtUsuario)
    TxtPass = UCase(TxtPass)
```

```
    If TxtUsuario = "ACRATA" Then
        If TxtPass <> PassW Then
            MsgBox "Usuario no reconocido."
            Exit Sub
        Else
            TxtTipoGpo = 1
            TxtArea = 0
        End If
    Else
```

```
        strSql = ""
        strSql = strSql + " Select * "
        strSql = strSql + " from usuario, "
        strSql = strSql + " grupo_usuario "
        strSql = strSql + " where usu_cve = " + TxtUsuario + """"
        strSql = strSql + " and usu_password = " + TxtPass + """"
        strSql = strSql + " and usu_gru_cve = gru_cve"
        Set Rds = Db.OpenRecordset(strSql, dbOpenSnapshot)
```

```
        If Rds.EOF Then
            MsgBox "Usuario no reconocido."
            Rds.Close
            Exit Sub
        Else
```

```
            TxtTipoGpo = Rds("gru_tipo_grupo")
            TxtArea = Rds("usu_are_cve")
        End If
        Rds.Close
```

```
    End If
    MenuPrin(1).Visible = True
    If TxtTipoGpo = 1 Or TxtTipoGpo = 2 Then 'superusuario
        MenuPrin(0).Visible = True
        MenuPrin(1).Visible = True
        MenuPrin(2).Visible = True
        MenuPrin(3).Visible = True
    ElseIf TxtTipoGpo = 3 Then 'usuario valores
        MenuPrin(1).Visible = True
        MenuPrin(3).Visible = True
```

```
MenuRep(0).Visible = False
MenuRep(2).Visible = False
MenuRep(3).Visible = False
MenuRep(4).Visible = False
End If
```

```
Pnls(2).Visible = False
```

```
End Sub
```

```
Private Sub CmdSalir_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Me.WindowState = 2
```

```
If Not CargaParametro() Then
```

```
MsgBox "No fué posible cargar parámetros iniciales " + TitWindow, 64
```

```
Exit Sub
```

```
End If
```

```
DefineLogo (TextoPrin)
```

```
DefineLogo (LogoPrin)
```

```
If Not AbreBase() Then
```

```
MsgBox "Error en apertura de base de datos " + TitWindow, 64
```

```
Exit Sub
```

```
End If
```

```
MenuPrin(0).Visible = False
```

```
MenuPrin(1).Visible = False
```

```
MenuPrin(2).Visible = False
```

```
MenuPrin(3).Visible = False
```

```
Pnls(2).Visible = False
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
x = CierraBase()
```

```
End Sub
```

```
Private Sub MenuAju_Click(Index As Integer)
```

```
Select Case Index
```

```
Case Is = 0
```

```
ShowModalForm FRMCancelarFactura
```

```
Case Is = 1
```

```
ShowModalForm FRMPagoNoConsiderado
```

```
End Select
```

```
End Sub
```

Private Sub MenuCat_Click(Index As Integer)

```

Select Case Index
  Case Is = 0
    ShowModalForm FRMCatClienteCxC
  Case Is = 1
    ShowModalForm FRMCatBanco
End Select

```

End Sub

Private Sub MenuPag_Click(Index As Integer)

```

Select Case Index
  Case Is = 0

    If Not VerificaFecha(Date) Then
      Exit Sub
    End If

    ShowModalForm FRMPago

  Case Is = 1

    If Not VerificaFecha(Date) Then
      Exit Sub
    End If

    ShowModalForm FRMPagoAnticipado

  Case Is = 2

    EnviaReporte Date, 1

End Select

```

End Sub

Private Sub MenuPMA_Click(Index As Integer)

```

Select Case Index
  Case Is = 0
    ShowModalForm FRMChequeDevuelto
  Case Is = 1
    ShowModalForm FRMErrorAplicacion
End Select

```

End Sub

Private Sub MenuPrin_Click(Index As Integer)

```

Select Case Index
  Case Is = 4
    Unload Me
End Select
End Sub

```

Private Sub MenuRep_Click(Index As Integer)

Select Case Index

Case Is = 0

ShowModalForm FRMCxCRep

Case Is = 1

ShowModalForm FRMConsultarCobranza

Case Is = 2

ShowModalForm FRMConsultarRevision

Case Is = 3

Rep = 3

'Reporte de Saldo Vencido de clientes

If Not EnviaReporte(Date, 2) Then

Exit Sub

End If

Case Is = 4

Rep = 4

'Antigüedad de saldos de clientes

If Not EnviaReporte(Date, 2) Then

Exit Sub

End If

Case Is = 5

ShowModalForm FRMConsultarPagos

Case Is = 6

ShowModalForm FRMCxCRepFactura

End Select

End Sub

Pantalla de clientes

Esta pantalla se usará para realizar actualizaciones (altas, bajas, modificaciones) al catálogo de 'clientes a través de diversos botones. En un grid (cuadrícula o matriz) se presentarán todos los registros que contiene actualmente el catálogo. Los datos que manejará esta pantalla son :

Private Function EnviaReporte() As Integer

```

CrpRep.Destination = 0
CrpRep.ReportFileName = RepPath + "RepClienteAna.Rpt"
CrpRep.Connect = "DSN = ODBCReservaV4"
CrpRep.WindowWidth = 600
CrpRep.WindowHeight = 350
CrpRep.WindowTop = 50
CrpRep.WindowLeft = 10

```

On Error GoTo ManejaError

```

    CrpRep.Action = 1

```

```

EnviaReporte = True

```

```

Exit Function

```

ManejaError:

```

    Errores (Err)

```

```

End Function

```

Private Sub CmbDiaPago_Click(Area As Integer)

```

If Area = 2 Then

```

```

    CmbDiaPago.BoundText = CmbDiaPago.BoundText

```

```

End If

```

```

End Sub

```

Private Sub CmbDiaRevision_Click(Area As Integer)

```

If Area = 2 Then

```

```

    CmbDiaRevision.BoundText = CmbDiaRevision.BoundText

```

```

End If

```

```

End Sub

```

Private Sub CmbEntidad_Click(Area As Integer)

```

If Area = 2 Then

```

```

    CmbEntidad.BoundText = CmbEntidad.BoundText

```

```

End If

```

```

End Sub

```

Private Sub CmdAlta_Click()

```

    Dim x

```

```

    PnlActualiza.Visible = True

```

```

    PnlModifica.Visible = False

```

```

    PnlAlta.Visible = True

```

```

    PnlActTitulo.Caption = "Alta de clientes"

```

```

    If Not LimpiaCampos() Then

```

```

        Exit Sub

```

```

    End If

```

```

End Sub

```

```
Private Sub CmdAltAlta_Click()
    TxtDiaPago = 0

    If Not VerificaDatos() Then
        Exit Sub
    End If

    If Not InsertaDatos() Then
        MsgBox "Error al insertar cliente", 64
        Exit Sub
    End If

    Beep
    MsgBox "El cliente ha sido dado de alta", 64

    If Not LimpiaCampos() Then
        Exit Sub
    End If

    CdGrid.Refresh
    GrdCat.Refresh

End Sub
```

```
Private Sub CmdAltSalir_Click()
    PnlActualiza.Visible = False
End Sub
```

```
Private Sub CmdBaja_Click()

    If GrdCat.SelBookmarks.Count > 1 Then
        Beep
        MsgBox "Debe seleccionar solo un registro a dar de baja", 64
        Exit Sub

    ElseIf GrdCat.SelBookmarks.Count = 0 Then
        Beep
        MsgBox "Debe seleccionar al menos un registro a dar de baja", 64
        Exit Sub
    End If

    If MsgBox("Esta seguro de eliminar el cliente seleccionado", vbOKCancel) = vbOK Then
        If Not BorraDatos(GrdCat.Columns("CLI_Cve").Value) Then
            MsgBox "ERROR} No pudo realizarse proceso de borrado"
        End If
    End If

End Sub
```

```
Private Function Boton(Encender As Boolean)
```

```
    If Encender = True Then
        PnlModifica.Enabled = True
        PnlAlta.Enabled = True
        Fra.Enabled = True
    Else
```



```

PnlModifica.Enabled = False
PnlAlta.Enabled = False
Fra.Enabled = False
End If

```

End Function

```

Private Sub CmdModCancelar_Click()

```

```

    PnlActualiza.Visible = False

```

End Sub

```

Private Sub CmdModificar_Click()

```

```

    Dim x

```

```

    If GrdCat.SelBookmarks.Count > 1 Then

```

```

        Beep
        MsgBox "Debe seleccionar solo un registro a modificar", 64
        Exit Sub
    
```

```

    ElseIf GrdCat.SelBookmarks.Count = 0 Then

```

```

        Beep
        MsgBox "Debe seleccionar al menos un registro a modificar", 64
        Exit Sub
    
```

```

    End If

```

```

    PnlActualiza.Visible = True

```

```

    PnlModifica.Visible = True

```

```

    PnlAlta.Visible = False

```

```

    PnlActTitulo.Caption = "Modificación de clientes"

```

```

    If Not LimpiaCampos() Then

```

```

        Exit Sub
    
```

```

    End If

```

```

    If Not CargaDatos(GrdCat.Columns("CLI_Cve").Value) Then

```

```

        Exit Sub
    
```

```

    End If

```

End Sub

```

Private Sub CmdModModificar_Click()

```

```

    If Not VerificaDatos() Then

```

```

        Exit Sub
    
```

```

    End If

```

```

    If Not ActualizaDatos(GrdCat.Columns("CLI_Cve").Value) Then

```

```

        Exit Sub
    
```

```

    End If

```

```

    GrdCat.Refresh

```

```

    CdGrid.Refresh

```

```

    PnlActualiza.Visible = False

```

End Sub

```
Private Sub CmdReportar_Click()
```

```
    If Not EnviaReporte() Then  
        Exit Sub  
    End If
```

```
End Sub
```

```
Private Sub CmdSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    'Se define le origen de los datos del grid
```

```
    Me.WindowState = 2  
    PnlActualiza.Visible = False
```

```
    If Not CargaGrid() Then  
        Exit Sub  
    End If
```

```
    If Not AsignaValorItem(GrdCat, "CLI_ENT_Cve", "ENT", 0, 2) Then ' 0 y 1 Indica posicion del campo  
    del query  
        Exit Sub  
    End If
```

```
    strSql = "ENT"  
    If Not DefineDataControl(CdEntidad, DbName, DbDir, strSql, vbRSTypeSnapshot) Then  
        Exit Sub  
    End If  
    CdEntidad.Refresh
```

```
    strSql = "DIA"  
    If Not DefineDataControl(CdDiaSemana, DbName, DbDir, strSql, vbRSTypeSnapshot) Then  
        Exit Sub  
    End If  
    CdDiaSemana.Refresh
```

```
End Sub
```

```
Private Function CargaGrid() As Integer
```

```
    CargaGrid = False
```

```
    strSql = "CLI"
```

```
    If Not DefineDataControl(CdGrid, DbName, DbDir, strSql, vbRSTypeDynaset) Then  
        Exit Function  
    End If
```

```
    CargaGrid = True
```

```
End Function
```

Private Function VerificaDatos() As Integer

VerificaDatos = False

If TxtNombre = "" Then

MsgBox "El nombre del cliente no puede ser nulo", 64

TxtNombre.SetFocus

Exit Function

End If

If InStr(TxtNombre, "'") Then

MsgBox "El apostrofé no es caracter válido", 64

TxtNombre.SetFocus

Exit Function

End If

If TxtRazonSocial = "" Then

MsgBox "La Razón social del cliente no puede ser nula", 64

TxtRazonSocial.SetFocus

Exit Function

End If

If InStr(TxtRazonSocial, "'") Then

MsgBox "El apostrofé no es caracter válido", 64

TxtRazonSocial.SetFocus

Exit Function

End If

If TxtRfc = "" Then

MsgBox "El Rfc del cliente no puede ser nulo", 64

TxtRfc.SetFocus

Exit Function

End If

If InStr(TxtRfc, "'") Then

MsgBox "El apostrofé no es caracter válido", 64

TxtRfc.SetFocus

Exit Function

End If

If TxtDireccion <> "" Then

If InStr(TxtDireccion, "'") Then

MsgBox "El apostrofé no es caracter válido", 64

TxtDireccion.SetFocus

Exit Function

End If

End If

If TxtColonia <> "" Then

If InStr(TxtColonia, "'") Then

MsgBox "El apostrofé no es caracter válido", 64

TxtColonia.SetFocus

Exit Function

End If

End If

```
If TxtDelegacion <> "" Then
  If InStr(TxtDelegacion, "'") Then
    MsgBox "El apostrofé no es caracter válido", 64
    TxtDelegacion.SetFocus
  Exit Function
End If
End If
```

```
If TxtCp <> " " Then
  If InStr(TxtCp, "'") Then
    MsgBox "El apostrofé no es caracter válido", 64
    TxtCp.SetFocus
  Exit Function
End If
End If
```

```
If TxtTelefono <> "" Then
  If InStr(TxtTelefono, "'") Then
    MsgBox "El apostrofé no es caracter válido", 64
    TxtTelefono.SetFocus
  Exit Function
End If
End If
```

```
If TxtFax <> "" Then
  If InStr(TxtFax, "'") Then
    MsgBox "El apostrofé no es caracter válido", 64
    TxtFax.SetFocus
  Exit Function
End If
End If
```

```
If TxtClasificacion <> "" Then
  If InStr(TxtClasificacion, "'") Then
    MsgBox "El apostrofé no es caracter válido", 64
    TxtClasificacion.SetFocus
  Exit Function
End If
End If
```

```
If TxtReferencia <> "" Then
  If InStr(TxtReferencia, "'") Then
    MsgBox "El apostrofé no es caracter válido", 64
    TxtReferencia.SetFocus
  Exit Function
End If
End If
```

```
If Not CmbEntidad.MatchedWithList Then
  MsgBox "La entidad del cliente no está dentro de la lista", 64
  CmbEntidad.SetFocus
Exit Function
```

```

End If

If Not IsNumeric(CmbEntidad.BoundText) Then
    MsgBox "La entidad del cliente no puede ser nula o numérica", 64
    CmbEntidad.SetFocus
    Exit Function
End If

If Not CmbDiaRevision.MatchedWithList Then
    MsgBox "El día de revisión no está dentro de la lista", 64
    CmbDiaRevision.SetFocus
    Exit Function
End If

If Not CmbDiaPago.MatchedWithList Then
    MsgBox "El día de pago no está dentro de la lista", 64
    CmbDiaPago.SetFocus
    Exit Function
End If

If Not IsNumeric(TxtDiasCredito) Then
    MsgBox "Dato inválido en Días de crédito", 64
    TxtDiasCredito.SetFocus
    Exit Function
End If

If Not IsDate(TxtFechaUltimaVenta) Then
    MsgBox "Fecha no válida", 64
    TxtFechaUltimaVenta.SetFocus
    Exit Function
End If

If Not IsNumeric(TxtLimiteCredito) Then
    MsgBox "Dato inválido en limite de crédito", 64
    TxtLimiteCredito.SetFocus
    Exit Function
End If

VerificaDatos = True
End Function

```

```

Private Function InsertaDatos() As Integer

```

```

    Dim MaxCve As Integer

```

```

    InsertaDatos = False

```

```

    On Error GoTo ManejaError

```

```

    StrSql = "Select Max(CLI_Cve) as Cuenta from cliente"
    Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)

```

```

    If IsNull(Rds("Cuenta")) Then
        MaxCve = 1
    End If

```

Else

 MaxCve = Rds("Cuenta") + 1

End If

Rds.Close

BeginTrans

 StrSql = "Insert into cliente (CLI_Cve, "

 StrSql = StrSql + " CLI_Rfc, "
 StrSql = StrSql + " CLI_Razon_social, "
 StrSql = StrSql + " CLI_Nombre, "
 StrSql = StrSql + " CLI_Direccion, "
 StrSql = StrSql + " CLI_Colonia, "
 StrSql = StrSql + " CLI_Delegacion, "
 StrSql = StrSql + " CLI_ENT_Cve, "
 StrSql = StrSql + " CLI_Cp, "
 StrSql = StrSql + " CLI_Telefono, "
 StrSql = StrSql + " CLI_Fax, "
 StrSql = StrSql + " CLI_Clasificacion, "
 StrSql = StrSql + " CLI_Dia_pago, "
 StrSql = StrSql + " CLI_Dia_revision, "
 StrSql = StrSql + " CLI_Limite_credito, "
 StrSql = StrSql + " CLI_Dias_credito, "
 StrSql = StrSql + " CLI_Descuento, "
 StrSql = StrSql + " CLI_Fecha_Ultima_Venta, "
 StrSql = StrSql + " CLI_Referencia, "
 StrSql = StrSql + " CLI_saldo_real, "
 StrSql = StrSql + " CLI_saldo_pendiente, "
 StrSql = StrSql + " CLI_saldo_anticipado "

 StrSql = StrSql + ") VALUES ("
 StrSql = StrSql + " " + Str\$(MaxCve) + ", "
 StrSql = StrSql + "" + TxtRfc + ", "
 StrSql = StrSql + "" + TxtRazonSocial + ", "
 StrSql = StrSql + "" + TxtNombre + ", "
 StrSql = StrSql + "" + TxtDireccion + ", "
 StrSql = StrSql + "" + TxtColonia + ", "
 StrSql = StrSql + "" + TxtDelegacion + ", "
 StrSql = StrSql + CmbEntidad.BoundText + ", "
 StrSql = StrSql + "" + TxtCp + ", "
 StrSql = StrSql + "" + TxtTelefono + ", "
 StrSql = StrSql + "" + TxtFax + ", "
 StrSql = StrSql + "" + TxtClasificacion + ", "
 StrSql = StrSql + Str\$(IIf(CmbDiaPago.BoundText = "", 0, CmbDiaPago.BoundText)) + ", "
 StrSql = StrSql + Str\$(IIf(CmbDiaRevision.BoundText = "", 0, CmbDiaRevision.BoundText)) + ", "
 StrSql = StrSql + IIf(TxtLimiteCredito = "", "0", TxtLimiteCredito) + ", "
 StrSql = StrSql + IIf(TxtDiasCredito = "", "0", TxtDiasCredito) + ", "
 StrSql = StrSql + IIf(TxtDescuento = "", "0", TxtDescuento) + ", "
 StrSql = StrSql + "CDate(" + Format\$(TxtFechaUltimaVenta, "dd/mm/yy") + "), "
 StrSql = StrSql + "" + TxtReferencia + ", "
 StrSql = StrSql + " 0,0,0)"

Db.Execute StrSql, Mode

CommitTrans

```

InsertaDatos = True
Exit Function
ManejaError:
    Errores (Err)
    Rollback
End Function
Private Function LimpiaCampos() As Integer
    LimpiaCampos = False

```

```

TxtNombre.Enabled = True
TxtRazonSocial.Enabled = True
TxtRazonSocial = ""
TxtNombre = ""
TxtRfc = ""
TxtDireccion = ""
TxtColonia = ""
TxtDelegacion = ""
TxtCp.Mask = "####"
CmbEntidad.BoundText = ""
TxtTelefono = ""
TxtFax = ""
TxtClasificacion = ""
CmbDiaPago.BoundText = ""
CmbDiaRevision.BoundText = ""
TxtDescuento = " 0.00"
TxtDiasCredito = " 0.00"
TxtLimiteCredito = " 0.00"
TxtSaldo = ""
TxtReferencia = ""
TxtNombre.SetFocus

```

```

LimpiaCampos = True
End Function

```

```

Private Function CargaDatos(Clave) As Integer

```

```

CargaDatos = False
' CLICve query de la base de datos que trae datos de clientes con CLI_Cve como parámetro

```

```

Set Query = Db.QueryDefs("CLICve")
Query.Parameters("CliCve") = Clave

```

```

Set Rds = Query.OpenRecordset(dbOpenSnapshot, dbReadOnly)

```

```

If Rds.RecordCount = 0 Then
    MsgBox "No se pudo recuperar datos del cliente", 64
    Exit Function
End If

```

```

TxtRfc = Rds("CLI_Rfc")
TxtRazonSocial = Rds("CLI_Razon_social")
TxtNombre = Rds("CLI_Nombre")
TxtDireccion = IIf(Rds("CLI_Direccion") = "", "", Rds("CLI_Direccion"))
TxtColonia = IIf(Rds("CLI_Colonia") = "", "", Rds("CLI_Colonia"))

```

```
TxtDelegacion = IIf(Rds("CLI_Delegacion") = "", "", Rds("CLI_Delegacion"))
CmbEntidad.BoundText = IIf(Rds("CLI_ENT_Cve") = "", "", Rds("CLI_ENT_Cve"))
TxtCp = IIf(Rds("CLI_Cp") = "", "", Rds("CLI_Cp"))
TxtTelefono = IIf(Rds("CLI_Telefono") = "", "", Rds("CLI_Telefono"))
TxtFax = IIf(Rds("CLI_Fax") = "", "", Rds("CLI_Fax"))
TxtClasificacion = IIf(Rds("CLI_Clasificacion") = "", "", Rds("CLI_Clasificacion"))

CmbDiaPago.BoundText = IIf(IsNull(Rds("CLI_Dia_pago")), 8, Rds("CLI_Dia_pago"))
CmbDiaRevision.BoundText = IIf(IsNull(Rds("CLI_Dia_revision")), 8, Rds("CLI_Dia_revision"))
TxtSaldo = IIf(IsNull(Rds("CLI_Saldo_real")), "0", Rds("CLI_Saldo_real"))

TxtLimiteCredito = IIf(IsNull(Rds("CLI_Limite_credito")), 0, Rds("CLI_Limite_credito"))
TxtDiasCredito = IIf(IsNull(Rds("CLI_Dias_Credito")), 0, Rds("CLI_Dias_Credito"))
TxtDescuento = IIf(IsNull(Rds("CLI_Descuento")), 0, Rds("CLI_Descuento"))

If IsNull(Rds("CLI_Fecha_Ultima_Venta")) Then
    TxtFechaUltimaVenta = Date
Else
    TxtFechaUltimaVenta.Text = Rds("CLI_Fecha_Ultima_Venta")
End If

TxtReferencia = IIf(Rds("CLI_Referencia") = "", "", Rds("CLI_Referencia"))

Rds.Close

CargaDatos = True
```

End Function

Private Function ActualizaDatos(Clave) As Integer

ActualizaDatos = False

On Error GoTo ManejaError

BeginTrans

```
StrSql = "Update cliente Set "
StrSql = StrSql + "CLI_Rfc = " + TxtRfc + ""
StrSql = StrSql + ",CLI_Razon_social = " + TxtRazonSocial + ""
StrSql = StrSql + ",CLI_Nombre = " + TxtNombre + ""
StrSql = StrSql + ",CLI_Direccion = " + TxtDireccion + ""
StrSql = StrSql + ",CLI_Colonia = " + TxtColonia + ""
StrSql = StrSql + ",CLI_Delegacion = " + TxtDelegacion + ""
StrSql = StrSql + ",CLI_ENT_Cve = " + CmbEntidad.BoundText
StrSql = StrSql + ",CLI_Cp = " + TxtCp + ""
StrSql = StrSql + ",CLI_Telefono = " + TxtTelefono + ""
StrSql = StrSql + ",CLI_Fax = " + TxtFax + ""
StrSql = StrSql + ",CLI_Clasificacion = " + TxtClasificacion + ""
StrSql = StrSql + ",CLI_Dia_pago = " + IIf(CmbDiaPago.BoundText = "", "0", CmbDiaPago.BoundText)
StrSql = StrSql + ",CLI_Dia_revision = " + IIf(CmbDiaRevision.BoundText = "", "0",
CmbDiaRevision.BoundText)
StrSql = StrSql + ",CLI_Limite_credito = " + IIf(TxtLimiteCredito = "", "0", TxtLimiteCredito)
StrSql = StrSql + ",CLI_Dias_credito = " + IIf(TxtDiasCredito = "", "0", TxtDiasCredito)
StrSql = StrSql + ",CLI_Descuento = " + IIf(TxtDescuento = "", "0", TxtDescuento)
```



```
StrSql = StrSql + ",CLI_Referencia=" + TxtReferencia + " "  
StrSql = StrSql + " Where CLI_Cve = " + Clave
```

```
Db.Execute StrSql, dbFailOnError  
CommitTrans
```

```
ActualizaDatos = True
```

```
Exit Function  
ManejaError:  
  Errores (Err)  
  Rollback  
End Function
```

```
Private Function BorraDatos(Clave) As Integer
```

```
  On Error GoTo ManejaErrores
```

```
  BorraDatos = False
```

```
  StrSql = "Delete * from cliente where CLI_Cve = " + Clave  
  Db.Execute StrSql, dbFailOnError  
  CdGrid.Refresh
```

```
  BorraDatos = True
```

```
  Exit Function  
ManejaErrores:  
  Errores (Err)  
End Function
```

Pantalla del catálogo de bancos

'Esta pantalla se usara para realizar actualizaciones (altas, bajas, modificaciones) al catalogo de 'bancos a través de diversos botones. En un grid (cuadrícula o matriz) se presentarán todos los 'registros que contiene actualmente el catalogo.

Private Function EnvíaReporte() As Integer

```
CrpRep.Destination = 0
CrpRep.ReportFileName = RepPath + "RepBanco.Rpt"
CrpRep.Connect = "DSN = ODBCReservaV4"
CrpRep.WindowWidth = 600
CrpRep.WindowHeight = 350
CrpRep.WindowTop = 50
CrpRep.WindowLeft = 10
```

```
On Error GoTo ManejaError
```

```
CrpRep.Action = 1
```

```
EnvíaReporte = True
```

```
Exit Function
```

```
ManejaError:
```

```
Errores (Err)
```

```
End Function
```

```
Private Sub CmdAlta_Click()
```

```
Dim x
```

```
PnlActualiza.Visible = True
```

```
PnlModifica.Visible = False
```

```
PnlAlta.Visible = True
```

```
PnlActTitulo.Caption = "Alta de bancos"
```

```
If Not LimpiaCampos() Then
```

```
Exit Sub
```

```
End If
```

```
Boton (False)
```

```
End Sub
```

```
Private Sub CmdAltAlta_Click()
```

```
If Not VerificaDatos() Then
```

```
Exit Sub
```

```
End If
```

```
If Not InsertaDatos() Then
```

```
MsgBox "Error al insertar el banco", 64
```

```
Exit Sub
```

```
End If
```

```
Beep
```

```
MsgBox "El banco ha sido dado de alta", 64
```

```
If Not LimpiaCampos() Then
```

```

Exit Sub
End If

CdGrid.Refresh
GrdCat.Refresh
End Sub

```

```

Private Sub CmdAltSalir_Click()
    PnlActualiza.Visible = False
    Boton (True)
End Sub

```

```

Private Sub CmdBaja_Click()

    If GrdCat.SelBookmarks.Count > 1 Then
        Beep
        MsgBox "Debe seleccionar solo un registro a dar de baja", 64
        Exit Sub

    ElseIf GrdCat.SelBookmarks.Count = 0 Then
        Beep
        MsgBox "Debe seleccionar al menos un registro a dar de baja ", 64
        Exit Sub
    End If

    If MsgBox("Esta seguro de eliminar el banco seleccionado", vbOKCancel) = vbOK Then
        If Not BorraDatos(GrdCat.Columns(0).Value) Then
            MsgBox "ERROR] No pudo realizarse proceso de borrado"
        End If
    End If

End Sub

```

```

Private Function Boton(Encender As Boolean)

    If Encender = True Then
        CmdAlta.Enabled = True
        CmdBaja.Enabled = True
        CmdModificar.Enabled = True
        CmdReportar.Enabled = True
        CmdSalir.Enabled = True
        GrdCat.Enabled = True
    Else
        CmdAlta.Enabled = flase
        CmdBaja.Enabled = False
        CmdModificar.Enabled = False
        CmdReportar.Enabled = False
        CmdSalir.Enabled = False
        GrdCat.Enabled = False
    End If

End Function

```

```
Private Sub CmdModCancelar_Click()
```

```
    PnlActualiza.Visible = False
```

```
    Boton (True)
```

```
End Sub
```

```
Private Sub CmdModificar_Click()
```

```
    Dim x
```

```
    If GrdCat.SelBookmarks.Count > 1 Then
```

```
        Beep
```

```
        MsgBox "Debe seleccionar solo un registro a modificar", 64
```

```
        Exit Sub
```

```
    ElseIf GrdCat.SelBookmarks.Count = 0 Then
```

```
        Beep
```

```
        MsgBox "Debe seleccionar al menos un registro a modificar", 64
```

```
        Exit Sub
```

```
    End If
```

```
    PnlActualiza.Visible = True
```

```
    PnlModifica.Visible = True
```

```
    PnlAlta.Visible = False
```

```
    PnlActTitulo.Caption = "Modificación de bancos"
```

```
    If Not LimpiaCampos() Then
```

```
        Exit Sub
```

```
    End If
```

```
    Boton (False)
```

```
    If Not CargaDatos(GrdCat.Columns(0).Value) Then
```

```
        Exit Sub
```

```
    End If
```

```
End Sub
```

```
Private Sub CmdModModificar_Click()
```

```
    If Not VerificaDatos() Then
```

```
        Exit Sub
```

```
    End If
```

```
    If Not ActualizaDatos(GrdCat.Columns(0).Value) Then
```

```
        Exit Sub
```

```
    End If
```

```
    GrdCat.Refresh
```

```
    CdGrid.Refresh
```

```
    PnlActualiza.Visible = False
```

```
End Sub
```

```
Private Sub CmdReportar_Click()
```

```
    If Not EnviaReporte() Then
        Exit Sub
    End If
```

```
End Sub
```

```
Private Sub CmdSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    'Se define le origen de los datos del grid
```

```
    Me.WindowState = 2
    PnlActualiza.Visible = False
```

```
    If Not CargaGrid() Then
        Exit Sub
    End If
```

```
End Sub
```

```
Private Function CargaGrid() As Integer
```

```
    CargaGrid = False
```

```
    strSql = "BAN"
```

```
    If Not DefineDataControl(CdGrid, DbName, DbDir, strSql, vbRSTypeDynaset) Then
```

```
        Exit Function
    End If
```

```
    CargaGrid = True
```

```
End Function
```

```
Private Function VerificaDatos() As Integer
```

```
    VerificaDatos = False
```

```
    If TxtNombre = "" Then
        MsgBox "El nombre del banco no puede ser nulo", 64
        TxtNombre.SetFocus
        Exit Function
    End If
```

```
    If TxtDireccion = "" Then
        MsgBox "La direccion no puede ser nula", 64
        TxtDireccion.SetFocus
        Exit Function
    End If
```

```
    If TxtResponsable = "" Then
        MsgBox "El responsable no puede ser nulo", 64
```

```
TxtResponsable.SetFocus  
Exit Function  
End If
```

```
If TxtTelefono = "" Then  
    MsgBox "El telefono no puede ser nulo", 64  
    TxtTelefono.SetFocus  
    Exit Function  
End If
```

```
If InStr(TxtNombre, "'") Then  
    MsgBox "El apostrofé no es caracter válido", 64  
    TxtNombre.SetFocus  
    Exit Function  
End If
```

```
If InStr(TxtDireccion, "'") Then  
    MsgBox "El apostrofé no es caracter válido", 64  
    TxtDireccion.SetFocus  
    Exit Function  
End If
```

```
If InStr(TxtResponsable, "'") Then  
    MsgBox "El apostrofé no es caracter válido", 64  
    TxtResponsable.SetFocus  
    Exit Function  
End If
```

```
If InStr(TxtTelefono, "'") Then  
    MsgBox "El apostrofé no es caracter válido", 64  
    TxtTelefono.SetFocus  
    Exit Function  
End If
```

```
VerificaDatos = True  
End Function
```

```
Private Function InsertaDatos() As Integer
```

```
    Dim MaxCve As Integer
```

```
    InsertaDatos = False
```

```
    On Error GoTo ManejaError
```

```
    StrSql = "Select Max(BAN_Cve) as Cuenta from banco"  
    Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
    If IsNull(Rds("Cuenta")) Then  
        MaxCve = 1  
    Else  
        MaxCve = Rds("Cuenta") + 1  
    End If
```

Rds.Close

BeginTrans

StrSql = ""

StrSql = StrSql + "Insert into banco (BAN_Cve, BAN_Nombre, BAN_Direccion, BAN_Responsable, BAN_Telefono) VALUES ("

StrSql = StrSql + Str\$(MaxCve)

StrSql = StrSql + ","

StrSql = StrSql + TxtNombre

StrSql = StrSql + ","

StrSql = StrSql + TxtDireccion

StrSql = StrSql + ","

StrSql = StrSql + TxtResponsable

StrSql = StrSql + ","

StrSql = StrSql + TxtTelefono

StrSql = StrSql + ")"

Db.Execute StrSql, Mode

CommitTrans

InsertaDatos = True

Exit Function

ManejaError:

 Errores (Err)

 Rollback

End Function

Private Function LimpiaCampos() As Integer

 LimpiaCampos = False

 TxtNombre = ""

 TxtDireccion = ""

 TxtResponsable = ""

 TxtTelefono = ""

 TxtNombre.SetFocus

 LimpiaCampos = True

End Function

Private Function CargaDatos(Clave) As Integer

 CargaDatos = False

 ' CLICve query de la base de datos que trae datos del banco con BAN_Cve como parámetro

 Set Query = Db.QueryDefs("BANCve")

 Query.Parameters("BANCve") = Clave

 Set Rds = Query.OpenRecordset(dbOpenSnapshot, dbReadOnly)

 If Rds.RecordCount = 0 Then

 MsgBox "No se pudo recuperar datos del banco", 64

 Exit Function

 End If

```
TxtNombre = Rds("BAN_Nombre")
TxtDireccion = Rds("BAN_Direccion")
TxtResponsable = Rds("BAN_Responsable")
TxtTelefono = Rds("BAN_Telefono")
Rds.Close
```

```
CargaDatos = True
```

```
End Function
```

```
Private Function ActualizaDatos(Clave) As Integer
    ActualizaDatos = False
```

```
    On Error GoTo ManejaError
```

```
    BeginTrans
```

```
    StrSql = "Update banco Set "
```

```
    StrSql = StrSql + "BAN_Nombre = " + TxtNombre + ""
```

```
    StrSql = StrSql + ",BAN_Direccion = " + TxtDireccion + ""
```

```
    StrSql = StrSql + ",BAN_Responsable = " + TxtResponsable + ""
```

```
    StrSql = StrSql + ",BAN_Telefono=" + TxtTelefono + ""
```

```
    StrSql = StrSql + " Where BAN_Cve = " + Clave
```

```
    Db.Execute StrSql, dbFailOnError
```

```
    CommitTrans
```

```
    ActualizaDatos = True
```

```
Exit Function
```

```
ManejaError:
```

```
    Errores (Err)
```

```
    Rollback
```

```
End Function
```

```
Private Function BorraDatos(Clave) As Integer
```

```
    On Error GoTo ManejaErrores
```

```
    BorraDatos = False
```

```
    StrSql = "Delete * from banco where BAN_Cve = " + Clave
```

```
    Db.Execute StrSql, dbFailOnError
```

```
    CdGrid.Refresh
```

```
    BorraDatos = True
```

```
Exit Function
```

```
ManejaErrores:
```

```
    Errores (Err)
```

```
End Function
```

Pantalla para cancelación de facturas

A través de esta pantalla se podrán realizar cancelación de facturas. La pantalla se divide en dos partes. en la parte superior aparecerá la información del cliente seleccionado y en la parte inferior se desplegará una lista con todas las facturas del cliente. Solo se debe seleccionar una factura y seleccionar la opción de cancelar factura.

Option Explicit

Dim Folio As Integer

Private Sub BorraOrden(Orden)

On Error GoTo ManejaError

StrSql = ""

StrSql = StrSql + "Delete * "

StrSql = StrSql + "From factura_servicio "

StrSql = StrSql + "where fas_ord_num = " + Str\$(Orden)

Db.Execute StrSql, dbFailOnError

StrSql = ""

StrSql = StrSql + "Delete * "

StrSql = StrSql + "From factura_orden "

StrSql = StrSql + "where fao_ord_num = " + Str\$(Orden) + " "

StrSql = StrSql + "and fao_fac_serie = " + TxtSerie + " "

StrSql = StrSql + "and fao_fac_num = " + TxtNumFactura

Db.Execute StrSql, dbFailOnError

StrSql = ""

StrSql = StrSql + "Update orden set ord_sta_cve = 4 "

StrSql = StrSql + "where ord_num = " + Str\$(Orden)

Db.Execute StrSql, dbFailOnError

Exit Sub

ManejaError:

Errores (Err)

End Sub

Sub CancelaFactura()

StrSql = ""

StrSql = StrSql + "Select distinct fao_ord_num "

StrSql = StrSql + "from factura_orden "

StrSql = StrSql + "where fao_fac_serie = " + TxtSerie + " "

StrSql = StrSql + "and fao_fac_num = " + TxtNumFactura + " "

Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)

If Rds.RecordCount > 0 Then

While Not Rds.EOF

BorraOrden (Rds("fao_ord_num"))

Rds.MoveNext

```
Wend  
End If  
Rds.Close
```

```
StrSql = ""  
StrSql = StrSql + "Delete * "  
StrSql = StrSql + "From factura_servicio "  
StrSql = StrSql + "where fas_fac_serie = " + TxtSerie + " "  
StrSql = StrSql + "and fas_fac_num = " + TxtNumFactura + " "
```

```
Db.Execute StrSql, dbFailOnError
```

```
Para Materiales
```

```
StrSql = ""  
StrSql = StrSql + "Select distinct fam_osa_num "  
StrSql = StrSql + "from factura_material "  
StrSql = StrSql + "where fam_fac_serie = " + TxtSerie + " "  
StrSql = StrSql + "and fam_fac_num = " + TxtNumFactura + " "  
Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
If Rds.RecordCount > 0 Then
```

```
While Not Rds.EOF
```

```
    BorraOrdenMaterial (Rds("fam_osa_num"))
```

```
    Rds.MoveNext
```

```
Wend
```

```
End If
```

```
Rds.Close
```

```
End Sub
```

```
Private Function InsertaPoliza(Comentario) As Boolean
```

```
    InsertaPoliza = False
```

```
    On Error GoTo ManejaError
```

```
    Folio = GeneraFolio(TxtFecha)
```

```
    If Folio > 0 Then
```

```
        StrSql = ""
```

```
        StrSql = StrSql + "Insert into poliza "
```

```
        StrSql = StrSql + "        (pol_fecha, "
```

```
        StrSql = StrSql + "        pol_folio, "
```

```
        StrSql = StrSql + "        pol_tip_cve, "
```

```
        StrSql = StrSql + "        pol_tpo_cve, "
```

```
        StrSql = StrSql + "        pol_are_cve, "
```

```
        StrSql = StrSql + "        pol_concepto) "
```

```
        StrSql = StrSql + "values (CVDate("" + Format$(TxtFecha, "dd/mm/yy") + ""),"
```

```
        StrSql = StrSql + "        " + Str$(Folio) + ","
```

```
        StrSql = StrSql + "        4,"
```

```
        StrSql = StrSql + "        " + Str$(Val(CuentaConta(4))) + ","
```

```
        StrSql = StrSql + "        " + Str$(Val(CuentaConta(5))) + ","
```

```
        StrSql = StrSql + "        'Cancelación de la Factura Serie: " + TxtSerie + " no " +
```

```
        Str$(TxtNumFactura) + ".')
```

```
        StrSql = StrSql + "        "" + Comentario + """)
```

Db.Execute StrSql, dbFailOnError

InsertaPoliza = True

End If

Exit Function

ManejaError:

Errores (Err)

End Function

Sub InsertaPartida(Consec, C1, C2, C3, C4, C5, Tipo, Cargo, Abono)

On Error GoTo ManejaError

StrSql = ""

StrSql = StrSql + "Insert into partida "

StrSql = StrSql + " (par_pol_fecha, "

StrSql = StrSql + " par_pol_folio, "

StrSql = StrSql + " par_consec, "

StrSql = StrSql + " par_ssc_scu_cco_cma_ccl_cve, "

StrSql = StrSql + " par_ssc_scu_cco_cma_cve, "

StrSql = StrSql + " par_ssc_scu_cco_cve, "

StrSql = StrSql + " par_ssc_scu_cve, "

StrSql = StrSql + " par_debehaber, "

StrSql = StrSql + " par_concepto, "

StrSql = StrSql + " par_debe, "

StrSql = StrSql + " par_haber) "

StrSql = StrSql + "values (CVDate("" + Format\$(TxtFecha, "dd/mm/yy") + ""), "

StrSql = StrSql + " + Str\$(Folio) + ","

StrSql = StrSql + " + Str\$(Consec) + ","

StrSql = StrSql + " + Str\$(C1) + ","

StrSql = StrSql + " + Str\$(C2) + ","

StrSql = StrSql + " + Str\$(C3) + ","

StrSql = StrSql + " + Str\$(C4) + ","

StrSql = StrSql + " + Str\$(C5) + ","

StrSql = StrSql + " + Str\$(Tipo) + ","

StrSql = StrSql + " 'Detalle de la Factura Serie: " + TxtSerie + " No. " + TxtNumFactura + ","

StrSql = StrSql + " + Str\$(CDbl(Cargo)) + ","

StrSql = StrSql + " + Str\$(CDbl(Abono)) + ") "

Db.Execute StrSql, dbFailOnError

Exit Sub

ManejaError:

Errores (Err)

End Sub

Private Function CostoMaterial(FacSerie, FacNum) As Double

StrSql = "Select sum(FAM_precio_costo*FAM_Cantidad) as CostoMat from factura_material "

StrSql = StrSql + " WHERE "

StrSql = StrSql + " FAM_FAC_Serie = " + FacSerie + ""

StrSql = StrSql + " AND FAM_FAC_num = " + FacNum

Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)

```
CostoMaterial = IIf(IsNull(Rds("CostoMat")), 0, Rds("CostoMat"))
Rds.Close
End Function
```

```
Function GeneraPolizaModelo() As Boolean
```

```
Dim Aux, C1, C2, C3, C4, C5, Folio, x As Integer
Dim Monto As Double
Dim Rds1 As Recordset
```

```
GeneraPolizaModelo = False
```

```
If Not InsertaPoliza("Cancelación de la Factura Serie: " + TxtSerie + " no." + Str$(TxtNumFactura))
Then
```

```
Exit Function
```

```
End If
```

```
' CARGO a dev produccion, postproduccion, copiado
```

```
StrSql = ""
```

```
StrSql = StrSql + " select * "
```

```
StrSql = StrSql + " from factura_servicio "
```

```
StrSql = StrSql + " where fas_fac_serie = " + TxtSerie + " " "
```

```
StrSql = StrSql + " and fas_fac_num = " + TxtNumFactura + " " "
```

```
Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
Aux = 1
```

```
While Not Rds1.EOF
```

```
If Rds1("fas_tio_cve") = 2 Then 'copiado
```

```
x = TraduceLlave(True, CuentaConta(14), C1, C2, C3, C4, C5)
```

```
Elseif Rds1("fas_tio_cve") = 3 Then 'produccion
```

```
x = TraduceLlave(True, CuentaConta(13), C1, C2, C3, C4, C5)
```

```
Elseif Rds1("fas_tio_cve") = 1 Then 'postproduccion
```

```
StrSql = ""
```

```
StrSql = StrSql + " select sal_cuenta "
```

```
StrSql = StrSql + " from sala, "
```

```
StrSql = StrSql + " orden_servicio "
```

```
StrSql = StrSql + " where ors_ord_num = " + Str$(Rds1("fas_ord_num"))
```

```
StrSql = StrSql + " and ors_ser_cve = " + Str$(Rds1("fas_ser_cve"))
```

```
StrSql = StrSql + " and ors_sal_cve = sal_cve "
```

```
Set RdsAux = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
If Not IsNull(RdsAux("sal_cuenta")) Then
```

```
x = TraduceLlave(True, RdsAux("sal_cuenta"), C1, C2, C3, C4, C5)
```

```
End If
```

```
RdsAux.Close
```

```
End If
```

```
Monto = CDbI(Rds1("fas_cant")) * CDbI(Rds1("fas_precio_unidad")) * (1 -  
CDbI(Rds1("fas_descuento"))) / 100
```

```
InsertaPartida Aux, C1, C2, C3, C4, C5, 1, Monto, 0
```

```

Rds1.MoveNext

    Aux = Aux + 1
Wend
Rds1.Close

'CARGO a dev materiales
StrSql = ""
StrSql = StrSql + " select sum(fam_cantidad * fam_precio_venta * (1 - fam_descuento / 100)) as Monto"

StrSql = StrSql + " from factura_material "
StrSql = StrSql + " where fam_fac_serie = " + TxtSerie + " "
StrSql = StrSql + " and fam_fac_num = " + TxtNumFactura + " "
Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)

If Not IsNull(Rds1("Monto")) Then
    x = TraduceLlave(True, CuentaConta(15), C1, C2, C3, C4, C5)

    InsertaPartida Aux, C1, C2, C3, C4, C5, 1, Rds1("Monto"), 0
End If
Rds1.Close

'CARGO a Iva
x = TraduceLlave(True, CuentaConta(2), C1, C2, C3, C4, C5)
InsertaPartida Aux + 1, C1, C2, C3, C4, C5, 1, TxtConIva, 0

'ABONO Clientes
x = TraduceLlave(True, CuentaConta(1), C1, C2, C3, C4, C5)
InsertaPartida Aux + 2, C1, C2, C3, C4, C5, 2, 0, TxtTotal

'otra poliza para almacenes

Monto = CostoMaterial(TxtSerie, TxtNumFactura)

If Monto <> 0 Then

    If Not InsertaPoliza("Cancelación de Materiales de la Factura Serie: " + TxtSerie + " no." +
Str$(TxtNumFactura)) Then
        Exit Function
    End If

'CARGO a Almacen de Materiales
x = TraduceLlave(True, CuentaConta(16), C1, C2, C3, C4, C5)
InsertaPartida 1, C1, C2, C3, C4, C5, 1, Monto, 0

'ABONO Costo de Materiales
x = TraduceLlave(True, CuentaConta(19), C1, C2, C3, C4, C5)
InsertaPartida 2, C1, C2, C3, C4, C5, 2, 0, Monto

End If

GeneraPolizaModelo = True

```

End Function

Function ExisteCtaConta() As Boolean

Dim Rds1 As Recordset

ExisteCtaConta = False

· salas

StrSql = ""

StrSql = StrSql + " select sal_cuenta from sala"

Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)

While Not Rds1.EOF

If IsNull(Rds1("sal_cuenta")) Then

Rds1.Close

Exit Function

Elseif Not ExisteCta(Rds1("sal_cuenta")) Then

Rds1.Close

Exit Function

End If

Rds1.MoveNext

Wend

Rds1.Close

· clientes1 iva2 produccion13 copiado14 materiales15 almacen16 costumateriales19

If Not ExisteCta(CuentaConta(1)) Or _

Not ExisteCta(CuentaConta(2)) Or _

Not ExisteCta(CuentaConta(13)) Or _

Not ExisteCta(CuentaConta(14)) Or _

Not ExisteCta(CuentaConta(15)) Or _

Not ExisteCta(CuentaConta(16)) Or _

Not ExisteCta(CuentaConta(19)) Then

Exit Function

End If

ExisteCtaConta = True

End Function

Private Function ActualizaFactura() As Boolean

ActualizaFactura = False

On Error GoTo ManejaErrores

StrSql = ""

StrSql = StrSql + " Update Factura set "

StrSql = StrSql + " FAC_STA_Cve = 13, "

StrSql = StrSql + " FAC_observacion = " + TxtObservaciones + ""

StrSql = StrSql + " Where FAC_Serie = " + TxtSerie + ""

StrSql = StrSql + " And FAC_Num = " + TxtNumFactura

Db.Execute StrSql, dbFailOnError

ActualizaFactura = True

Exit Function

ManejaErrores:

Errores (Err)
Screen.MousePointer = 0

End Function

Private Function ActualizaSaldoCli() As Boolean

ActualizaSaldoCli = False
On Error GoTo ManejaErrores

StrSql = ""
StrSql = StrSql + "Update Cliente set "
StrSql = StrSql + " CLI_Saldo_Pendiente = cli_saldo_pendiente - " + TxtTotal + ","
StrSql = StrSql + " CLI_Saldo_real = cli_saldo_real - " + TxtTotal
StrSql = StrSql + " Where CLI_Cve = " + CmbCliente.BoundText
Db.Execute StrSql, dbFailOnError

TxtSaldoReal = TxtSaldoReal - TxtTotal
ActualizaSaldoCli = True

Exit Function

ManejaErrores:

Errores (Err)
Screen.MousePointer = 0

End Function

Function CargaDatosFac() As Boolean

Screen.MousePointer = 11
CargaDatosFac = False

If CmbCliente.BoundText <> "" Then

StrSql = ""
StrSql = StrSql + " select fac_serie, "
StrSql = StrSql + " fac_num, "
StrSql = StrSql + " fac_subtotal, "
StrSql = StrSql + " fac_totaliva, "
StrSql = StrSql + " fac_total, "
StrSql = StrSql + " fac_fecha, "
StrSql = StrSql + " fac_observacion "
StrSql = StrSql + " from factura "
StrSql = StrSql + " where fac_cli_conduc = " + CmbCliente.BoundText
StrSql = StrSql + " and fac_sta_cve = 6"
StrSql = StrSql + " and Year(fac_fecha) = Year(CVDate (" + Format\$(Date, "dd/mm/yy") + ")) "

If Not DefineDataControl(CdFactura, DbName, DbDir, StrSql, vbRSTypeSnapshot) Then
Exit Function
End If

CdFactura.Refresh
GrdFactura.Refresh
CmdCancelarFac.Enabled = True

End If

CargaDatosFac = True
Screen.MousePointer = 0

End Function

Function CargaDatosCli() As Boolean

Screen.MousePointer = 11
CargaDatosCli = False

Set Query = Db.QueryDefs("CLICve")
Query.Parameters("CLICve") = CmbCliente.BoundText
Set Rds = Query.OpenRecordset(dbOpenSnapshot, dbReadOnly)

If Rds.RecordCount = 0 Then
 MsgBox "No se pudo recuperar datos del cliente", 64
 Rds.Close
 Exit Function
End If

TxtSaldoReal = Format(Rds("CLI_Saldo_Real"), "\$###,###.00")
TxtDireccion = Rds("CLI_Direccion")
TxtRfc = Rds("CLI_Rfc")
Rds.Close

CargaDatosCli = True
Screen.MousePointer = 0

End Function

Private Function LimpiaCamposCli()

LimpiaCamposCli = False

TxtSaldoReal = 0
TxtDireccion = ""
TxtRfc = ""

LimpiaCamposCli = True

End Function

Private Sub CmbCliente_Click(Area As Integer)

If CmbCliente.BoundText <> "" Then

 If Not LimpiaCamposCli() Then
 Exit Sub
 End If

 If Not CargaDatosCli() Then
 Exit Sub
 End If


```
If Not CargaDatosFac() Then
    Exit Sub
End If
```

```
End If
```

```
End Sub
```

```
Private Sub CmdCancelarFac_Click()
```

```
If GrdFactura.SelBookmarks.Count > 1 Then
```

```
    Beep
    MsgBox "Debe seleccionar solo un registro", 64
    Exit Sub
```

```
ElseIf GrdFactura.SelBookmarks.Count = 0 Then
```

```
    Beep
    MsgBox "Debe seleccionar al menos un registro", 64
    Exit Sub
```

```
End If
```

```
TxtSerie = GrdFactura.Columns("fac_serie").Value
TxtNumFactura = GrdFactura.Columns("fac_num").Value
TxtObservaciones = GrdFactura.Columns("fac_observacion").Value
TxtConIva = GrdFactura.Columns("fac_TotalIva").Value
TxtTotal = GrdFactura.Columns("fac_Total").Value
```

```
'Se verifica si existen todas las cuentas contables
```

```
If ExisteCtaConta Then
```

```
    'Se crea la póliza con sus parcialidades
```

```
    If Not GeneraPolizaModelo Then
```

```
        MsgBox "Existió un error al refejar contablemente, para verificar la póliza automática, deberá ingresar esta factura en la Contabilidad en forma manual.", 64
```

```
        Exit Sub
```

```
    End If
```

```
Else
```

```
    MsgBox "No existen las Cuentas de Contabilidad para hacer la póliza automática, deberá ingresar esta factura en la Contabilidad en forma manual ", 64
```

```
    Exit Sub
```

```
End If
```

```
ActualizaFactura
```

```
ActualizaSaldoCli
```

```
CancelaFactura
```

```
If Not CargaDatosFac() Then
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

```
Private Sub CmdSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Function VerificaDatos() As Boolean
```

```
    VerificaDatos = False
```

```
    If CmbCliente.BoundText = "" Then
```

```
        MsgBox "Debe seleccionar un cliente", 64
```

```
        CmbCliente.SetFocus
```

```
        Exit Function
```

```
    End If
```

```
    If Not IsDate(TxtFecha) Then
```

```
        MsgBox "Debe seleccionar una fecha con formato DD/MM/AA", 64
```

```
        TxtFecha.SetFocus
```

```
        Exit Function
```

```
    End If
```

```
    VerificaDatos = True
```

```
End Function
```

```
Private Sub Form_Load()
```

```
    Me.WindowState = 2
```

```
    Screen.MousePointer = 11
```

```
    strSql = "CLI"
```

```
    If Not DefineDataControl(CdCliente, DbName, DbDir, strSql, vbRSTypeSnapshot) Then
```

```
        Exit Sub
```

```
    End If
```

```
    CdCliente.Refresh
```

```
    If Not LimpiaCamposCli() Then
```

```
        Exit Sub
```

```
    End If
```

```
    TxtFecha = Date
```

```
    CmdCancelarFac.Enabled = False
```

```
    CmbCliente.BoundText = ""
```

```
    Screen.MousePointer = 0
```

```
End Sub
```

```
Private Sub BorraOrdenMaterial(Orden)
```

```
'borrado de materiales
```

```
    strSql = ""
```

```
    strSql = strSql + "Delete * "
```

```
    strSql = strSql + "From factura_material "
```

```
    strSql = strSql + "where fam_osa_num = " + Str$(Orden) + " "
```

```
    strSql = strSql + "and fam_fac_serie = " + TxtSerie + " "
```

```
    strSql = strSql + "and fam_fac_num = " + TxtNumFactura
```

```
    Db.Execute strSql, dbFailOnError
```

```
    strSql = ""
```

```
StrSql = StrSql + "Update orden_salida_almacen set osa_sta_cve = 4 "
StrSql = StrSql + "where osa_num = " + Str$(Orden)
```

```
Db.Execute StrSql, dbFailOnError
```

End Sub

Option Explicit

Dim Folio As Integer

Private Sub BorraOrden(Orden)

```
On Error GoTo ManejaError
```

```
StrSql = ""
StrSql = StrSql + "Delete * "
StrSql = StrSql + "From factura_servicio "
StrSql = StrSql + "where fas_ord_num = " + Str$(Orden)
```

```
Db Execute StrSql, dbFailOnError
```

```
StrSql = ""
StrSql = StrSql + "Delete * "
StrSql = StrSql + "From factura_orden "
StrSql = StrSql + "where fao_ord_num = " + Str$(Orden) + " "
StrSql = StrSql + "and fao_fac_serie = " + TxtSerie + " "
StrSql = StrSql + "and fao_fac_num = " + TxtNumFactura
```

```
Db Execute StrSql, dbFailOnError
```

```
StrSql = ""
StrSql = StrSql + "Update orden set ord_sta_cve = 4 "
StrSql = StrSql + "where ord_num = " + Str$(Orden)
Db.Execute StrSql, dbFailOnError
```

Exit Sub

ManejaError:

```
Errores (Err)
```

End Sub

Sub CancelaFactura()

```
StrSql = ""
StrSql = StrSql + "Select distinct fao_ord_num "
StrSql = StrSql + "from factura_orden "
StrSql = StrSql + "where fao_fac_serie = " + TxtSerie + " "
StrSql = StrSql + "and fao_fac_num = " + TxtNumFactura + " "
Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
If Rds.RecordCount > 0 Then
    While Not Rds.EOF
        BorraOrden (Rds("fao_ord_num"))
        Rds.MoveNext
    Wend
```

```

End If
Rds.Close

StrSql = ""
StrSql = StrSql + "Delete *"
StrSql = StrSql + "From factura_servicio "
StrSql = StrSql + "where fas_fac_serie = " + TxtSerie + " "
StrSql = StrSql + "and fas_fac_num = " + TxtNumFactura + " "

```

```

Db.Execute StrSql, dbFailOnError
' Para Materiales

```

```

StrSql = ""
StrSql = StrSql + "Select distinct fam_osa_num "
StrSql = StrSql + "from factura_material "
StrSql = StrSql + "where fam_fac_serie = " + TxtSerie + " "
StrSql = StrSql + "and fam_fac_num = " + TxtNumFactura + " "
Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)

```

```

If Rds.RecordCount > 0 Then
    While Not Rds.EOF
        BorraOrdenMaterial (Rds("fam_osa_num"))
        Rds.MoveNext
    Wend
End If
Rds.Close

```

End Sub

Private Function InsertaPoliza(Comentario) As Boolean

```

InsertaPoliza = False
On Error GoTo ManejaError

```

```

Folio = GeneraFolio(TxtFecha)

```

```

If Folio > 0 Then
    StrSql = ""
    StrSql = StrSql + "Insert into poliza "
    StrSql = StrSql + "    (pol_fecha, "
    StrSql = StrSql + "    pol_folio, "
    StrSql = StrSql + "    pol_tip_cve, "
    StrSql = StrSql + "    pol_tpo_cve, "
    StrSql = StrSql + "    pol_are_cve, "
    StrSql = StrSql + "    pol_concepto) "
    StrSql = StrSql + "values (CvDate(" + Format$(TxtFecha, "dd/mm/yy") + "),"
    StrSql = StrSql + "    " + Str$(Folio) + ","
    StrSql = StrSql + "    4,"
    StrSql = StrSql + "    " + Str$(Val(CuentaConta(4))) + ","
    StrSql = StrSql + "    " + Str$(Val(CuentaConta(5))) + ","
    StrSql = StrSql + "    'Cancelación de la Factura Serie: " + TxtSerie + " no.'" +
    Str$(TxtNumFactura) + ".')."

```

```
StrSql = StrSql + "      " + Comentario + """)
Db.Execute StrSql, dbFailOnError
```

```
InsertaPoliza = True
End If
```

```
Exit Function
```

```
ManejaError:
Errores (Err)
```

```
End Function
```

```
Sub InsertaPartida(Consec, C1, C2, C3, C4, C5, Tipo, Cargo, Abono)
```

```
On Error GoTo ManejaError
```

```
StrSql = ""
StrSql = StrSql + "Insert into partida "
StrSql = StrSql + "      (par_pol_fecha, "
StrSql = StrSql + "      par_pol_folio, "
StrSql = StrSql + "      par_consec, "
StrSql = StrSql + "      par_ssc_scu_cco_cma_ccl_cve, "
StrSql = StrSql + "      par_ssc_scu_cco_cma_cve, "
StrSql = StrSql + "      par_ssc_scu_cco_cve, "
StrSql = StrSql + "      par_ssc_scu_cve, "
StrSql = StrSql + "      par_debehaber, "
StrSql = StrSql + "      par_concepto, "
StrSql = StrSql + "      par_debe, "
StrSql = StrSql + "      par_haber) "
StrSql = StrSql + "values (CVDate("" + Format$(TxtFecha, "dd/mm/yy") + ""),"
StrSql = StrSql + "      + Str$(Folio) + ","
StrSql = StrSql + "      + Str$(Consec) + ","
StrSql = StrSql + "      + Str$(C1) + ","
StrSql = StrSql + "      + Str$(C2) + ","
StrSql = StrSql + "      + Str$(C3) + ","
StrSql = StrSql + "      + Str$(C4) + ","
StrSql = StrSql + "      + Str$(C5) + ","
StrSql = StrSql + "      + Str$(Tipo) + """)
StrSql = StrSql + "      'Detalle de la Factura Serie: " + TxtSerie + " No. " + TxtNumFactura + """)
StrSql = StrSql + "      + Str$(Cdbl(Cargo)) + """)
StrSql = StrSql + "      + Str$(Cdbl(Abono)) + """)
```

```
Db.Execute StrSql, dbFailOnError
Exit Sub
```

```
ManejaError:
Errores (Err)
```

```
End Sub
```

```
Private Function CostoMaterial(FacSerie, FacNum) As Double
```

```
StrSql = "Select sum(FAM_precio_costo*FAM_Cantidad) as CostoMat from factura_material "
StrSql = StrSql + " WHERE "
StrSql = StrSql + " FAM_FAC_Serie = "" + FacSerie + """
```

```
StrSql = StrSql + " AND FAM_FAC_num = " + FacNum
Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
CostoMaterial = If(IsNull(Rds("CostoMat")), 0, Rds("CostoMat"))
Rds.Close
End Function
Function GeneraPolizaModelo() As Boolean
    Dim Aux, C1, C2, C3, C4, C5, Folio, x As Integer
    Dim Monto As Double
    Dim Rds1 As Recordset

    GeneraPolizaModelo = False

    If Not InsertaPoliza("Cancelación de la Factura Serie: " + TxtSerie + " no." + Str$(TxtNumFactura))
Then
        Exit Function
    End If

    ' CARGO a dev produccion, postproduccion, copiado
    StrSql = ""
    StrSql = StrSql + " select "
    StrSql = StrSql + " from factura_servicio "
    StrSql = StrSql + " where fas_fac_serie = " + TxtSerie + " "
    StrSql = StrSql + " and fas_fac_num = " + TxtNumFactura + " "
    Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)

    Aux = 1
    While Not Rds1.EOF

        If Rds1("fas_tio_cve") = 2 Then 'copiado
            x = TraduceLlave(True, CuentaConta(14), C1, C2, C3, C4, C5)

        ElseIf Rds1("fas_tio_cve") = 3 Then 'produccion
            x = TraduceLlave(True, CuentaConta(13), C1, C2, C3, C4, C5)

        ElseIf Rds1("fas_tio_cve") = 1 Then 'postproduccion

            StrSql = ""
            StrSql = StrSql + " select sal_cuenta "
            StrSql = StrSql + " from sala, "
            StrSql = StrSql + " orden_servicio "
            StrSql = StrSql + " where ors_ord_num = " + Str$(Rds1("fas_ord_num"))
            StrSql = StrSql + " and ors_ser_cve = " + Str$(Rds1("fas_ser_cve"))
            StrSql = StrSql + " and ors_sal_cve = sal_cve "
            Set RdsAux = Db.OpenRecordset(StrSql, dbOpenSnapshot)

            If Not IsNull(RdsAux("sal_cuenta")) Then
                x = TraduceLlave(True, RdsAux("sal_cuenta"), C1, C2, C3, C4, C5)
            End If
            RdsAux.Close
        End If

        Monto = Cdbl(Rds1("fas_cant")) * Cdbl(Rds1("fas_precio_unidad")) * (1 -
Cdbl(Rds1("fas_descuento"))) / 100
    End While
End Function
```

InsertaPartida Aux, C1, C2, C3, C4, C5, 1, Monto, 0

Rds1.MoveNext

Aux = Aux + 1

Wend

Rds1.Close

'CARGO a dev materiales

StrSql = ""

StrSql = StrSql + " select sum(fam_cantidad * fam_precio_venta * (1 - fam_descuento / 100)) as Monto

StrSql = StrSql + " from factura_material "

StrSql = StrSql + " where fam_fac_serie = "" + TxtSerie + "" "

StrSql = StrSql + " and fam_fac_num = "" + TxtNumFactura + "" "

Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)

If Not IsNull(Rds1("Monto")) Then

x = TraduceLlave(True, CuentaConta(15), C1, C2, C3, C4, C5)

InsertaPartida Aux, C1, C2, C3, C4, C5, 1, Rds1("Monto"), 0

End If

Rds1.Close

' CARGO a Iva

x = TraduceLlave(True, CuentaConta(2), C1, C2, C3, C4, C5)

InsertaPartida Aux + 1, C1, C2, C3, C4, C5, 1, TxtConIva, 0

' ABONO Clientes

x = TraduceLlave(True, CuentaConta(1), C1, C2, C3, C4, C5)

InsertaPartida Aux + 2, C1, C2, C3, C4, C5, 2, 0, TxtTotal

'otra poliza para almacenes

Monto = CostoMaterial(TxtSerie, TxtNumFactura)

If Monto <> 0 Then

If Not InsertaPoliza("Cancelación de Materiales de la Factura Serie: " + TxtSerie + " no." +

Str\$(TxtNumFactura)) Then

Exit Function

End If

' CARGO a Almacen de Materiales

x = TraduceLlave(True, CuentaConta(16), C1, C2, C3, C4, C5)

InsertaPartida 1, C1, C2, C3, C4, C5, 1, Monto, 0

' ABONO Costo de Materiales

x = TraduceLlave(True, CuentaConta(19), C1, C2, C3, C4, C5)

InsertaPartida 2, C1, C2, C3, C4, C5, 2, 0, Monto

End If

GeneraPolizaModelo = True

End Function

Function ExisteCtaConta() As Boolean

Dim Rds1 As Recordset

ExisteCtaConta = False

' salas

StrSql = ""

StrSql = StrSql + " select sal_cuenta from sala"

Set Rds1 = Db.OpenRecordset(StrSql, dbOpenSnapshot)

While Not Rds1.EOF

 If IsNull(Rds1("sal_cuenta")) Then

 Rds1.Close

 Exit Function

 Elseif Not ExisteCta(Rds1("sal_cuenta")) Then

 Rds1.Close

 Exit Function

 End If

 Rds1.MoveNext

Wend

Rds1.Close

' clientes1 iva2 produccion13 copiado14 materiales15 almacen16 costumateriales19

If Not ExisteCta(CuentaConta(1)) Or _

 Not ExisteCta(CuentaConta(2)) Or _

 Not ExisteCta(CuentaConta(13)) Or _

 Not ExisteCta(CuentaConta(14)) Or _

 Not ExisteCta(CuentaConta(15)) Or _

 Not ExisteCta(CuentaConta(16)) Or _

 Not ExisteCta(CuentaConta(19)) Then

 Exit Function

End If

ExisteCtaConta = True

End Function

Private Function ActualizaFactura() As Boolean

ActualizaFactura = False

On Error GoTo ManejaErrores

StrSql = ""

StrSql = StrSql + " Update Factura set "

StrSql = StrSql + " FAC_STA_Cve = 13, "

StrSql = StrSql + " FAC_observacion = " + TxtObservaciones + ""

StrSql = StrSql + " Where FAC_Serie = " + TxtSerie + ""

StrSql = StrSql + " And FAC_Num = " + TxtNumFactura

Db Execute StrSql, dbFailOnError

ActualizaFactura = True

Exit Function

ManejaErrores:

 Errores (Err)

 Screen.MousePointer = 0

End Function

Private Function ActualizaSaldoCli() As Boolean

 ActualizaSaldoCli = False

 On Error GoTo ManejaErrores

 StrSql = ""

 StrSql = StrSql + "Update Cliente set "

 StrSql = StrSql + " CLI_Saldo_Pendiente = cli_saldo_pendiente - " + TxtTotal + ";

 StrSql = StrSql + " CLI_Saldo_real = cli_saldo_real - " + TxtTotal

 StrSql = StrSql + " Where CLI_Cve = " + CmbCliente.BoundText

 Db.Execute StrSql, dbFailOnError

 TxtSaldoReal = TxtSaldoReal - TxtTotal

 ActualizaSaldoCli = True

Exit Function

ManejaErrores:

 Errores (Err)

 Screen.MousePointer = 0

End Function

Function CargaDatosFac() As Boolean

 Screen.MousePointer = 11

 CargaDatosFac = False

 If CmbCliente.BoundText <> "" Then

 StrSql = ""

 StrSql = StrSql + " select fac_serie, "

 StrSql = StrSql + " fac_num, "

 StrSql = StrSql + " fac_subtotal, "

 StrSql = StrSql + " fac_totaliva, "

 StrSql = StrSql + " fac_total, "

 StrSql = StrSql + " fac_fecha, "

 StrSql = StrSql + " fac_observacion "

 StrSql = StrSql + " from factura "

 StrSql = StrSql + " where fac_cli_conduc = " + CmbCliente.BoundText

 StrSql = StrSql + " and fac_sta_cve = 6"

 StrSql = StrSql + " and Year(fac_fecha) = Year(CVDate (" + Format\$(Date, "dd/mm/yy") + "))) "

 If Not DefineDataControl(CdFactura, DbName, DbDir, StrSql, vbRSTypeSnapshot) Then

 Exit Function

 End If

```
CdFactura.Refresh  
GrdFactura.Refresh  
CmdCancelarFac.Enabled = True
```

```
End If
```

```
CargaDatosFac = True  
Screen.MousePointer = 0
```

```
End Function
```

```
Function CargaDatosCli() As Boolean
```

```
Screen.MousePointer = 11  
CargaDatosCli = False
```

```
Set Query = Db.QueryDefs("CLICve")  
Query.Parameters("CLICve") = CmbCliente.BoundText  
Set Rds = Query.OpenRecordset(dbOpenSnapshot, dbReadOnly)
```

```
If Rds.RecordCount = 0 Then  
    MsgBox "No se pudo recuperar datos del cliente", 64  
    Rds.Close  
    Exit Function  
End If
```

```
TxtSaldoReal = Format(Rds("CLI_Saldo_Real"), "$###,###.00")  
TxtDireccion = Rds("CLI_Direccion")  
TxtRfc = Rds("CLI_Rfc")  
Rds.Close
```

```
CargaDatosCli = True  
Screen.MousePointer = 0
```

```
End Function
```

```
Private Function LimpiaCamposCli()
```

```
LimpiaCamposCli = False  
TxtSaldoReal = 0  
TxtDireccion = ""  
TxtRfc = ""  
LimpiaCamposCli = True
```

```
End Function
```

```
Private Sub CmbCliente_Click(Area As Integer)
```

```
If CmbCliente.BoundText <> "" Then  
  
    If Not LimpiaCamposCli() Then  
        Exit Sub  
    End If
```

```
If Not CargaDatosCli() Then
    Exit Sub
End If

If Not CargaDatosFac() Then
    Exit Sub
End If

End If

End Sub

Private Sub CmdCancelarFac_Click()

    If GrdFactura.SelBookmarks.Count > 1 Then
        Beep
        MsgBox "Debe seleccionar solo un registro", 64
        Exit Sub
    ElseIf GrdFactura.SelBookmarks.Count = 0 Then
        Beep
        MsgBox "Debe seleccionar al menos un registro", 64
        Exit Sub
    End If

    TxtSerie = GrdFactura.Columns("fac_serie").Value
    TxtNumFactura = GrdFactura.Columns("fac_num").Value
    TxtObservaciones = GrdFactura.Columns("fac_observacion").Value
    TxtConlva = GrdFactura.Columns("fac_Totalva").Value
    TxtTotal = GrdFactura.Columns("fac_Total").Value

    'Se verifica si existen todas las cuentas contables
    If ExisteCtaConta Then

        'Se crea la póliza con sus parcialidades
        If Not GeneraPolizaModelo Then
            MsgBox "Existió un error al refejar contablemente, para verificar la póliza automática, deberá ingresar esta factura en la Contabilidad en forma manual.", 64
            Exit Sub
        End If

    Else
        MsgBox "No existen las Cuentas de Contabilidad para hacer la póliza automática, deberá ingresar esta factura en la Contabilidad en forma manual.", 64
        Exit Sub
    End If

    ActualizaFactura
    ActualizaSaldoCli
    CancelaFactura

    If Not CargaDatosFac() Then
        Exit Sub
    End If
```

End Sub

Private Sub CmdSalir_Click()

 Unload Me

End Sub

Private Function VerificaDatos() As Boolean

 VerificaDatos = False

 If CmbCliente.BoundText = "" Then

 MsgBox "Debe seleccionar un cliente", 64

 CmbCliente.SetFocus

 Exit Function

 End If

 If Not IsDate(TxtFecha) Then

 MsgBox "Debe seleccionar una fecha con formato DD/MM/AA", 64

 TxtFecha.SetFocus

 Exit Function

 End If

 VerificaDatos = True

End Function

Private Sub Form_Load()

 Me.WindowState = 2

 Screen.MousePointer = 11

 StrSql = "CLI"

 If Not DefineDataControl(CdCliente, DbName, DbDir, StrSql, vbRSTypeSnapshot) Then

 Exit Sub

 End If

 CdCliente.Refresh

 If Not LimpiaCamposCli() Then

 Exit Sub

 End If

 TxtFecha = Date

 CmdCancelarFac.Enabled = False

 CmbCliente.BoundText = ""

 Screen.MousePointer = 0

End Sub

Private Sub BorraOrdenMaterial(Orden)

 'borrado de materiales

 StrSql = ""

 StrSql = StrSql + "Delete *"

 StrSql = StrSql + "From factura_material"

 StrSql = StrSql + "where fam_osa_num = " + Str\$(Orden) + ""

 StrSql = StrSql + "and fam_fac_serie = " + TxtSerie + ""

 StrSql = StrSql + "and fam_fac_num = " + TxtNumFactura

```
Db.Execute strSql, dbFailOnError
```

```
strSql = ""
```

```
strSql = strSql + "Update orden_salida_almacen set osa_sta_cve = 4 "
```

```
strSql = strSql + "where osa_num = " + Str$(Orden)
```

```
Db.Execute strSql, dbFailOnError
```

```
End Sub
```

Pantalla para realización de pagos

Esta pantalla permitirá realizar pagos tanto ordinarios de facturas que se deben como anticipados.

La pantalla se divide en dos partes. En la parte superior aparecerá una lista de clientes, de la cual se podrá seleccionar alguno. Una vez seleccionado el cliente se desplegará tanto su rfc como su dirección, además de pedir información adicional para el pago como: forma de pago, número de documento y banco si se trata de un cheque, y el importe a pagar.

En la parte inferior de la pantalla aparecerán todas las facturas que están pendientes por pagar. Existe un botón para ir seleccionado que facturas deseamos cubrir con este pago.

Option Explicit

Dim Anticipo, Anti, HayPAF, Salir As Boolean

Dim TotalNoFac(50), TotalFac(50) As Double

Dim SerieNoFac(50), SerieFac(50) As String

Dim NumNoFac(50), NumFac(50) As Integer

Dim Pago, Suma, Pendiente As Double

Dim Status As Integer

Dim Folio As Integer

Private Function ActualizaAdelanto() As Boolean

ActualizaAdelanto = False

On Error GoTo ManejaErrores

If Suma > 0 Then

StrSql = ""

StrSql = StrSql + "Update Pago set "

StrSql = StrSql + "PAG_Anticipo=" + Format(Suma, "#####0.00")

StrSql = StrSql + " Where PAG_Cve = " + TxtNoRecibo

Db.Execute StrSql, dbFailOnError

End If

ActualizaAdelanto = True

Exit Function

ManejaErrores:

Errores (Err)

Screen.MousePointer = 0

Rollback

Exit Function

End Function

Private Function ActualizaFactura(Anti, Status, Pendiente, Indice) As Boolean

ActualizaFactura = False

On Error GoTo ManejaErrores

StrSql = ""

StrSql = StrSql + "Update Factura set "

```

StrSql = StrSql + "FAC_STA_Cve=" + Str$(Status)
StrSql = StrSql + ",FAC_Fecha_Pago=" + "CVDate(" + Format$(TxtFecha, "dd/mm/yy") + ")")
StrSql = StrSql + ",FAC_Pago_Pendiente=" + Format(Pendiente, "#####0.00")
If Anti Then
    StrSql = StrSql + " Where FAC_Serie = " + SerieNoFac(Indice) + ""
    StrSql = StrSql + " And FAC_Num = " + Str$(NumNoFac(Indice))
Else
    StrSql = StrSql + " Where FAC_Serie = " + SerieFac(Indice) + ""
    StrSql = StrSql + " And FAC_Num = " + Str$(NumFac(Indice))
End If
Db.Execute StrSql, dbFailOnError

```

ActualizaFactura = True

Exit Function

ManejaErrores:

```

Errores (Err)
Screen.MousePointer = 0
Rollback
Exit Function

```

End Function

Private Function ActualizaSaldoCli(Tipo, Cantidad) As Boolean

Dim Saldo As Double

ActualizaSaldoCli = False

On Error GoTo ManejaErrores

If Tipo = 1 Then

```

StrSql = ""
StrSql = StrSql + "Select CLI_Saldo_Pendiente from cliente "
StrSql = StrSql + " Where CLI_Cve = " + CmbClave.BoundText
Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)

```

If IsNull(Rds("CLI_Saldo_Pendiente")) Then

Saldo = 0

Else

Saldo = Rds("CLI_Saldo_Pendiente")

End If

Saldo = Saldo - Cantidad

Rds.Close

```

StrSql = ""
StrSql = StrSql + "Update Cliente set "
StrSql = StrSql + "CLI_Saldo_Pendiente = " + Format(Saldo, "#####0.00")
StrSql = StrSql + " Where CLI_Cve = " + CmbClave.BoundText
Db.Execute StrSql, dbFailOnError

```

Elsif Tipo = 2 Then

```

StrSql = ""
StrSql = StrSql + "Select CLI_Saldo_Real from cliente "

```

```
StrSql = StrSql + " Where CLI_Cve = " + CmbClave.BoundText  
Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
If IsNull(Rds("CLI_Saldo_Real")) Then  
    Saldo = 0  
Else  
    Saldo = Rds("CLI_Saldo_Real")  
End If
```

```
Saldo = Saldo - Cantidad  
Rds.Close
```

```
StrSql = ""  
StrSql = StrSql + "Update Cliente set "  
StrSql = StrSql + "CLI_Saldo_Real = " + Format(Saldo, "#####0.00")  
StrSql = StrSql + " Where CLI_Cve = " + CmbClave.BoundText  
Db.Execute StrSql, dbFailOnError  
TxtSaldo.Text = Saldo
```

```
Elseif Tipo = 3 Then
```

```
StrSql = ""  
StrSql = StrSql + "Select CLI_Saldo_Anticipado from cliente "  
StrSql = StrSql + " Where CLI_Cve = " + CmbClave.BoundText  
Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
If IsNull(Rds("CLI_Saldo_Anticipado")) Then  
    Saldo = 0  
Else  
    Saldo = Rds("CLI_Saldo_Anticipado")  
End If
```

```
Saldo = Saldo + Cantidad  
Rds.Close
```

```
StrSql = ""  
StrSql = StrSql + "Update Cliente set "  
StrSql = StrSql + "CLI_Saldo_Anticipado = " + Format(Saldo, "#####0.00")  
StrSql = StrSql + " Where CLI_Cve = " + CmbClave.BoundText  
Db.Execute StrSql, dbFailOnError
```

```
End If
```

```
ActualizaSaldoCli = True
```

```
Exit Function
```

```
ManejaErrores:
```

```
    Errores (Err)
```

```
    Screen.MousePointer = 0
```

```
    Rollback
```

```
    Exit Function
```

```
End Function
```

Private Function AsignaTotalFacNoPagadas()

```
TotalNoFac(LstFacNoPagadas.ListCount) = TotalFac(LstFacPagadas.ListIndex)
SerieNoFac(LstFacNoPagadas.ListCount) = SerieFac(LstFacPagadas.ListIndex)
NumNoFac(LstFacNoPagadas.ListCount) = NumFac(LstFacPagadas.ListIndex)
```

End Function

Private Function AsignaTotalFacPagadas()

```
TotalFac(LstFacPagadas.ListCount) = TotalNoFac(LstFacNoPagadas.ListIndex)
SerieFac(LstFacPagadas.ListCount) = SerieNoFac(LstFacNoPagadas.ListIndex)
NumFac(LstFacPagadas.ListCount) = NumNoFac(LstFacNoPagadas.ListIndex)
```

End Function

Private Function BuscaFecha() As Integer

Dim Num As Integer

```
StrSql = ""
StrSql = StrSql + " Select * "
StrSql = StrSql + " From Factura "
StrSql = StrSql + " Where fac_cli_conduc= " + Str$(CmbClave.BoundText)
StrSql = StrSql + " And FAC_STA_Cve=6 "
StrSql = StrSql + " Or FAC_STA_Cve=9 "
StrSql = StrSql + " And Fac_Fecha= "
StrSql = StrSql + "(Select min(Fac_Fecha) as F "
StrSql = StrSql + " From factura "
StrSql = StrSql + " Where fac_cli_conduc= " + Str$(CmbClave.BoundText)
StrSql = StrSql + " And (FAC_STA_Cve=6 "
StrSql = StrSql + " Or FAC_STA_Cve=9))"
Set Rds = Db.OpenRecordset(StrSql, dbOpenDynaset)
```

Num = 0

If Rds.RecordCount <> 0 Then

While Num < LstFacNoPagadas.ListCount

If Rds("fac_serie") = SerieNoFac(Num) And Rds("fac_num") = NumNoFac(Num) Then

BuscaFecha = Num

LstFacNoPagadas.ListIndex = Num

Rds.Close

Exit Function

End If

Num = Num + 1

Wend

End If

Rds.Close

End Function

```
Private Function CobraFacturaRestante(Tipo)
    Dim Aux1, Num As Integer

    CobraFacturaRestante = False
    Aux1 = LstFacNoPagadas.ListCount
    If Suma > 0 And LstFacNoPagadas.ListCount > 0 Then
        While Aux1 <> 0
            Num = BuscaFecha
            If TotalNoFac(Num) <> 0 And SerieNoFac(Num) <> Str$(0) And NumNoFac(Num) <> 0 And Suma
                > 0 Then

                Anti = True
                If Not GuardaPagoFactura(Anti, Num) Then
                    Exit Function
                End If
                AsignaTotalFacPagadas
                TxtTotalFacNo = TxtTotalFacNo.Text - Pago
                MueveTotalFacNoPagadas

                If Status = 9 And Suma = 0 Then
                    TxtTotalFac = TxtTotalFac + Pago
                    StrSql = "De la factura " + LstFacNoPagadas.List(LstFacNoPagadas.ListIndex) + " solo se
hizo un pago de $" + Str$(Pago)
                    MsgBox StrSql
                    CobraFacturaRestante = True
                    Aux1 = 0
                Else
                    LstFacPagadas.AddItem LstFacNoPagadas.List(LstFacNoPagadas.ListIndex)
                    LstFacPagadas.ItemData(LstFacPagadas.ListCount - 1) =
LstFacNoPagadas.ItemData(LstFacNoPagadas.ListIndex)
                    LstFacNoPagadas.RemoveItem LstFacNoPagadas.ListIndex
                    TxtTotalFac = TxtTotalFac + TotalFac(LstFacPagadas.ListCount - 1)
                    Aux1 = LstFacNoPagadas.ListCount
                End If
            Else
                Aux1 = 0
            End If

        Wend

    End If

    If Tipo = 2 Then

        If Not ActualizaSaldoCli(1, TxtMonto - Suma) Then
            Screen.MousePointer = 0
            Exit Function
        End If
        If Not ActualizaSaldoCli(2, TxtMonto) Then
            Screen.MousePointer = 0
            Exit Function
        End If
        If Not ActualizaSaldoCli(3, Suma) Then
            Screen.MousePointer = 0
```

```
Exit Function
End If
```

```
If Not ActualizaAdelanto() Then
    Screen.MousePointer = 0
Exit Function
End If
```

```
End If
```

```
CobraFacturaRestante = True
```

```
End Function
```

```
Private Function DetectaStatus(Anti, Pago, Suma, Pendiente As Double, Status As Integer, Indice) As Boolean
```

```
DetectaStatus = False
```

```
If Anti Then
```

```
    If TotalNoFac(Indice) <= Val(Suma) Then
```

```
        Status = 8
```

```
        Pendiente = 0
```

```
        Pago = TotalNoFac(Indice)
```

```
        Suma = Suma - Pago
```

```
    Else
```

```
        Status = 9
```

```
        Pendiente = TotalNoFac(Indice) - Val(Suma)
```

```
        Pago = Suma
```

```
        Suma = Suma - Pago
```

```
    End If
```

```
Else
```

```
    If TotalFac(Indice) <= Val(Suma) Then
```

```
        Status = 8
```

```
        Pendiente = 0
```

```
        Pago = TotalFac(Indice)
```

```
        Suma = Suma - Pago
```

```
    Else
```

```
        Status = 9
```

```
        Pendiente = TotalFac(Indice) - Val(Suma)
```

```
        Pago = Suma
```

```
        Suma = Suma - Pago
```

```
    End If
```

```
End If
```

```
DetectaStatus = True
```

```
End Function
```

```
Private Function EnvíaReporte() As Integer
```

```
    CrpRep.Destination = 0
```

```
    CrpRep.ReportFileName = RepPath + "RepRecPago.Rpt"
```

```
    CrpRep.Connect = "DSN = ODBCReservaV4"
```

```
    strSql = "{Pago Pag_Cve}=" + TxtNoRecibo
```

```
CrpRep.SelectionFormula = StrSql  
CrpRep.WindowWidth = 600  
CrpRep.WindowHeight = 350  
CrpRep.WindowTop = 50  
CrpRep.WindowLeft = 10
```

```
On Error GoTo ManejaError
```

```
    CrpRep.Action = 1
```

```
EnviaReporte = True
```

```
Exit Function
```

```
ManejaError:
```

```
    Errores (Err)
```

```
End Function
```

```
Function CargaDatos(Clave) As Boolean
```

```
    Screen.MousePointer = 11
```

```
    CargaDatos = False
```

```
    StrSql = ""
```

```
    StrSql = StrSql + " Select max (pag_cve) as cuenta "
```

```
    StrSql = StrSql + " from pago "
```

```
    Set Rds = Db.OpenRecordset(StrSql, dbOpenSnapshot)
```

```
    If IsNull(Rds("cuenta")) Then
```

```
        TxtNoRecibo = 1
```

```
    Else
```

```
        TxtNoRecibo = Rds("cuenta") + 1
```

```
    End If
```

```
    Rds.Close
```

```
    Set Query = Db.QueryDefs("CLICve")
```

```
    Query.Parameters("CLICve") = Clave
```

```
    Set Rds = Query.OpenRecordset(dbOpenSnapshot, dbReadOnly)
```

```
    If Rds.RecordCount = 0 Then
```

```
        MsgBox "No se pudo recuperar datos del cliente", 64
```

```
        Exit Function
```

```
    End If
```

```
    TxtSaldo = Format(Rds("CLI_Saldo_Real"), "$###,###.00")
```

```
    TxtDireccion = Rds("CLI_Direccion")
```

```
    TxtRfc = Rds("CLI_Rfc")
```

```
    Rds.Close
```

```
    CargaDatos = True
```

```
    Screen.MousePointer = 0
```

```
End Function
```

```
Function CargaFacNoPagadas() As Boolean
```

```
    On Error GoTo ManejaErrores
```

```
    Screen.MousePointer = 11
```

```
    CargaFacNoPagadas = False
```

```

If CmbClave.BoundText <> "" Then
  strSql = ""
  strSql = strSql + " Select *, "
  strSql = strSql + "      (Str$(FAC_FECHA) + ' ' + "
  strSql = strSql + "      FAC_SERIE + ' ' + "
  strSql = strSql + "      Str$(FAC_Num) + ' $' + "
  strSql = strSql + "      Str$(FAC_Pago_Pendiente)) as Cadena "
  strSql = strSql + " From Factura "
  strSql = strSql + " Where fac_cli_conduc=" + Str$(CmbClave.BoundText)
  strSql = strSql + " And (FAC_STA_Cve=6 "
  strSql = strSql + " Or FAC_STA_Cve=9) "
  strSql = strSql + " order by FAC_Fecha"
  Set Rds = Db.OpenRecordset(StrSql, dbOpenDynaset)

  LstFacNoPagadas.Clear
  LstFacPagadas.Clear
  Erase TotalFac
  Erase TotalNoFac
  Erase SerieFac
  Erase SerieNoFac
  Erase NumFac
  Erase NumNoFac

  If Rds.RecordCount = 0 Then
    PnlFa.Visible = False
    MsgBox "No existen facturas por pagar para este cliente", 64
    HayPAF = True
    Screen.MousePointer = 0
    Exit Function
  End If

  TxtTotalFacNo = 0
  TxtTotalFac = 0
  While Not Rds.EOF
    TotalNoFac(LstFacNoPagadas.ListCount) = Rds("fac_pago_pendiente")
    SerieNoFac(LstFacNoPagadas.ListCount) = Rds("fac_serie")
    NumNoFac(LstFacNoPagadas.ListCount) = Rds("fac_num")

    TxtTotalFacNo = TxtTotalFacNo + TotalNoFac(LstFacNoPagadas.ListCount)

    LstFacNoPagadas.AddItem Rds("Cadena")
    LstFacNoPagadas.ItemData(LstFacNoPagadas.ListCount - 1) = LstFacNoPagadas.ListCount
    Rds.MoveNext
  Wend
  Rds.Close

  Screen.MousePointer = 0
  CargaFacNoPagadas = True
End If

Exit Function

ManejaErrores

```

Errores (Err)
Screen.MousePointer = 0
Exit Function

End Function

Private Function GeneraPAF(Pago, Anti, Indice) As Boolean

GeneraPAF = False
On Error GoTo ManejaErrores

```
StrSql = ""  
StrSql = StrSql + "Insert into Pago_Factura "  
StrSql = StrSql + "(PAF_PAG_Cve, "  
StrSql = StrSql + "PAF_FAC_SERIE, "  
StrSql = StrSql + "PAF_FAC_NUM, "  
StrSql = StrSql + "PAF_MONTO) "  
StrSql = StrSql + "Values (" + TxtNoRecibo + ", "  
If Anti = False Then  
    StrSql = StrSql + SerieFac(Indice) + ", "  
    StrSql = StrSql + Str$(NumFac(Indice)) + ", "  
Else  
    StrSql = StrSql + SerieNoFac(Indice) + ", "  
    StrSql = StrSql + Str$(NumNoFac(Indice)) + ", "  
End If
```

```
StrSql = StrSql + Str$(Format(Pago, "#####.00"))  
StrSql = StrSql + ") "  
Db.Execute (StrSql), ModeSql
```

GeneraPAF = True

Exit Function

ManejaErrores:

 Errores (Err)
 Screen.MousePointer = 0
 Rollback

Exit Function

End Function

Private Function GuardaPago() As Boolean

 GuardaPago = False

On Error GoTo ManejaError

If LstFacPagadas.ListCount = 0 Then

 OppAnti.Value = True

End If

```
StrSql = ""  
StrSql = StrSql + "Insert into pago "  
StrSql = StrSql + "    (pag_cve, "  
StrSql = StrSql + "    pag_cli_cve, "  
StrSql = StrSql + "    pag_sta_cve, "  
StrSql = StrSql + "    pag_fop_cve, "  
StrSql = StrSql + "    pag_fecha, "  
StrSql = StrSql + "    pag_monto, "
```

```

StrSql = StrSql + "      pag_num_docto, "
StrSql = StrSql + "      pag_observaciones, "
StrSql = StrSql + "      pag_ban_cve, "
StrSql = StrSql + "      pag_antipico)"
StrSql = StrSql + "values (" + TxtNoRecibo
StrSql = StrSql + "," + Str$(CmbClave.BoundText)
StrSql = StrSql + "," + Str$(10)
StrSql = StrSql + "," + Str$(CmbFormaPago.BoundText)
StrSql = StrSql + ",CVDate(" + Format$(TxtFecha, "dd/mm/yy") + ")")
StrSql = StrSql + "," + Format(TxtMonto, "#####.00")
StrSql = StrSql + "," + TxtNoDocto
StrSql = StrSql + "," + TxtObservaciones
If CmbBanco = "" Then
    StrSql = StrSql + ",0"
Else
    StrSql = StrSql + "," + CmbBanco.BoundText
End If
If OpbAnti.Value = True Then
    StrSql = StrSql + "," + Format(TxtMonto, "#####.00")
    Antipico = True
    HayPAF = True
Elseif OpbOrdinario.Value = True Then
    StrSql = StrSql + ",0"
    Antipico = False
End If
StrSql = StrSql + ")")
Db.Execute StrSql, dbFailOnError

GuardaPago = True
Exit Function
ManejaError:
    Errores (Err)
End Function

```

```

Private Function GuardaPagoFactura(Anti, Indice)

    GuardaPagoFactura = False

    Status = 0
    If Not DetectaStatus(Anti, Pago, Suma, Pendiente, Status, Indice) Then
        Exit Function
    End If

    If Not GeneraPAF(Pago, Anti, Indice) Then
        Exit Function
    End If
    If Not ActualizaFactura(Anti, Status, Pendiente, Indice) Then
        Exit Function
    End If

    GuardaPagoFactura = True

End Function

```

```

Private Function HayFacturas()

```

```
StrSql = ""
StrSql = StrSql + " Select * "
StrSql = StrSql + " From Factura "
StrSql = StrSql + " Where fac_cli_conduc= " + Str$(CmbClave.BoundText)
StrSql = StrSql + " And( FAC_STA_Cve=6 "
StrSql = StrSql + " Or FAC_STA_Cve=9)"
Set Rds = Db.OpenRecordset(StrSql, dbOpenDynaset)
```

```
If Rds.RecordCount = 0 Then
    OpbAnti.Value = True
    Anticipo = True
    HayPAF = True
    MsgBox " No existen facturas por cobrar de este cliente", 64
End If
Rds.Close
```

End Function

Private Function InsertaPoliza()

```
InsertaPoliza = False
On Error GoTo ManejaError
```

```
Folio = GeneraFolio(TxtFecha)
```

```
If Folio > 0 Then
```

```
    StrSql = ""
    StrSql = StrSql + "Insert into poliza "
    StrSql = StrSql + "      (pol_fecha, "
    StrSql = StrSql + "      pol_folio, "
    StrSql = StrSql + "      pol_tip_cve, "
    StrSql = StrSql + "      pol_tpo_cve, "
    StrSql = StrSql + "      pol_are_cve, "
    StrSql = StrSql + "      pol_concepto)"
    StrSql = StrSql + "values (CDate(" + Format$(TxtFecha, "dd/mm/yy") + "),"
    StrSql = StrSql + "      " + Str$(Folio) + ","
    StrSql = StrSql + "      4,"
    StrSql = StrSql + "      "" + CuentaConta(4) + ""),"
    StrSql = StrSql + "      "" + CuentaConta(5) + ""),"
    StrSql = StrSql + "      'Pago de recibo no.' + Str$(TxtNoRecibo) + '')"
    Db.Execute StrSql, dbFailOnError
```

```
InsertaPoliza = True
```

```
InsertaPartida
```

```
End If
Exit Function
```

```
ManejaError:
    Errores (Err)
```

```
End Function
```

```
Sub InsertaPartida()
```

Dim Tipo, CCL, CMA, CCO, SCU, SSC As Integer
 Dim Cadena As String

On Error GoTo ManejaError

Aux = 1

While Aux < 3

if Aux = 1 Then

Tipo = 1

Cadena = CuentaConta(3)

Else

Tipo = 2

Cadena = CuentaConta(1)

End If

If Not TraduceLlave(True, Cadena, CCL, CMA, CCO, SCU, SSC) Then

Exit Sub

End If

StrSql = ""

StrSql = StrSql + "Insert into partida "

StrSql = StrSql + " (par_pol_fecha, "

StrSql = StrSql + " par_pol_folio, "

StrSql = StrSql + " par_consec, "

StrSql = StrSql + " par_ssc_scu_cco_cma_ccl_cve, "

StrSql = StrSql + " par_ssc_scu_cco_cma_cve. "

StrSql = StrSql + " par_ssc_scu_cco_cve. "

StrSql = StrSql + " par_ssc_scu_cve. "

StrSql = StrSql + " par_ssc_cve, "

StrSql = StrSql + " par_debehaber, "

StrSql = StrSql + " par_concepto, "

StrSql = StrSql + " par_debe, "

StrSql = StrSql + " par_haber) "

StrSql = StrSql + "values (CVDate("" + Format\$(TxtFecha, "dd/mm/yy") + ""), "

StrSql = StrSql + " " + Str\$(Folio) + ", "

StrSql = StrSql + " " + Str\$(Aux) + ", "

StrSql = StrSql + " " + Str\$(CCL) + ", "

StrSql = StrSql + " " + Str\$(CMA) + ", "

StrSql = StrSql + " " + Str\$(CCO) + ", "

StrSql = StrSql + " " + Str\$(SCU) + ", "

StrSql = StrSql + " " + Str\$(SSC) + ", "

StrSql = StrSql + " " + Str\$(Tipo) + ", "

StrSql = StrSql + " 'Pago del Recibo No. " + Str\$(TxtNoRecibo) + "' "

If Tipo = 1 Then

StrSql = StrSql + " " + TxtMonto + ",0"

Else

StrSql = StrSql + " 0," + TxtMonto

End If

StrSql = StrSql + ")"

Db.Execute StrSql, dbFailOnError

Aux = Aux + 1

Wend

Exit Sub

ManejaError:
 Errores (Err)
End Sub

Private Function LimpiaCampos()

LimpiaCampos = False
TxtSaldo = "\$ 0.00"
TxtDireccion = ""
TxtRfc = ""
CmbFormaPago = ""
TxtNoDocto = ""
CmbBanco = ""
TxtMonto = "\$ 0.00"
TxtObservaciones = ""
LstFacNoPagadas.Clear
LstFacPagadas.Clear
LimpiaCampos = True

End Function

Private Sub CmbBanco_Click(Area As Integer)

 If Area = 2 Then
 CmbBanco.BoundText = CmbBanco.BoundText
 End If

End Sub

Private Sub CmbClave_Click(Area As Integer)

 If CmbClave.BoundText <> "" Then
 CmdMostrar.Enabled = True
 PnlFa.Visible = False
 PnlFa.Enabled = True
 CmdAceptar.Enabled = False
 OpbOrdinario.Value = True
 CmdAceptarPag.Enabled = False
 If Not LimpiaCampos() Then
 Exit Sub
 End If

 If Not CargaDatos(CmbClave.BoundText) Then
 Exit Sub
 End If

End If

End Sub

Private Sub CmbFormaPago_Click(Area As Integer)

 If Area = 2 Then
 CmbFormaPago.BoundText = CmbFormaPago.BoundText

```
If CmbFormaPago <> "" Then
```

```
    TxtNoDocto.Enabled = True
    TxtNoDocto = ""
    CmbBanco = ""
    TxtMonto = ""
```

```
    If CmbFormaPago.BoundsText = 2 Then
```

```
        TxtNoDocto = TxtNoRecibo
        TxtNoDocto.Enabled = False
        CmbBanco.Enabled = False
```

```
    ElseIf CmbFormaPago.BoundsText = 1 Or CmbFormaPago.BoundsText = 4 Or
```

```
CmbFormaPago.BoundsText = 5 Then
```

```
        CmbBanco.Enabled = True
        ElseIf CmbFormaPago.BoundsText = 3 Then
            CmbBanco.Enabled = False
```

```
    End If
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub CmdAceptar_Click()
```

```
    Screen.MousePointer = 11
```

```
    If Not AceptarPago() Then
```

```
        Screen.MousePointer = 0
```

```
        Exit Sub
```

```
    End If
```

```
    Aux = 0
```

```
    Suma = TxtMonto
```

```
    If LstFacPagadas.ListCount <> 0 Then
```

```
        While Aux < LstFacPagadas.ListCount
```

```
            HayPAF = True
```

```
            If TotalFac(Aux) <> 0 And SerieFac(Aux) <> Str$(0) And NumFac(Aux) <> 0 And Suma > 0 Then
```

```
                Anti = False
```

```
                If Not GuardaPagoFactura(Anti, Aux) Then
```

```
                    Screen.MousePointer = 0
```

```
                    Exit Sub
```

```
                End If
```

```
            End If
```

```
            Aux = Aux + 1
```

```
        Wend
```

```
    If Not ActualizaSaldoCli(1, TxtMonto - Suma) Then
```

```
        Screen.MousePointer = 0
```

```
        Exit Sub
```

```
    End If
```

```
    If Not ActualizaSaldoCli(2, TxtMonto) Then
```

```
        Screen.MousePointer = 0
```

```
        Exit Sub
```

End If

```
If Not ActualizaSaldoCli(3, Suma) Then
    Screen.MousePointer = 0
    Exit Sub
End If
```

```
If Not ActualizaAdelanto() Then
    Screen.MousePointer = 0
    Exit Sub
End If
```

```
PnlFa.Enabled = False
CmdAceptar.Enabled = False
```

```
If Not EnviaReporte() Then
    Exit Sub
End If
```

End If

```
Screen.MousePointer = 0
```

End Sub

Private Function AceptarPago()

```
AceptarPago = False
```

```
If Not GuardaPago() Then
    Screen.MousePointer = 0
    Exit Function
End If
```

```
If CmbFormaPago.BoundText = 3 Then
```

```
'Inserta póliza con fecha de hoy si es nota de crédito
If Not AltaNotaCredito(TxtMonto, TxtNoRecibo, TxtFecha) Then
    Screen.MousePointer = 0
    Exit Function
End If
```

Else

```
'Inserta póliza con fecha de hoy si no es nota de crédito
If Not InsertaPoliza() Then
    Screen.MousePointer = 0
    Exit Function
End If
```

End If

```
'Si es anticipo solo actualiza el saldo anticipado y real del cliente
AceptarPago = True
```

If Anticipo Then

```
AceptarPago = False
CmdAceptar.Enabled = False
If Not ActualizaSaldoCli(3, TxtMonto) Then
    Exit Function
End If
If Not ActualizaSaldoCli(2, TxtMonto) Then
    Exit Function
End If
```

```
AceptarPago = True
If Not EnvíaReporte() Then
    Exit Function
End If
```

End If

End Function

Private Sub Flechas()

```
If LstFacNoPagadas.ListCount = 0 Then
    CmdAsignar.Enabled = False
Else
    LstFacNoPagadas.ListIndex = 0
    CmdAsignar.Enabled = True
End If
```

```
If LstFacPagadas.ListCount = 0 Then
    CmdDesasignar.Enabled = False
Else
    LstFacPagadas.ListIndex = 0
    CmdDesasignar.Enabled = True
End If
```

End Sub

Private Sub CmdAsignar_Click()

```
If LstFacNoPagadas.ListIndex = -1 Then
    MsgBox "Debe seleccionar una Factura."
    Exit Sub
End If
```

```
AsignaTotalFacPagadas
TxtTotalFac = TxtTotalFac + TotalFac(LstFacPagadas.ListCount)
MueveTotalFacNoPagadas
TxtTotalFacNo = TxtTotalFacNo.Text - TotalFac(LstFacPagadas.ListCount)
LstFacPagadas.AddItem LstFacNoPagadas.List(LstFacNoPagadas.ListIndex)
LstFacPagadas.ItemData(LstFacNoPagadas.ListCount - 1) =
LstFacNoPagadas.ItemData(LstFacNoPagadas.ListIndex)
LstFacNoPagadas.RemoveItem LstFacNoPagadas.ListIndex
Flechas
```

End Sub

Private Function MueveTotalFacNoPagadas()

```
Aux = LstFacNoPagadas.ListIndex
While Aux < LstFacNoPagadas.ListCount
    TotalNoFac(Aux) = TotalNoFac(Aux + 1)
    SerieNoFac(Aux) = SerieNoFac(Aux + 1)
    NumNoFac(Aux) = NumNoFac(Aux + 1)
    Aux = Aux + 1
Wend
TotalNoFac(LstFacNoPagadas.ListCount) = 0
SerieNoFac(LstFacNoPagadas.ListCount) = Str$(0)
NumNoFac(LstFacNoPagadas.ListCount) = 0
```

End Function

Private Sub CmdDesasignar_Click()

```
If LstFacPagadas.ListIndex = -1 Then
    MsgBox "Debe seleccionar una Factura."
    Exit Sub
End If
```

```
AsignaTotalFacNoPagadas
MueveTotalFacPagadas
```

```
TxtTotalFacNo = TxtTotalFacNo.Text + TotalNoFac(LstFacNoPagadas.ListCount)
TxtTotalFac = TxtTotalFac - TotalNoFac(LstFacNoPagadas.ListCount)
```

```
LstFacNoPagadas.AddItem LstFacPagadas.List(LstFacPagadas.ListIndex)
LstFacNoPagadas.ItemData(LstFacNoPagadas.ListCount - 1) =
LstFacPagadas.ItemData(LstFacPagadas.ListIndex)
```

```
LstFacPagadas.RemoveItem LstFacPagadas.ListIndex
```

Flechas

End Sub

Private Function MueveTotalFacPagadas()

```
Aux = LstFacPagadas.ListIndex
While Aux < LstFacPagadas.ListCount
    TotalFac(Aux) = TotalFac(Aux + 1)
    SerieFac(Aux) = SerieFac(Aux + 1)
    NumFac(Aux) = NumFac(Aux + 1)
    Aux = Aux + 1
Wend
TotalFac(LstFacPagadas.ListCount) = 0
SerieFac(LstFacPagadas.ListCount) = Str$(0)
NumFac(LstFacPagadas.ListCount) = 0
```

End Function

Private Sub CmdMostrar_Click()

```
Screen.MousePointer = 11
```

```
HayPAF = False
```

```

If Not VerificaDatos() Then
    Screen.MousePointer = 0
    HayPAF = True
    Exit Sub
End If

```

```

Anticipo = False
HayFacturas
If Anticipo = False Then
    Botones (True)
    If Not CargaFacNoPagadas() Then
        CmdAceptar.Enabled = False
        Screen.MousePointer = 0
        Exit Sub
    End If
    Flechas

```

```

End If
CmdAceptar.Enabled = True
CmdMostrar.Enabled = False
Screen.MousePointer = 0

```

```
End Sub
```

```

Private Sub CmdSalir_Click()
    Unload Me
End Sub

```

```

Private Sub Botones(Encender As Boolean)

```

```

    If Encender Then
        PnlFa.Visible = True
        CmdAceptar.Enabled = True
    Else
        PnlFa.Visible = False
        CmdAceptar.Enabled = False
    End If

```

```
End Sub
```

```

Private Function VerificaDatos() As Boolean
    VerificaDatos = False

```

```

    If CmbClave = "" Then
        MsgBox "Debe seleccionar un cliente", 64
        CmbClave.SetFocus
        Exit Function
    End If

```

```

    If Not CmbFormaPago.MatchedWithList Then
        MsgBox "Debe seleccionar de la lista una forma de pago". 64
        CmbFormaPago.SetFocus
        Exit Function
    End If

```

```

    If CmbFormaPago.BoundText = 1 Or CmbFormaPago.BoundText = 4 Or CmbFormaPago.BoundText = 5 Then

```

```
If CmbBanco = "" Then
    MsgBox "Debe seleccionar un banco", 64
    CmbBanco.SetFocus
    Exit Function
End If
If Not CmbBanco.MatchedWithList Then
    MsgBox "Debe seleccionar de la lista un banco", 64
    CmbBanco.SetFocus
    Exit Function
End If
End If

If TxtNoDocto = "" Then
    MsgBox "Debe introducir un no. de documento", 64
    TxtNoDocto.SetFocus
    Exit Function
End If

If InStr(TxtNoDocto, "'") Then
    MsgBox "El apostrofé no es válido", 64
    TxtNoDocto.SetFocus
    Exit Function
End If

If TxtMonto = "" Then
    MsgBox "Debe introducir un monto a pagar", 64
    TxtMonto.SetFocus
    Exit Function
End If

If Not IsNumeric(TxtMonto) Then
    MsgBox "Debe introducir un monto a pagar", 64
    TxtMonto.SetFocus
    Exit Function
End If

If InStr(TxtObservaciones, "'") Then
    MsgBox " El Apostrofé no es caracter válido"
    TxtObservaciones.SetFocus
    Exit Function
End If

VerificaDatos = True
End Function

Private Sub Form_Load()

    Me.WindowState = 2
    Screen.MousePointer = 11

    'Verifica si existen la cuentas de contabilidad
    If Not ExisteCta(CuentaConta(1)) Or Not ExisteCta(CuentaConta(3)) Then
```

FE DE ERRATAS

En el encabezado de páginas pares rango 310-364 existe un error .Debe decir Diseño e implementación del sistema.

En la penúltima línea del ultimo párrafo de la página 112 no se lee bien. Dice estar en la red, distribución y balanceo de carga, ruteo de transacciones por rutas

En la última línea del tercer párrafo de la página 114 no se lee bien. Dice sistema local a una transacción especial en el sistema remoto.

Existen varios errores de dedo ,los pondremos en una tabla:

Página	Párrafo	Línea	Error	Corrección
118	1	2	computadara	computadora
118	1	8	realzar	realizar
119	5	5	diosponible	disponible
121	5	2	susuario	usuario
122	1	7	umentar	aumentar
122	3	4	Ingualmente	Igualmente
141	11	2	desarrillo	desarrollo
163	4	3	reulta	resulta
165	1	3	necesdidad	necesidad
370-385	-	-	catalogo	catálogo
370	2	3	numero	número
388	3	4	numero	número
381	3	13	limite	límite
383	2	2	limite	límite
381	3	13-14	credito	crédito