



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

FACULTAD DE INGENIERÍA

**SISTEMA PARA EL MANEJO CENTRALIZADO  
DE LA CONTABILIDAD NACIONAL  
DE UNA EMPRESA UTILIZANDO  
SYBASE Y POWERBUILDER**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACION

PRESENTAN:

**NOEMI CASAS SÁNCHEZ  
ALFONSO MONTERO REYES  
XAVIER ROJEL MARTÍNEZ**

ASESOR: **M.I. JUAN CARLOS ROA BEIZA**



MEXICO D.F.

ENERO 2000



Universidad Nacional  
Autónoma de México



**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 la U.N.A.M. y a la facultad de Ingeniería,  
por habernos formado profesionalmente.*

*Nuestro agradecimiento especial al  
M.F. Juan Carlos Roa Beiza,  
Por su acertada dirección en el  
desarrollo de este trabajo.*

*A mi padre por sus desvelos.  
A mi madre por esa capacidad de dar amor.  
Gracias a ambos por sus sacrificios y enseñanzas.  
Mi admiración y cariño por siempre.*

*A mis hermanas por su apoyo y por todos los  
momentos gratos que hemos pasado juntas.*

*A Alex, porque su existencia me enseñó a  
ver lo verdaderamente valioso de las personas.*

*A mis compañeros Alfonso y Xavier,  
por su participación en la terminación de este logro.  
Gracias a ambos por su ayuda.*

*Noemi Casas Pánchez.*

*A todos mis profesores y en especial a la  
Profesora Teresita Martínez Pérez y al  
Profesor Aurelio Javier Rojas Ramírez.*

*Xavier Rojas Martínez.*

*A mi mamá, por brindarme todo  
lo que necesité durante todo este tiempo,  
pero sobre todo por el amor que siempre me ha dado.*

*A mis abuelitos, porque sin ustedes  
no hubiera podido ser ni la mitad de lo que soy  
(allá en el cielo abuelita, mil gracias)*

*A Pete, Quien con su muy especial forma de ser  
me ha brindado todo su apoyo.*

*A mis hermanos, a quienes siempre tengo  
en mi mente (en las buenas y en las malas)*

*A mi bebé, por alentarme cada día con tu amor y  
comprensión, pero sobre todo, por tu infinita paciencia.*

*Alfonso Montero Reyes*

## CONTENIDO

Introducción.....	v
Objetivo .....	viii
<b>1. Marco Teórico Contabilidad</b>	
<b>1.1. Conceptos básicos de Contabilidad</b>	
1.1.1. Origen y Definición .....	1
1.1.2. Importancia y Utilidad .....	2
1.1.3. Principios de Contabilidad Generalmente Aceptados .....	4
1.1.4. Registro de Operaciones .....	7
1.1.5. Clasificación de Entidades Económicas .....	8
1.1.6. Ecuación Básica de la Contabilidad.....	10
1.1.6.1. Definición y Clasificación de Activo .....	11
1.1.6.2. Definición y Clasificación de Pasivo .....	12
1.1.6.3. Definición de Capital Contable .....	13
1.1.7. Principio de Dualidad Económica .....	14
1.1.8. Reglas del Cargo y el Abono .....	15
1.1.9. Cuentas y Catálogo de Cuentas .....	16
1.1.10. Libros Principales	
1.1.10.1. Libro de Diario .....	20
1.1.10.2. Libro de Mayor .....	20
1.1.11. Libros Auxiliares .....	21
1.1.12. Balanza de Comprobación.....	22
1.1.13. Estados Financieros	

1.1.13.1. Objetivo y Utilidad .....	27
1.1.13.2. Balance General.....	29
1.1.13.3. Estado de Resultados .....	31
1.1.13.4. Estado de Variaciones en el Capital Contable .....	34
1.1.13.5. Estado de Cambios en la Situación Financiera en Base a Efectivo .	34
<b>2. Marco Teórico Computación</b>	
<b>2.1. Sistemas Operativos (Características, Ventajas y Desventajas)</b>	
2.1.1. UNIX .....	35
2.1.2. WINDOWS NT.....	45
<b>2.2. Comunicaciones</b>	
2.2.1. Redes de Computadoras .....	68
2.2.1.1. Uso de Redes de Computadoras .....	70
2.2.1.2. Objetivos .....	70
2.2.1.3. Aplicaciones .....	72
2.2.1.4. Estructura .....	73
2.2.1.5. Arquitectura.....	85
2.2.1.6. Canales Punto a Punto .....	86
2.2.1.7. Canales Multipunto.....	88
2.2.1.8. Medios de Transmisión .....	89
2.2.2. Modelos de Interconexión.....	98
2.2.3. Modelo OSI.....	103
2.2.4. Arquitectura TCP/IP.....	112
2.2.5. Pasado y Presente .....	117
2.2.5.1. Servicio de Modo Conexión y en modo sin Conexión .....	117
2.2.5.2. Direccionamiento.....	119
2.2.5.3. Ruteo.....	122
2.2.5.4. Servicio de Nombres .....	123
2.2.5.5. Suite del Protocolo TCP/IP.....	124
2.2.5.5.1. Aplicaciones y Utilidades .....	124
2.2.5.5.2. Protocolo de Control de Transferencia .....	125
2.2.5.5.3. Protocolo de Transferencia de Archivo Simple .....	127

2.2.5.5.4. Protocolo de Transferencia de Correo Simple .....	128
2.2.5.5.5. Protocolo TELNET .....	130
2.2.5.5.6. Protocolos de TCP/IP .....	131
2.2.5.5.7. Protocolo de Control de Transmisión (TCP) .....	131
2.2.5.5.8. Protocolo de Datagrama de Usuario (UDP).....	136
2.2.5.5.9. Protocolo de Internet (IP).....	137
2.2.5.5.10. Protocolo de Control de Mensaje de Internet IICMP).....	140
2.2.5.5.11. Protocolo de Resolución de Direcciones (ARP).....	140
2.2.5.5.12. Protocolo de Manejo de Red Simple (SNMP).....	141
2.2.5.5.13. Protocolo de Acceso a Red .....	141
2.2.5.5.14. Protocolo de Acceso a Subred .....	144
2.2.5.5.15. Una Red de Ejemplo de TCP/IP .....	144
2.3. Base de Datos Relacionales (Características, Ventajas y Desventajas).....	149
2.4. Modelo Cliente – Servidor	
2.4.1. Introducción .....	167
2.4.2. Conceptos.....	167
2.4.3. Ventajas.....	172
2.4.4. Arquitectura Cliente / Servidor .....	174
2.4.5. Cliente.....	180
2.4.6. Servidor .....	182
2.4.7. Modelo Cliente / Servidor de Dos y Tres Niveles.....	196
2.5. Metodologías de Ingeniería de Software Orientadas a Objetos (Ventajas y Desventajas).....	199
2.6. Conceptos básicos de PowerBuilder.....	220
3. Planteamiento del Problema y Propuesta de Solución	
3.1. Situación Actual.....	245
3.2. El usuario y sus requerimientos .....	271
3.3. Recopilación y Análisis de información .....	277
3.4. Planteamiento del Problema .....	293
3.5. Descomposición funcional.....	302
3.6. Opciones de solución .....	320

3.7. Elección de la solución óptima .....	325
<b>4. Desarrollo e Implementación del Sistema</b>	
4.1. BackEnd para cada módulo .....	353
4.1.1. Diagrama Entidad - Relación .....	353
4.1.2. Diagrama contextual .....	382
4.1.3. Diagramas de procesos .....	383
4.1.4. Diagramas de Flujo de Datos .....	403
4.1.5. Diagramas de eventos .....	409
4.1.6. Diagramas de estados .....	417
4.2. Generación de Código para el procesamiento de la información .....	425
4.3. Diseño y Construcción del FrontEnd .....	555
4.4. Diseño e Implementación de Rutinas de Diagnóstico y Evaluación de Datos .....	569
4.5. Pruebas en Red, factibilidad técnica y operativa .....	618
Conclusiones .....	623
Bibliografía .....	627
<b>Apéndices</b>	
A. Diagrama Entidad - Relación .....	A - 1
B. Manual del Usuario .....	B - 1
C. Manual Técnico .....	C - 1

## **INTRODUCCIÓN**

En los últimos años, la presencia de la Ingeniería en Computación en las actividades del ser humano ha sido esencial para alcanzar nuevos desarrollos. Al igual que en cualquier otra rama de la ingeniería, la Ingeniería en Computación realiza numerosas tareas para el beneficio de la sociedad.

Es prácticamente imposible imaginarse un banco, una fábrica, un laboratorio de investigación de cualquier ciencia o incluso una tienda de autoservicio, sin computadoras, vivimos en la era de la informática y es por ello que los ingenieros en computación tenemos el compromiso de atender todas aquellas demandas de la sociedad en materia de cómputo, la realización de la presente tesis es precisamente una muestra de dicho compromiso.

En la actualidad, toda persona física o moral (entidad) tiene la necesidad de tomar decisiones sobre la forma en que debe distribuir sus recursos. Para lograr este fin, se deben apoyar en la teoría contable, la cual refleja de una manera sistemática y estructurada, la información necesaria expresada en unidades monetarias. A través de un proceso cronológico, la contabilidad capta, clasifica, registra y resume con claridad, los eventos económicos identificables y cuantificables que realiza dicha entidad.

Lo anterior refleja que la contabilidad es la base principal para producir información financiera, que permita a los grupos de alta dirección de cualquier empresa, la toma de decisiones.

La contabilidad es una de las actividades del ser humano que más se ha apoyado en el uso de la computación, hoy en día es muy difícil encontrar a alguien que lleve la contabilidad de una empresa y que no se apoye en al menos una hoja de cálculo.

A lo largo del presente trabajo se explicarán los procedimientos que se llevaron a cabo para la realización del "Sistema para el Manejo Centralizado de la Contabilidad Nacional de una Empresa utilizando Sybase y PowerBuilder", con el cual se pretende dar solución a la necesidad de una empresa paraestatal con grandes volúmenes de información, utilizando las bondades que representan las redes de computadoras en el desarrollo de aplicaciones cliente/servidor.

El presente trabajo está dividido en tres capítulos, los cuales se detallan a continuación:

En el primer capítulo se establecen los fundamentos teóricos más importantes sobre los cuales se sustenta esta tesis, se abarcan desde los conocimientos contables necesarios para desarrollar un sistema contable, hasta los conocimientos técnicos que apoyan el desarrollo del sistema, como son el conocimiento de los sistemas operativos, las comunicaciones, las bases de datos, así como la teoría de la construcción de sistemas en ambiente cliente – servidor, además se hace énfasis en las metodologías de Ingeniería de Software orientadas a Objetos.

En el segundo capítulo se realizó un profundo estudio de la situación en la que se encontraba la empresa al momento de empezar el presente trabajo, se estudió la forma en que está estructurada la institución, se analizaron los requerimientos del usuario, se planteó el problema y se establecieron diversas alternativas de solución, tomando en

cuenta, las herramientas existentes en la empresa, así como en el mercado nacional, para pasar posteriormente a determinar cuál de ellas era la más viable.

En el tercer y último capítulo se expone la forma en que se desarrolló el sistema, se presentan los diagramas que sustentan el desarrollo orientado a objetos, así como el código. Al utilizar metodologías de desarrollo se garantiza la calidad del sistema, pero no obstante ello, en este último capítulo se muestran las pruebas que se realizaron una vez que el sistema estaba terminado, a fin de que la empresa y el usuario final tengan la seguridad que la herramienta que utilizan cubre plenamente sus expectativas.

Se cuenta con apéndices en los cuales se muestran el diagrama entidad - relación, el manual del usuario, y el manual técnico, documentos importantes para el buen funcionamiento del sistema, el primero se estructura de acuerdo a los módulos que conforman el sistema y el segundo es una herramienta de suma importancia que en caso de contingencia puede evitar grandes problemas.

El producto final obtenido, es un sistema que opera bajo ambiente Windows y que facilita la captura, el proceso y mantenimiento de toda la información contable generada en toda la red aeroportuaria del país. Asimismo, permite la generación de los principales reportes financieros.

## **OBJETIVO**

Desarrollar un sistema que automatice por completo el proceso operativo contable en una empresa paraestatal a nivel nacional, por medio de un sistema que cuente con características tales que proporcione un ambiente amigable al usuario, seguridad en el almacenamiento de la información, rapidez en el procesamiento de los datos y principalmente, uniformidad en los catálogos de toda la red aeroportuaria evitando incongruencias al momento de generar estados financieros.

Mejorar y Agilizar la captura de pólizas utilizando las ventajas que proporcionan las redes de computadas, permitiendo que cada usuario pueda verificar sus aplicaciones contables.

Lograr con el uso del sistema, la concentración y consolidación de toda la información contable generada por cada aeropuerto, minimizando los tiempos y trámites para obtener información al día.

## **1. MARCO TEÓRICO (CONTABILIDAD).**

### **1.1 CONCEPTOS BÁSICOS DE CONTABILIDAD.**

#### **1.1.1 Origen y Definición.**

Los inicios de la contabilidad se remontan a las antiguas civilizaciones Chinas, Griegas y Egipcias, en las cuales, sus dirigentes utilizaban registros de cuentas para conocer el costo de la mano de obra y de los materiales usados en la construcción de sus estructuras. Pero 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 Luca Pacioli 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.

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.

La evolución de todos los procedimientos de contabilidad se inició en los Estados Unidos a fines del siglo XIX y la primera mitad del siglo XX, época en que se lograron grandes adelantos, tanto en la filosofía de las cuentas, como en procedimientos de registro, en los cuales se tiene el auxilio de la tecnología que está a las órdenes de la contabilidad para implantar, reajustar o sofisticar sistemas completos, que proporcionen información financiera con la claridad, veracidad y oportunidad deseada.

Se ha confirmado entonces que toda sociedad ha tenido, tiene y tendrá necesidad de contar con información financiera para alcanzar sus metas.

Si definimos como *ENTIDAD*, la unidad económica social que se compone de un conjunto de recursos materiales y humanos para la obtención de fines determinados. Podemos decir entonces, que la *CONTABILIDAD* es una técnica que se utiliza para producir sistemática y estructuralmente, información cuantitativa expresada en unidades monetarias de las transacciones que realiza una entidad y de ciertos eventos económicos identificables que la afectan, con objeto de facilitar a los diversos interesados en la toma de decisiones en relación con dicha entidad.

### 1.1.2 Importancia y Utilidad .

La importancia de la contabilidad se debe a la cualidad que tiene de adecuarse al propósito del usuario. La utilidad de la información está en función de su contenido informativo y de su oportunidad.

El contenido informativo está basado en:

- *La significación de la información*, es decir, en su capacidad de representar simbólicamente (con palabras y cantidades) la entidad y su evolución, su estado en diferentes puntos en el tiempo y los resultados de su operación.
- *La relevancia de la información*, que es la cualidad de seleccionar los elementos de la misma que mejor permitan al usuario captar el mensaje y operar sobre ella para lograr sus fines particulares.
- *La veracidad*, cualidad esencial, pues sin ella se desvirtúa la representación contable de la entidad, que abarca la inclusión de eventos realmente sucedidos y de su correcta medición de acuerdo con las reglas aceptadas como válidas por el sistema.
- *La comparabilidad*, es decir, la cualidad de la información de ser válidamente comparable en los diferentes puntos de tiempo para una entidad y de ser válidamente comparables dos o más entidades entre sí, permitiendo juzgar la evolución de las entidades económicas.

La oportunidad de la información contable se refiere al hecho de que llegue a manos del usuario cuando éste pueda usarla para tomar sus decisiones a tiempo y así lograr sus fines.

Otras características de la información contable son:

- *La confiabilidad*, se refiere al hecho de que el usuario acepta y utiliza la información contable para tomar decisiones. Este crédito que el usuario da a la información está fundamentado en que el proceso de cuantificación, o sea la operación del sistema, es estable, objetiva y verificable. Estas tres características abarcan la captura de datos, su manejo (clasificación y cálculo), y la presentación de la información en los estados financieros.
- *La estabilidad del sistema*, indica que su operación no cambia en el tiempo y que la información que produce ha sido obtenida aplicando las mismas reglas. Esta característica no debe ser un freno a la evolución y perfeccionamiento de la información contable. Cualquier cambio que se haga y que sea de efectos importantes deber ser dado a conocer para evitar errores a los usuarios de la información.
- *La objetividad*, se refiere a que el sistema debe manejar la información de una forma objetiva, imparcial y veraz de tal manera que se obtenga la equidad de ésta y no afecte los intereses de los usuarios.
- *La verificabilidad*, la información debe ser procesada de tal manera que su contenido pueda ser comparado en cualquier momento, siguiendo las reglas de cuantificación establecidas.
- *La provisionalidad*, significa que la información contable no representa hechos totalmente acabados. La necesidad de tomar decisiones obliga a hacer cortes en la vida de la empresa para presentar los resultados de operación y la situación financiera con sus cambios, incluyendo eventos cuyos efectos no

terminan a la fecha de los estados financieros. Esta característica representa una limitación a la precisión de la información.

### 1.1.3 Principios de Contabilidad Generalmente Aceptados.

Los principios de contabilidad son enunciados que determinan las bases de observación obligatoria para los contadores públicos, y que afecta, tanto a los procedimientos de registro, como a las técnicas de información financiera.

La implementación de estos principios obedeció a que en el desarrollo de la contabilidad, la forma del registro de las operaciones y la información de las mismas, difería entre los contadores, lo que ocasionaba tener criterios heterogéneos al respecto que hacía difícil aun en una misma entidad, la comparación de la información de fechas distintas.

Paulatinamente los principios se fueron implementando dentro del esquema básico de la contabilidad, siendo probable que en el futuro se incorporen otros. Los principios de contabilidad vigentes en la actualidad son nueve agrupados de la siguiente manera:

➤ **Los principios relativos a la entidad y sus efectos financieros.**

1. *Principio de la Entidad.* La Entidad es la unidad económica social que se compone de un conjunto de recursos materiales y humanos para la obtención de fines determinados.
2. *Principio de la Realización.* La contabilidad cuantifica en términos monetarios las operaciones que realiza una entidad con otros participantes en la actividad económica y ciertos eventos económicos que la afectan. Las operaciones y eventos económicos que la contabilidad cuantifica, se consideran por ella realizados cuando:

- Se han efectuado transacciones con otros entes económicos.
- Se han tenido transformaciones internas que modifican la estructura de recursos o de sus fuentes.
- Han ocurrido eventos económicos externos a la entidad o derivados de las operaciones de ésta y cuyo efecto puede cuantificarse razonablemente en términos monetarios.

3. *Principio del Periodo Contable.* La necesidad de conocer los resultados de operación y la situación financiera de la entidad, que tiene una existencia continua, obliga a dividir su vida en periodos convencionales. Las operaciones y eventos así como sus efectos derivados, susceptibles de ser cuantificados, se identifican con el periodo en que ocurren; por tanto cualquier información contable debe indicar claramente el periodo a que se refiere. En términos generales, los costos y gastos deben identificarse con el ingreso que originaron, independientemente de la fecha en que se pague.

- **Principios que cuantifican las operaciones de las entidades y su presentación.**

4. *Principio del Valor Histórico Original.* Las transacciones y eventos económicos que la contabilidad cuantifica se registran según las cantidades de efectivo que se afecten o su equivalente o la estimación razonable que de ellos se haga al momento en que se consideren realizados contablemente. Estas cifras deberán ser modificadas en el caso de que ocurran eventos posteriores que les hagan perder su significado, aplicando métodos de ajuste en forma sistemática que preserven la imparcialidad y objetividad de la información contable. Si se ajustan las cifras por cambios en el nivel general de precios y se aplican a todos los conceptos susceptibles de ser modificados que integran los estados financieros, se considerará que no ha habido violación de este principio; sin embargo, esta situación debe quedar debidamente aclarada en la información que se produzca.

5. *Principio del Negocio en Marcha.* La entidad se presume en existencia permanente, salvo especificación, por lo que las cifras de sus estados financieros representarán valores históricos, o modificaciones de ellos, sistemáticamente obtenidos. Cuando las cifras representen valores estimados de liquidación, esto deberá especificarse claramente y solamente serán aceptados para información general cuando la entidad esté en liquidación.
6. *Principio de la Dualidad Económica.* Esta dualidad se constituye de:
- Los recursos de los que dispone la entidad para la realización de sus fines. y
  - Las fuentes de dichos recursos, que rara vez, son la especificación de los derechos que sobre los mismos existen, considerados en su conjunto.
- **Principio relativo a la información de las entidades.**
7. *Principio de la Revelación Suficiente.* La información contable presentada en los estados financieros debe contener en forma clara y comprensible todo lo necesario para juzgar los resultados de operación y la situación financiera de la entidad.
- **Principios que abarcan requisitos generales del sistema.**
8. *Principio de la Importancia Relativa.* La información que aparece en los estados financieros debe mostrar los aspectos importantes de la entidad susceptibles de ser cuantificados en términos monetarios. Tanto para efectos de los datos que entran al sistema de información contable como para la información resultante de su operación, se debe equilibrar el detalle y multiplicidad de los datos con los requisitos de utilidad y finalidad de la información.

9. *Principio de la consistencia.* Los usos de la información contable requieren que se sigan procedimientos de cuantificación que pertenezcan en el tiempo. La información contable debe ser obtenida mediante la aplicación de los mismos principios y reglas particulares de cuantificación para, mediante la comparación de los estados financieros de la entidad, conocer su evolución y mediante la comparación con estados de otras entidades económicas, conocer su posición relativa.

#### 1.1.4 Registro de Operaciones.

Para que los recursos de las entidades puedan ser administrados en forma eficaz, es necesario que sus operaciones sean controladas plenamente, requiriéndose establecer antes el proceso contable y cumplir con sus fases de: Sistematización, Valuación y Registro.

En la etapa de *Sistematización* del proceso contable, la entidad seleccionará el sistema de registro que se adapte a sus características y posibilidades. Lo anterior, debe estudiarse, ya que cada sistema ofrece ventajas y desventajas en cuanto a:

- Forma de presentación de los registros e información.
- Grado de detalle de la información.
- Grado de control sobre las operaciones.
- Costos de implantación y mantenimiento.
- Personal y equipo requerido para su implementación y funcionamiento.

En la etapa de *Valuación* el importe dado como valor a cada operación queda asentado en documentos que constituirán el antecedente o comprobante de la misma, como por ejemplo: facturas, notas de remisión, cheques, pagarés, recibos, letras de cambio, fichas de depósito, etc. Los documentos anteriores

reciben el calificativo de "comprobatorios" ya que dan fe de lo realizado y deben tener una serie de datos indispensables tanto para la entidad que realizó la operación, como para el Fisco que aceptará como deducibles de impuestos a aquellos que justifiquen gastos relacionados con el giro del negocio.

La tercera etapa que es el *Registro*, está constituida por los siguientes pasos:

- **Captura.** En este paso se requiere revisar que toda operación tenga su documento comprobatorio y que éste contenga los requisitos necesarios, tanto para la empresa como para el Fisco.
- **Clasificación de las Operaciones.** Estas deben ordenarse por tipo de transacción (ventas, compras, gastos, pagos, etc.), con la finalidad de hacer más ágil el registro de todas ellas, pudiendo incluso registrarse en una sola operación, varias transacciones de la misma naturaleza y condiciones.
- **Registro de Operaciones.** Este paso consiste en anotar los aumentos y disminuciones de los recursos, deudas, patrimonio, producto o gastos, que se tienen como consecuencia de ellas. El registro de las operaciones se controla en registros individuales llamados "cuentas", en las cuales, se lleva de manera ordenada, sistemática y cronológica los aumentos y disminuciones provocados por las operaciones realizadas.

#### **1.1.5 Clasificación de Entidades Económicas.**

A la contabilidad le interesa identificar la entidad que persigue fines económicos particulares y que es independiente de otras entidades. Se utilizan para identificar una entidad, dos criterios:

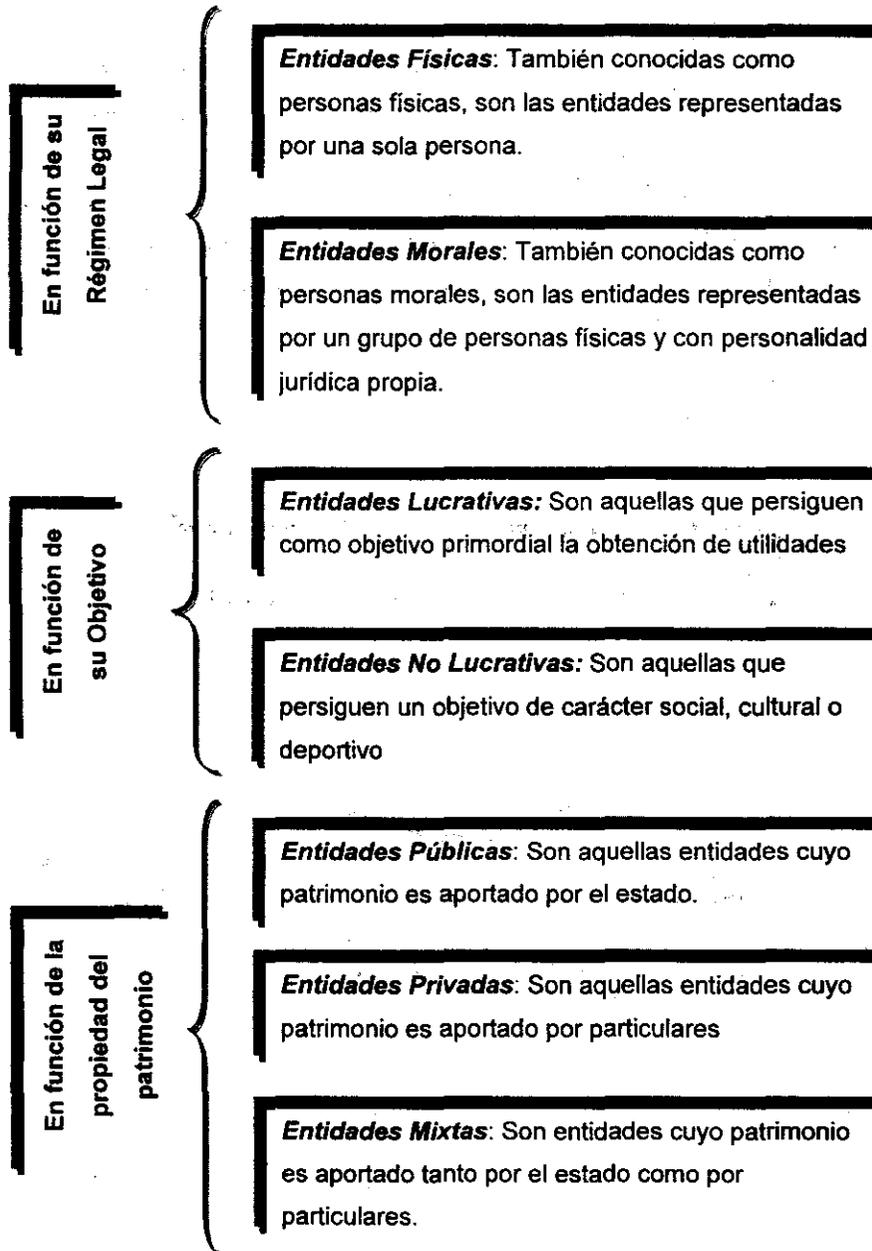


Figura 1.1.1 Clasificación de Entidades Económicas

1) conjunto de recursos destinados a satisfacer alguna necesidad social con estructura y operación propios, y 2) centro de decisiones independientes con respecto al logro de fines específicos, es decir, a la satisfacción de una necesidad social. Por tanto, la personalidad de un negocio es independiente de sus accionistas o propietarios y en sus estados financieros sólo deben incluirse los bienes, valores, derechos y obligaciones de este ente económico independiente. La entidad puede ser una persona física, una persona moral o una combinación de varias de ellas. Las entidades económicas se clasifican como se muestra en la figura 1.1.1.

#### 1.1.6 Ecuación Básica de la Contabilidad.

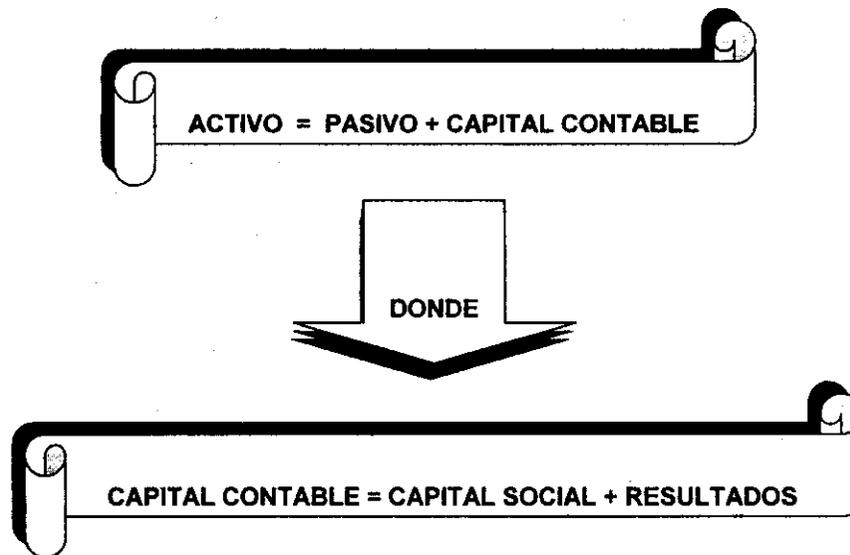


Figura 1.1.2 Ecuación Básica de la Contabilidad

### 1.1.6.1 Definición y Clasificación de Activo

Activo es el conjunto o segmento, cuantificable, de los beneficios económicos futuros fundadamente esperados y controlados por una entidad, representados por efectivo, derechos, bienes o servicios, como consecuencia de transacciones pasadas o de otros eventos ocurridos.

En resumen, el activo es el conjunto de bienes y derechos (recursos disponibles) que son propiedad de la entidad.

La clasificación del Activo se hace atendiendo al grado de "Disponibilidad" que tiene cada uno de sus elementos, entendiéndose por esto, la mayor o menor facilidad que tiene cada uno de ellos para convertir su valor a dinero. A mayor facilidad en este tipo de conversión, los activos tendrán mayor grado de disponibilidad; a mayor dificultad para lograr esta conversión, los activos tendrán menor grado de disponibilidad.

*El Activo Circulante* contiene a los recursos productores de ingresos de las entidades, y que por lo tanto poseen mayor grado de disponibilidad, ya que en el corto plazo (período no mayor a 1 año) convierten su valor a dinero. Como ejemplo de ellos tenemos: Caja, Bancos, Acciones y Valores a la Vista, Clientes, Documentos por Cobrar, Deudores Diversos, Mercancías o Almacén, Pagos anticipados a Corto Plazo, etc.

*El Activo no Circulante o Activo Fijo* concentra al resto de los recursos de una entidad, es decir, aquellos que presentan menor grado de disponibilidad (convertibilidad de su valor a dinero en un plazo superior a 1 año). Dentro de esta clasificación se tienen los bienes que la empresa adquiere con la finalidad de usar y no de comerciar, así como pagos anticipados cuyos beneficios se recibirán a través de varios años. Como ejemplo de estos recursos se tiene:

Documentos por cobrar a largo plazo, Terrenos, Edificios, Equipo de Oficina o Mobiliario, Equipo de Transporte, Maquinaria, Crédito Mercantil, etc.

*El Activo Diferido* son aquellas inversiones que con el transcurso del tiempo se convierten en gastos. Esta clase de activo no tiene una forma física pero se representa mediante documentos, contratos, acuerdos o descripciones, por ejemplo: Derechos de Autor, Patentes, Franquicias, etc.

#### **1.1.6.2 Definición y Clasificación de Pasivo.**

Pasivo es el conjunto o segmento, cuantificable, de las obligaciones presentes de una entidad particular, virtualmente ineludibles, de transferir efectivo, bienes o servicios en el futuro a otras entidades, como consecuencia de transacciones o eventos pasados.

En resumen, el pasivo es un conjunto de deudas y obligaciones que tiene una empresa.

La clasificación del pasivo se realiza atendiendo al grado de "exigibilidad" que tienen las deudas de la entidad. Existirá mayor grado de exigibilidad cuando el tiempo de que se disponga para el pago de las mismas, sea menor. Se tendrá menor grado de exigibilidad, cuando la entidad disponga de más tiempo para cubrir sus deudas.

*El Pasivo a Corto Plazo* es la clasificación que recibe también el nombre de Pasivo Circulante o Flotante, y concentra a aquellas deudas con mayor grado de exigibilidad, es decir, las de vencimiento menor a un año. Como ejemplos de estas obligaciones se encuentran: Proveedores, Préstamos Bancarios a Corto Plazo, Documentos por Pagar, Acreedores Diversos, Dividendos por Pagar, etc.

*El Pasivo a largo plazo* se denomina también Pasivo Fijo o Consolidado, y contempla a todas las obligaciones cuyo vencimiento supera el término de un año, es decir, aquellas que para la entidad tienen menor grado de exigibilidad. Dentro de esta clasificación tenemos los siguientes conceptos: Hipotecas por Pagar o Acreedor Hipotecario, Documentos por Pagar a Largo Plazo, Impuestos por Pagar, etc.

#### **1.1.6.3 Definición de Capital Contable.**

Capital Contable es el derecho de los propietarios sobre los activos netos que surge por aportaciones de los dueños, por transacciones y otros eventos o circunstancias que afectan una entidad y el cual se ejerce mediante reembolso o distribución.

El Capital Contable representa la totalidad del patrimonio de una entidad y se clasifica de la siguiente manera:

*El Capital Social* es la aportación inicial efectuada por los socios de la empresa, necesaria para comenzar las actividades de la misma.

*Las Reservas de Capital* son separaciones que legal o voluntariamente hacen de sus utilidades los propietarios o directivos de una entidad, con la finalidad de retener aquellas para un fin específico futuro.

*Las Utilidades o Pérdidas de Ejercicios Anteriores o Acumuladas*, son los efectos que han aumentado o disminuido al capital de la entidad, y que permanecen dentro de ella por no haberse repartido (las utilidades), o por no haberse cubierto aún (las pérdidas). La Utilidad o Pérdida del Ejercicio representa los aumentos o disminuciones que tiene el capital de la entidad, por haber realizado operaciones que la redituaron en un ejercicio mayores productos que gastos, o viceversa.

### 1.1.7 Principio de Dualidad Económica.

El Principio de Dualidad Económica se constituye por un lado de los recursos de que dispone la entidad para lograr sus fines y de las fuentes de dichos recursos, por el otro.

El Principio de Dualidad Económica o Partida Doble está basada en el principio de la lógica denominado "ley de causalidad" cuyo postulado consiste en que no hay causa sin efecto, y que un efecto tiene como origen una causa, pudiendo existir pluralidad en ambos sentidos (varios efectos provienen de una causa y viceversa).

Este principio científico como corolario establece la proporcionalidad entre causa y efecto, siendo evidente que a mayor acción de la causa corresponde mayor cantidad de efecto. Dentro de las empresas ocurren tres circunstancias:

- Que los hechos o causas determinantes están bien definidas, pues los constituye el movimiento de valores.
- Que las causas que el movimiento de valores implica, están precisamente cuantificadas y, por lo tanto, en los efectos se refleja esta cuantificación precisa, no alterándose la igualdad de la fórmula del balance.
- Que como los hechos u operaciones mercantiles se refieren a movimiento de valores circunscritos dentro de los capítulos del balance, en éstos deberán tener lugar las causas y producirse los efectos.

La Partida Doble debe cumplirse siempre; desde el asiento inicial hasta los estados financieros, pues de otra manera se rompería con la técnica contable y la información sería deformada.

### 1.1.8 Reglas del Cargo y el Abono.

Dentro de la técnica contable para cumplir debidamente con La Partida Doble En el registro de las operaciones, se aplican las siguientes reglas (figura 1.1.3).

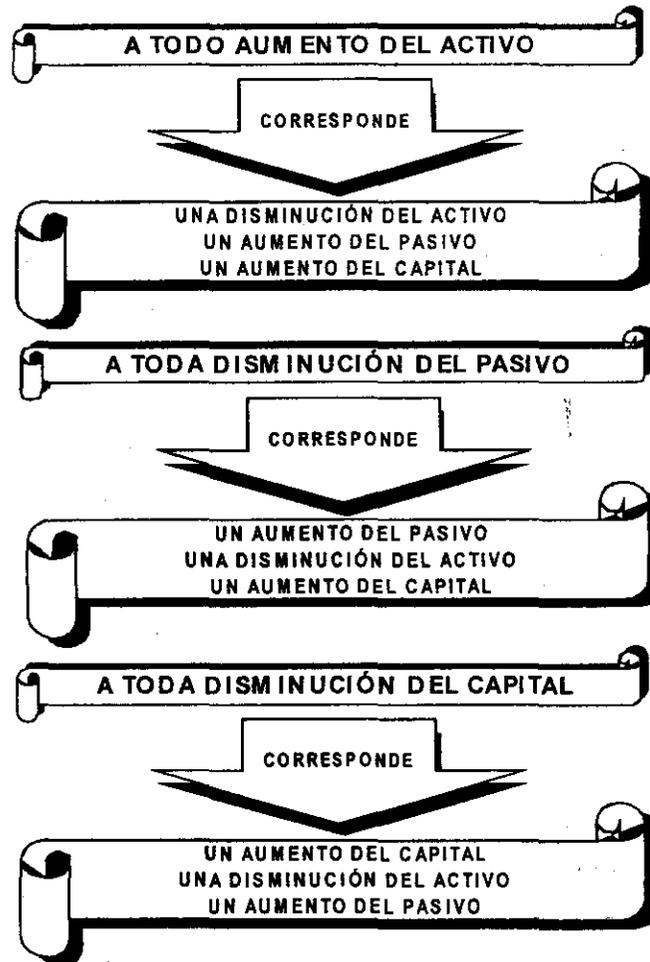


Figura 1.1.3 Reglas del Cargo y el Abono

### 1.1.9 Cuentas y Catálogo de Cuentas.

La cuenta es el instrumento del cual se vale la teoría contable para clasificar los diferentes conceptos de activo, pasivo, capital, ingresos y egresos que se afectan al realizarse una operación financiera. De hecho, la cuenta representa la descripción y el cómputo de los aumentos y disminuciones que presenta cada renglón o elemento que interviene en las operaciones que practica la entidad.

Dentro de la Contabilidad, se denomina CUENTAS a cada uno de los apartados en los cuales se lleva un registro, sea éste perteneciente al activo, pasivo, capital, ingresos o egresos.

Al conjunto de cuentas debidamente clasificadas se le denomina técnicamente "CATALOGO DE CUENTAS". Este catálogo se diseña e implanta en la primera etapa del proceso contable.

La nomenclatura que se le asigna a cada una de las cuentas debe indicar o ser representativa del concepto que trata de describir.

Para anotar en la cuenta los aumentos o las disminuciones que se produzcan, como consecuencia de las operaciones practicadas, la cuenta presenta dos partes iguales: A la parte izquierda se le denomina DEBE y a la derecha se le llama HABER.

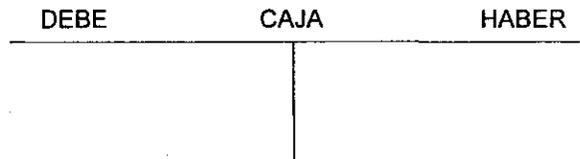


Figura 1.1.4 Elementos que integran una Cuenta.

A las anotaciones que se hagan dentro del DEBE (lado izquierdo) de cualquier cuenta, se le conoce con el nombre de CARGOS o DEBITOS. Las afectaciones que se hagan en el HABER (lado derecho) de las cuentas, se denominan ABONOS o CREDITOS.

A la suma de los cargos de cualquier cuenta, técnicamente se le conoce con el nombre de MOVIMIENTO DEUDOR; la suma de los abonos de las cuentas, se conoce con la denominación de MOVIMIENTO ACREEDOR.

Normalmente una cuenta presenta un movimiento mayor que otro, es decir, puede tener su movimiento deudor mayor que el acreedor (los cargos superan a los abonos) o viceversa (los abonos superan a los cargos). La diferencia aritmética entre ambos movimientos recibe el nombre de SALDO.

La clasificación de las cuentas según su Saldo es de dos tipos: SALDO DEUDOR o SALDO ACREEDOR (figura 1.1.5a). Existirá Saldo Deudor en una cuenta, cuando su movimiento deudor sea mayor a su movimiento acreedor. Existe Saldo Acreedor en una cuenta, cuando su movimiento acreedor supera al movimiento deudor.

Existen ocasiones en que una cuenta presenta igualdad en la suma de sus movimientos deudor y acreedor; en este caso, la diferencia aritmética entre ambas sumatorias es igual a CERO, por lo que se dice que la cuenta está SALDADA.

Las cuentas atendiendo a los estados financieros básicos de toda entidad, se clasifican en: Cuentas de Balance o de Situación Financiera y Cuentas de Resultados (figura 1.1.5b).

*Las Cuentas de Balance* o de Situación Financiera, son aquellas que controlan cualquier elemento del activo, pasivo y capital de las entidades. A su vez, estas cuentas presentan subdivisiones, que son:

- Cuentas de Activo Circulante
- Cuentas de Activo No Circulante
- Cuentas de Pasivo a Corto Plazo
- Cuentas de Pasivo a Largo Plazo
- Cuentas de Capital Contable

Las *Cuentas de Resultados* son aquellas que controlan los aumentos y disminuciones de cualquier elemento de los ingresos (ventas, productos financieros y Otros productos) y egresos de toda entidad (costo de ventas, gastos de ventas, gastos de administración, gastos financieros y otros gastos). Las cuentas de resultados se dividen en:

- Cuentas de Ingresos (las que generan utilidades, y por tanto, aumento en el capital contable)
- Cuentas de Egresos (las que provocan pérdidas, y por tanto, disminución en el capital contable).

Las cuentas que más frecuentemente se emplean dentro de la contabilidad comercial, agrupadas de acuerdo a la clasificación descrita anteriormente, se muestran a continuación:

#### A. CUENTAS DE BALANCE O DE SITUACIÓN FINANCIERA

##### A.1 Cuentas de Activo

##### A.1.1 Cuentas de Activo Circulante

- A.1.1.1 Caja
- A.1.1.2 Bancos
- A.1.1.3 Acciones y Títulos a la Vista
- A.1.1.4 Clientes
- A.1.1.5 Documentos por cobrar
- A.1.1.6 Deudores Diversos
- A.1.1.7 Mercancías o Almacén

A.1.1.8 Pagos Anticipados a Corto Plazo

A.1.1.9 IVA Acreditable

**A.1.2 Cuentas de Activo No Circulante**

A.1.2.1 Documentos por Cobrar a Largo Plazo

A.1.2.2 Terrenos

A.1.2.3 Edificios

A.1.2.4 Mobiliario y Equipo de Computo

A.1.2.5 Maquinaria

A.1.2.6 Primas de Seguros

A.1.2.7 Propaganda y Publicidad

A.1.2.8 Papelería y Artículos de Escritorio

**A.2 Cuentas de Pasivo**

**A.2.1 Cuentas de Pasivo a Corto Plazo**

A.2.1.1 Proveedores

A.2.1.2 Préstamos Bancarios a Corto Plazo

A.2.1.3 Documentos por Pagar

A.2.1.4 Acreedores Diversos

A.2.1.5 Anticipos de Clientes

A.2.1.6 Impuestos por Pagar

**A.2.2 Cuentas de Pasivo a Largo Plazo**

A.2.2.1 Hipotecas por Pagar a Acreedor Hipotecario

A.2.2.2 Préstamos Bancarios a Largo Plazo

A.2.2.3 Obligaciones por Pagar

A.2.2.4 Intereses Cobrados por Anticipado

A.2.2.5 Rentas Cobradas por Anticipado

**A.3 Cuentas de Capital Contable**

A.3.1 Capital Social

A.3.2 Reservas de Capital

A.3.3 Utilidades o Pérdidas de Ejercicios Anteriores

A.3.4 Utilidad o Pérdida del Ejercicio

## B. CUENTAS DE RESULTADOS

### B.1 De Ingresos

#### B.1.1 Ventas

#### B.1.2 Productos Financieros

#### B.1.3 Otros Productos

### B.2 De Egresos

#### B.2.1 Costo de Adquisición de lo Vendido o Costo de Ventas

#### B.2.2 Gastos de Venta

#### B.2.3 Gastos de Administración

#### B.2.4 Gastos Financieros

#### B.2.5 Otros Gastos

## 1.1.10 Libros Principales

### 1.1.10.1 Libro de Diario

Es utilizado, como su nombre lo indica, para registrar en él todas las operaciones obedeciendo un orden cronológico. Esto es, procesándolas en función a la fecha en que se susciten para lo cual hace acto de presencia el principio de contabilidad de realización, el cual manifiesta "Registrar las operaciones en el momento de su realización", al ser esta la característica básica del Libro Diario, este puede adquirir cualquier formato con la condición de que permita informar del registro cronológico de las operaciones.

En la figura 1.1.6 se muestra un ejemplo de un Libro Diario.

### 1.1.10.2 Libro de Mayor

A diferencia del Libro Diario, en el cual se registran todas las operaciones en forma cronológica sin importar la afectación continua de cuentas de Balance y

Resultados, surgiendo en última instancia una tarea interminable para determinar el saldo de las cuentas. Con el Libro Mayor se establece un registro para el control de cada cuenta.

En el Libro Diario después de 25 hojas de operaciones registradas para elaborar los estados financieros, sería tarea de extrema laboriosidad el determinar el saldo de las cuentas, en tanto que se tendría necesidad de localizar dentro de tales 25 hojas, los cargos y abonos de cada una de las cuentas.

El Libro de Mayor viene a solucionar esta situación, destinando cada una de sus hojas para controlar los cargos y abonos de cada cuenta (figura 1.1.7).

El formato del Libro de Mayor no es único, es flexible, pudiéndose optar por el más adecuado a las características de la entidad, siempre y cuando permita controlar en forma individual los cargos y abonos de cada cuenta, en función a las operaciones registradas en el Libro Diario.

#### **1.1.11 Libros Auxiliares**

Los datos que se muestran en el Libro Mayor son sintéticos y no bastan para dar toda la información necesaria, por lo tanto, para disponer de una información analítica, relacionada con los conceptos y valores que integran el saldo de las cuentas, la información del Libro Diario debe ser procesada en Libros Auxiliares de todas las cuentas deseadas.

Los libros auxiliares pueden ser en lo general, de dos formas:

➤ Con clasificación de columnas y registros sucesivos de detalle. En este tipo de libros se llevan los análisis de cuentas cuyas subdivisiones pueden preverse. Por ejemplo, en la cuenta "Gastos de Administración", de antemano

pueden precisarse los conceptos que interesa conocer como: Sueldos de Gerencia, Gastos de Viaje, Correo, Teléfonos, Papelería, Útiles de escritorio, Luz, Renta del Local, Transportes locales y Reparación de equipo entre otros.

➤ Con clasificación de subcuentas, por medio de tarjetas u hojas intercambiables. Existen varias cuentas como: Clientes, Proveedores, Deudores Diversos, Acreedores Diversos, etc., cuya expansión depende del mayor número de cuentas personales que se tenga que abrir. El auxiliar no sería posible llevarlo en forma tabular, así que se necesita una Subcuenta con las mismas características que la cuenta.

#### 1.1.12 Balanza de Comprobación.

La Balanza de Comprobación es el documento en que se relacionan todas y cada una de las cuentas utilizadas durante un periodo contable, indicando sus movimientos y saldos para verificar que el registro de las operaciones se efectúa conforme a la teoría de la Partida Doble:

- Verificar que se registraron cargos con sus correspondientes abonos.
- Verificar que los movimientos deudor y acreedor, determinados en cada cuenta, son aritméticamente correctos.
- Verificar que el saldo determinado en cada cuenta es aritméticamente correcto.

A continuación se muestra un resumen de la clasificación de las cuentas (figura 1.1.5a y b) y ejemplos de Libro Diario (figura 1.1.6), Libro Mayor (figura 1.1.7) y Balanza de Comprobación (figura 1.1.8).

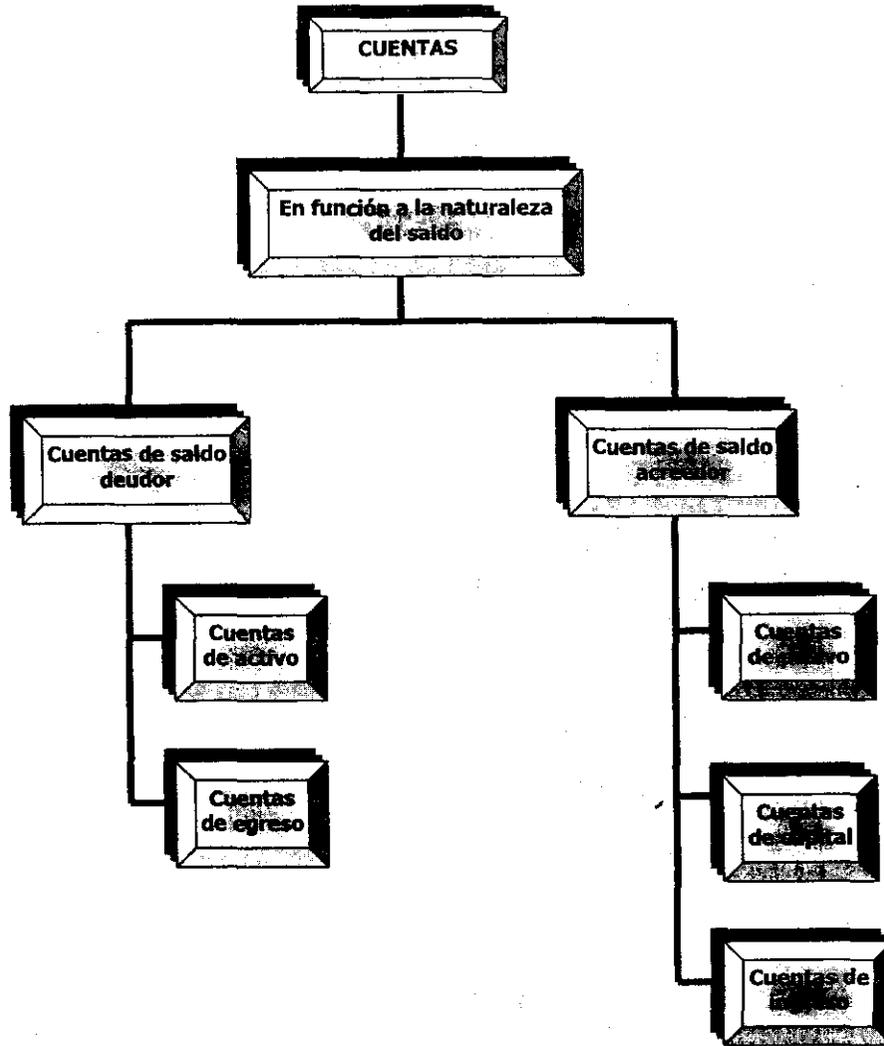


Figura 1.1.5 a. Clasificación de cuentas (según saldo)

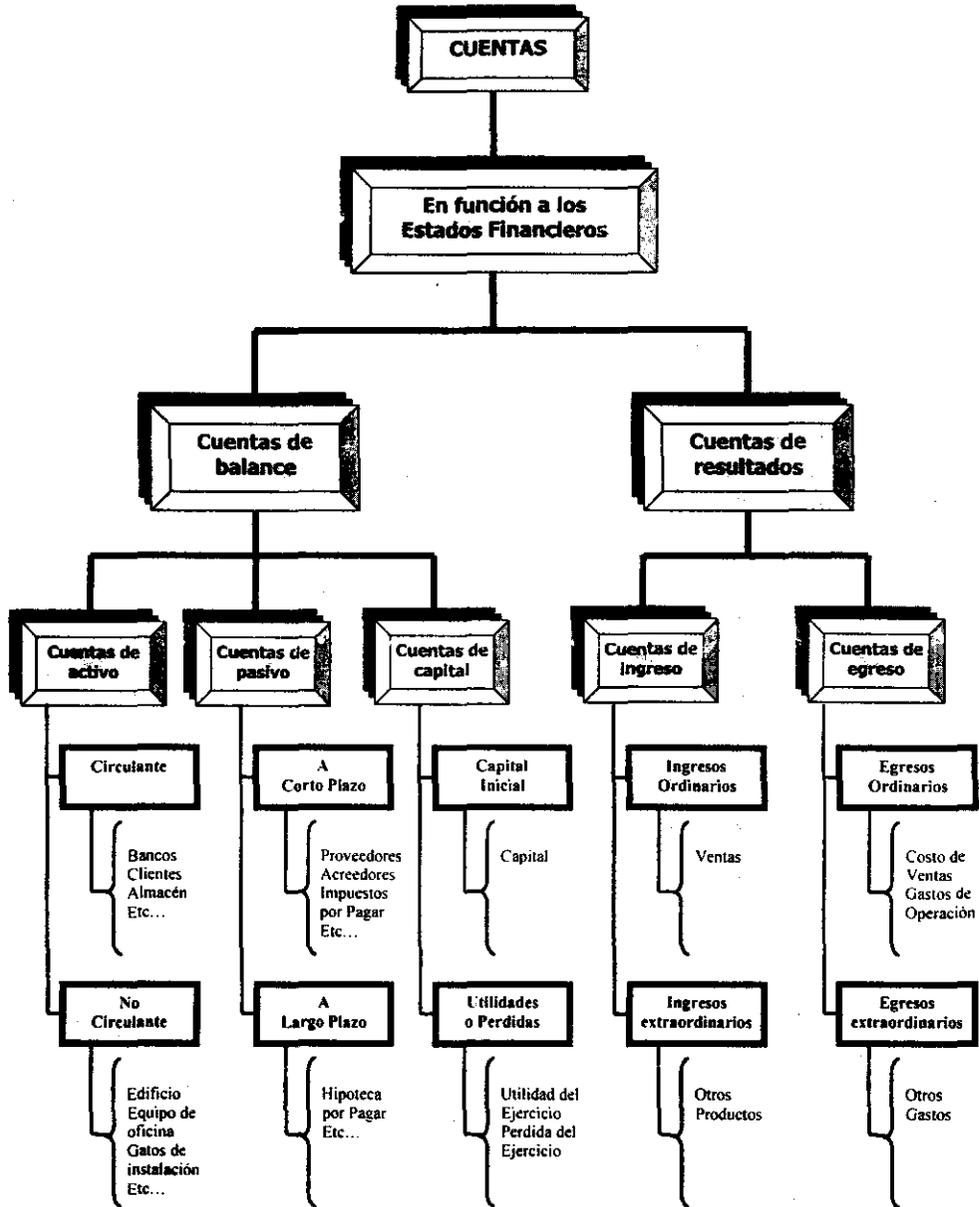


Figura 1.1.5 b. Clasificación de cuentas (estados financieros)

## ELECTRONICA ARIES

LIBRO DIARIO		CONCEPTO		DEBITOS		CREDITOS		VALORES	
FECHA	No. de entrada		ENTRADA DAS	SALIDA DAS	DE GENERAL	COSTO PROMEDIO	DEBE	HABER	SALDO
05-sep-98	1	COMPRA FACTURA 346	20	20	1000.00	1,000.00	20,000.00		20,000.00
10-sep-98	2	FLETES POR COMPRA F346		20		1,100.00	2,000.00		22,000.00
16-sep-98	3	COMPRA FACTURA 422	10	30	1250.00	1,150.00	12,500.00		34,500.00
20-sep-98	4	DEVOLUCION SFACT 346		5	25	1,180.00		5,000.00	29,500.00
03-sep-98	5	VENTA CON FACTURA 1		10	15	1,180.00		11,800.00	17,800.00
10-sep-98	6	COMPRA FACTURA 518	10	25	1100.00	1,180.00	11,000.00		28,700.00
15-sep-98	7	DEVOLUCION SFACT 518		1	24	1,180.00		1,100.00	2,700.00
20-sep-98	8	VENTA CON FACTURA 2		14	10	1,150.00		16,100.00	11,500.00
...									

Figura 1.1.6 Ejemplo de un Libro Diario

BANCOS		CAPITAL		PROVEEDORES	
(S)	\$120,000.00	\$2,000.00 (2)	\$120,000.00 (S)	(4)	\$5,000.00 \$20,000.00 (1)
		12,500.00 (3)		(7)	1,100.00 5,500.00 (6)
		5,500.00 (6)			
	\$120,000.00	\$20,000.00	\$120,000.00	\$6,100.00	\$25,500.00
	\$100,000.00		\$120,000.00		\$19,400.00

ALMACEN		CLIENTES		VENTAS	
(1)	\$20,000.00	\$5,000.00 (4)	(5)	\$18,000.00	\$18,000.00 (5)
(2)	2,000.00	11,800.00 (5a)	(8)	\$27,300.00	\$27,300.00 (8)
(3)	12,500.00	1,100.00 (7)			
(6)	11,000.00	16,100.00 (8a)			
	\$45,500.00	\$34,000.00	\$45,300.00		\$45,300.00
	\$11,500.00		\$45,300.00		45300.00

COSTO DE VENTAS	
(5a)	\$11,800.00
(8a)	16,100.00
	\$27,900.00
	\$27,900.00

Figura 1.1.7 Ejemplo de un Libro de Mayor

CUENTAS	MOVIMIENTOS		SALDOS	
	DEUDOR	ACREEDOR	DEUDOR	ACREEDOR
CAJA	8,580	1,780	6,800	
BANCOS	26,300	8,390	17,910	
ACCIONES Y VALS. A LA VISTA	1,000		1,000	
CLIENTES	4,000	2,400	1,600	
DOCUMENTOS POR COBRAR	13,000		13,000	
DEUDORES DIVERSOS	100		100	
ALMACEN	39,500	10,500	29,000	
RENTAS PAGS. POR ANTICIP. C/P	1,500		1,500	
TERRENOS	6,000		6,000	
EQUIPO DE OFICINA	7,000		7,000	
EQUIPO DE REPARTO	13,000		13,000	
PRIMAS DE SEGUROS	1,200		1,200	
PROVEEDORES	1,500	7,000		5,500
DOCUMENTOS POR PAGAR	200	16,000		15,800
ACREEDORES DIVERSOS		7,650		7,650
IMPUESTOS POR PAGAR		250		250
INTS. COBS. POR ANTICIPADO C/P		20		20
CAPITAL SOCIAL		56,000		56,000
UTILIDAD DEL PERIODO		2,000		2,000
VENTAS		24,200		24,200
COSTO DE VENTAS	10,300		10,300	
GASTOS DE VENTA	1,940		1,940	
GASTOS DE ADMINISTRACION	1,060		1,060	
GASTOS FINANCIEROS	20		20	
PRODUCTOS FINANCIEROS		10		10
<b>SUMAS IGUALES</b>	<b>136,200</b>	<b>136,200</b>	<b>111,430</b>	<b>111,430</b>

Figura 1.1.8 Ejemplo de Balanza de Comprobación

### 1.1.13 Estados Financieros.

#### 1.1.13.1 Objetivo y Utilidad

Los Estados Financieros son los documentos elaborados por una entidad económica, en los cuales se consignan datos valuados en unidades monetarias, referentes a la obtención y aplicación de recursos materiales.

Los Estados Financieros básicos son la parte esencial de la información financiera, la cual además de ser de primera necesidad para la gerencia, es la misma información que se presenta al usuario general, para que éste obtenga conclusiones sobre como está cumpliendo la empresa con sus objetivos financieros.

El objetivo de los Estados Financieros básicos será el de informar sobre la situación financiera de la empresa en cierta fecha y los resultados de sus operaciones, así como las variaciones en el capital contable.

Los Estados Financieros básicos deben servir para tomar decisiones de inversión. Los principales interesados al respecto serían aquellas personas que pudieran aportar financiamiento como capital o crédito. Este grupo estaría interesado en medir:

- La capacidad de crecimiento, estabilidad y redituabilidad de la empresa, para así asegurar la obtención de un rendimiento y la recuperación de su inversión.
- Aquilatar la solvencia y liquidez de la empresa, así como su capacidad para generar recursos. Aquí tendrían interés los diversos acreedores, principalmente quienes han otorgado préstamos, al igual que a los

propietarios que están interesados en medir el posible flujo de dinero hacia ellos derivado de los rendimientos.

➤ Evaluar el origen y las características de los recursos financieros del negocio, así como el rendimiento de los mismos. En esta área el interés es general, pues todos están interesados en conocer de qué recursos financieros dispone la empresa para llevar a cabo sus fines, cómo los obtuvo, cómo los aplicó y finalmente qué rendimiento han tenido o pueden esperarse de ellos.

En resumen se entiende que se satisface al usuario general de la información, si los estados financieros son suficientes para que una persona con adecuados conocimientos técnicos, pueda formarse un juicio sobre:

- El nivel de rentabilidad
- La posición financiera, que incluye su solvencia y liquidez
- La capacidad financiera de crecimiento
- El flujo de fondos

Las características que deben de cumplir los Estados Financieros son los mismos que corresponden a la información contable:

- Utilidad
- Confiabilidad
- Provisionalidad

Basándose en los principios de contabilidad generalmente aceptados, los estados financieros básicos son: Balance General, Estado de Resultados, Estado de Variaciones en el Capital Contable y Estado de Cambios en la Situación Financiera en base a Efectivo.

### 1.1.13.2 Balance General

El balance General relaciona todos los activos, pasivos y capital de una entidad, valuados en unidades monetarias, a una fecha determinada, por lo general al final de un mes o de un año.

También se le conoce como Estado de Situación Financiera porque refleja la situación financiera del negocio al momento de su preparación (cómo esta la entidad en un momento específico), por lo que, si se preparará un Balance General al día siguiente, no presentaría exactamente la misma situación.

El Balance General debe de contener los siguiente datos:

- Encabezado
  - Nombre de la Empresa
  - Indicación de qué trata el Balance General
  - Fecha de formulación
  
- Cuerpo
  - Nombre y saldo de cada una de las cuentas que forman el activo
  - Nombre y saldo de cada una de las cuentas que forman el pasivo
  - Nombre y saldo de cada una de las cuentas que forman el capital contable
  
- Firmas
  - Del contador que lo hizo y autorizó
  - Del propietario del negocio

El Balance General puede elaborarse de dos formas: en forma de Cuenta (figura 1.1.9a) o, en forma de reporte (Figura 1.1.9b)

ACTIVO		PASIVO	
CIRCULANTE	14,241,000.00	A CORTO PLAZO	4,492,000.00
CAJA	791,000.00	PROVEEDORES	2,992,000.00
BANCOS	950,000.00	ACREEDORES DIVS.	1,500,000.00
CLIENTES	1,500,000.00	A LARGO PLAZO	900,000.00
DOC. POR COB.	1,000,000.00	DOCTOS. POR PAG. L/P	900,000.00
MERCANCIAS	10,000,000.00	SUMA DEL PASIVO	5,392,000.00
No. CIRCULANTE	3,655,000.00	<b>CAPITAL</b>	
EPO. DE OFICINA	1,755,000.00	CAPITAL	12,504,000.00
EPO. DE REPARTO	1,900,000.00		
<b>SUMA DEL ACTIVO</b>	<b>17,896,000.00</b>	<b>SUMA P. Y C</b>	<b>17,896,000.00</b>

C.P. ELSA CASAS SANCHEZ  
CONTADOR.

ACTIVO		
CIRCULANTE:		14,241,000.00
CAJA	791,000.00	
BANCOS	950,000.00	
CLIENTES	1,500,000.00	
DOCUMENTOS POR COBRAR	1,000,000.00	
MERCANCIAS	10,000,000.00	
NO CIRCULANTE:		3,655,000.00
EQUIPO DE OFICINA	1,755,000.00	
EQUIPO DE REPARTO	1,900,000.00	
<b>TOTAL DEL ACTIVO</b>		<b>17,896,000.00</b>
<b>PASIVO</b>		
A CORTO PLAZO:		4,492,000.00
PROVEEDORES	2,992,000.00	
ACREEDORES DIVERSOS	1,500,000.00	
A LARGO PLAZO:		900,000.00
DOCTOS. POR PAGAR A L/P	900,000.00	
<b>TOTAL PASIVO</b>		<b>5,392,000.00</b>
<b>CAPITAL CONTABLE</b>		<b>12,504,000.00</b>
<b>SUMA PASIVO Y CAPITAL</b>		<b>17,896,000.00</b>

C.P. ELSA CASAS SANCHEZ  
CONTADOR.

Figura 1.1.9 Ejemplos de Balanza General  
a) En Forma de Cuenta ; b) En Forma de Reporte

### 1.1.13.3 Estado de Resultados

El Estado de Resultados o Estado de Pérdida y Ganancias, es el estado financiero que muestra la utilidad o pérdida obtenida en las operaciones practicadas por una entidad en un periodo determinado. Dicho de otra manera, es el documento que muestra en forma ordenada y sistemática, los ingresos y egresos de una entidad en un periodo cualquiera, obtenidos como consecuencia de las operaciones realizadas por ella y mostrando al final del mismo, el efecto positivo (utilidad) o negativo (pérdida).

Los elementos que integran el Estado de Resultados son:

- Los Ingresos; cualquier operación practicada por la entidad cuyos efectos aumenten al capital contable, es decir, se tipifican como tales, todas las operaciones que generan productos y consecuentemente, utilidades.
- Los Egresos; cualquier disminución del capital contable de una entidad, originado por sus operaciones realizadas. De acuerdo con lo anterior, se entiende que serán egresos aquellas operaciones que generen costos y gastos que provocarían pérdidas para la entidad, y que por éstas, su patrimonio se vea disminuido.

Las cuentas consideradas como ingresos de una entidad, son:

- Las Ventas
- Los Productos Financieros
- Otros Productos

De la misma manera, los apartados conceptuales como egresos, son:

- EL Costo de Ventas

- Los Gastos de Ventas
- Los Gastos de Administración
- Los Gastos Financieros
- Otros Gastos

Al igual que el estado de situación financiera, el Estado de Resultados tiene una presentación definida, la cual puede ser Condensada (figura 1.1.10a) o Analítica (figura 1.1.10b). La primera, contempla en detalle a todos los elementos u operaciones que intervienen en un periodo determinado para obtener la utilidad o pérdida que ahí se muestra; la segunda en cambio, muestra de manera breve las operaciones que originaron el resultado presentado.

<b>EL SOL S.A.</b>			
<b>ESTADO DE RESULTADOS DEL 1° AL 31 DE DICIEMBRE DE 1998</b>			
	<b>VENTAS</b>		<b>5,450,000.00</b>
<b>MENOS:</b>			
	<b>COSTO DE VENTAS</b>		<b>1,480,000.00</b>
		<b>UTILIDAD BRUTA</b>	<b>3,970,000.00</b>
	<b>GASTOS DE OPERACIÓN</b>		
	GASTOS DE VENTAS	1,140,000.00	
	GASTOS DE ADMINISTRACIÓN	1,340,000.00	2,480,000.00
		<b>UTILIDAD EN OPERACIÓN</b>	<b>1,490,000.00</b>
	GASTOS FINANCIEROS	93,000.00	
	PRODUCTOS FINANCIEROS	67,000.00	26,000.00
	<b>UTILIDAD DESPUÉS DE GASTOS Y PRODUCTOS FINANCIEROS</b>		<b>1,464,000.00</b>
	OTROS PRODUCTOS	30,000.00	
	OTROS GASTOS	26,000.00	4,000.00
		<b>UTILIDAD DEL EJERCICIO *</b>	<b>1,438,000.00</b>
* Antes de Impuestos			
	C.P. ELSA CASAS SANCHEZ CONTADOR		ING. JERONIMO VALDIVIA GERENTE GENERAL

Figura 1.1.10 a) Ejemplo de un Estado de Resultados en forma condensada

<b>EL SOL S.A.</b>				
<b>ESTADO DE RESULTADOS DEL 1° AL 31 DE DICIEMBRE DE 2005</b>				
<b>VENTAS TOTALES</b>				
MENOS: DEVOLUCIONES SOBRE VENTAS	120,000.00			
REBAJAS SOBRE VENTAS	30,000.00	150,000.00		
<b>VENTAS NETAS</b>		<b>5,450,000.00</b>		
INVENTARIO INICIAL		2,000,000.00		
COMPRAS	1,100,000.00			
MAS: GASTOS DE COMPRAS	50,000.00			
<b>COMPRAS TOTALES</b>	<b>1,150,000.00</b>			
MENOS: DEVOLUCIONES SOBRE COMPRAS	120,000.00			
REBAJAS SOBRE COMPRAS	50,000.00	170,000.00		
<b>COMPRAS NETAS</b>		<b>980,000.00</b>		
MERCANCIAS DISPONIBLES PARA VENDER		2,980,000.00		
MENOS: INVENTARIO FINAL		1,500,000.00		
<b>COSTO DE VENTAS</b>		<b>1,480,000.00</b>		
<b>UTILIDAD BRUTA</b>		<b>3,970,000.00</b>		
<b>GASTOS DE OPERACIÓN</b>				
<b>GASTOS DE VENTA:</b>				
/SUELDOS DEL PERSONAL	950,000.00			
/LUZ, TELEFONO, ETC.	120,000.00			
/COMISIONES A AGENTES VENDEDORES	70,000.00	1,140,000.00		
<b>GASTOS DE ADMINISTRACIÓN:</b>				
/SUELDOS DEL PERSONAL	880,000.00			
/SERVICIOS DIVERSOS A LAS OFICINAS	300,000.00			
/HORAS EXTRA	120,000.00			
/PRESTACIONES AL PERSONAL ADMINISTRATIVO	40,000.00	1,340,000.00		
<b>TOTAL DE GASTOS DE OPERACIÓN</b>		<b>2,480,000.00</b>		
<b>UTILIDAD EN OPERACIÓN</b>		<b>1,490,000.00</b>		
<b>MENOS: GASTOS FINANCIEROS</b>				
/INTERESES PAGADOS	80,000.00			
/COMISIONES PAGADAS A BANCOS	13,000.00	93,000.00		
<b>MAS: PRODUCTOS FINANCIEROS:</b>				
/INTERESES COBRADOS	45,000.00			
UTILIDAD EN CAMBIO DE MONEDA EXTRANJERA	22,000.00	67,000.00		
<b>GASTOS FINANCIEROS NETOS</b>		<b>26,000.00</b>		
<b>UTILIDAD DESPUES DE GASTOS Y PRODUCTOS FINANCIEROS</b>		<b>1,464,000.00</b>		
<b>MAS: OTROS PRODUCTOS:</b>				
/UTILIDAD EN VENTA DE DESPERDICIOS	30,000.00	30,000.00		
<b>MENOS: OTROS GASTOS</b>				
/PERDIDA EN VENTA DE ACTIVO FIJO	20,000.00			
DONATIVOS	6,000.00	26,000.00		
<b>OTROS PRODUCTOS NETES</b>		<b>4,000.00</b>		
<b>UTILIDAD DEL EJERCICIO *</b>		<b>1,468,000.00</b>		
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border-top: 1px solid black; text-align: center;">C.P. ELSA CASAS SANCHEZ CONTADOR</td> <td style="width: 50%; border-top: 1px solid black; text-align: center;">LIC. RAFAELA S. REYES GERENTE GENERAL</td> </tr> </table>			C.P. ELSA CASAS SANCHEZ CONTADOR	LIC. RAFAELA S. REYES GERENTE GENERAL
C.P. ELSA CASAS SANCHEZ CONTADOR	LIC. RAFAELA S. REYES GERENTE GENERAL			
* ANTES DE IMPUESTOS				

Figura 1.1.10 b) Ejemplo de Estado de Resultados en forma analítica

#### **1.1.13.4 Estado de Variación en el Capital Contable**

El Estado de Variación en el Capital Contable, es un documento que muestra las modificaciones que han tenido las inversiones de los propietarios de una entidad en un periodo determinado.

#### **1.1.13.5 Estado de Cambios en la Situación Financiera en Base a Efectivo**

Es también conocido con el nombre de Estado de Flujo de Efectivo, y es un documento que informa sobre el manejo del efectivo (obtención y aplicación), por parte de una entidad durante un periodo determinado, mostrando al mismo tiempo un resumen de los cambios efectuados en la situación financiera (recursos y obligaciones).

## 2. MARCO TEÓRICO (COMPUTACIÓN).

### 2.1 SISTEMAS OPERATIVOS (CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS).

#### 2.1.1. UNIX

##### Introducción

El sistema operativo de una máquina se puede entender como el director de tráfico de todos los componentes que la forman. Es un conjunto de programas que controlan y organizan las actividades de la computadora.

El sistema operativo UNIX fue desarrollado en los laboratorios BELL, los cuales constituyen uno de los mejores grupos de investigación en el mundo. En 1969, los laboratorios BELL tenían una computadora General Electric modelo 645, que contaba con un sistema operativo llamado MULTICS.

MULTICS no cumplía con las características que requiere un equipo de programadores, por lo que un grupo de investigadores de los laboratorios se dio a la tarea de crear un entorno de programación que facilitara sus labores internas de investigación y desarrollo, y el resultado fue UNIX.

UNIX fue desarrollado por Ken Thompson, con el apoyo de Dennis Ritchie y otros. Esta primera versión estaba escrita en lenguaje ensamblador para una computadora PDP-7 de DEC. Al desarrollar un sistema operativo en lenguaje ensamblador lo hace dependiente de la máquina en la que se trabaja, Ken Thompson creó un lenguaje de programación llamado BCPL el cual fue mejorado por Dennis Ritchie llamándole "C" y reescribieron UNIX en "C" utilizando una PDP-9, rompiendo así con la tradición de escribir sistemas operativos en lenguaje ensamblador.

- UNIX soporta los lenguajes más importantes de programación, incluyendo Pascal, "C", COBOL, FORTRAN, etc.
- Existe una serie de asistentes de programación, desde un reordenador de código "C", hasta un generador de PARSER y SCANNERS para compiladores.
- En UNIX hay compatibilidad entre archivos y procesos, lo cual permite al sistema operativo tratar de la misma manera un archivo en disco que una terminal.
- UNIX cuenta con comunicación entre procesos, lo cual reduce la necesidad de uso de archivos intermedios para procesos encadenados (pipeline).

### Elementos del Sistema Operativo

UNIX cuenta con tres partes básicas, que son:

*KERNEL*: Es la parte del sistema que administra los recursos de la computadora, supervisa todos los dispositivos (discos, cintas, impresoras, terminales, líneas de comunicación, etc.). Controla la ejecución de procesos, asigna espacio en memoria, etc.

*Sistema de archivos* También conocido como File System, es la estructura organizadora de datos. Provee los medios para organizar los datos en forma compleja.

*Intérprete de Comandos* También conocido como Shell, propiamente es un programa de utilería, es la parte del sistema que el usuario ve, "escucha" al usuario y traduce sus requerimientos en acciones del KERNEL y utilerías.

## Definiciones Básicas

Como ocurre dentro de todo ambiente de trabajo, se han establecido ciertas definiciones y se ha creado un "argot" tanto en el medio de la computación como en UNIX mismo. A continuación presentamos algunos términos más comúnmente utilizados:

### Archivo (file)

Unidad de datos almacenada en un medio magnético normalmente. El usuario se refiere a los archivos por su nombre.

### Directorio(directory)

Archivo de tipo especial, que no contiene información relevante para el usuario. Sirve para agrupar archivos dentro de la jerarquía del sistema de archivos. Se puede decir que un directorio contiene una serie de archivos.

### Directorio Base (HOME directory)

Directorio en el que se encuentra el usuario al entrar al sistema.

### Directorio Raíz (ROOT directory)

Es el directorio que se encuentra en la más alta jerarquía del sistema, es decir, que no hay un directorio que lo contenga.

### Directorio de Trabajo (Working directory)

Directorio en el que se encuentra el usuario en un momento dado. Mientras aquel no cambie de directorio. Su dirección de trabajo será el mismo directorio HOME.

### Entrada/Salida Estándar

Cada proceso de UNIX puede enviar o leer información de tres partes, llamadas "entrada estándar", "salida estándar" y "error estándar", (standard

input, standard output y standard error), todas están asociadas con la terminal (a menos de que haya redirección).

#### Grupo (Group)

A efectos de desarrollar un trabajo común, los usuarios pueden agruparse. Los miembros de un grupo pueden tener prerrogativas de las que no gozan los que no pertenecen a él.

#### Login

Se habla de hacer un "login" cuando se entra al sistema

#### Nodo-i (i-node)

El nombre real de un archivo para el sistema operativo. Aunque este no se manifieste al usuario, el sistema no identifica los archivos por su nombre, lo hace mediante su nodo-i, un número que representa la localización de los archivos en el disco.

#### Nombre de Trayectoria (pathname)

Especificación completa de la localización de un archivo, conteniendo todos los directorios por encima de él, ya sea desde la raíz (root) o desde el directorio de trabajo (working directory).

#### Pipeline

Línea de comandos encadenados de manera que la salida de uno sirve de entrada del siguiente.

#### Super Usuario (super user)

Es normalmente el usuario que tiene el poder absoluto sobre el sistema. Es el administrador del sistema, es decir, tiene el poder para correr programas de mantenimiento, montar sistemas de archivos, manipular archivos y procesos de cualquier usuario. Requiere de un password y su nombre generalmente "root".

### Redirección

El proceso de desviar la entrada o salida estándar de datos de un comando a otro lugar (un archivo) especificado por el usuario.

### Usuario (user)

Persona habilitada por el sistema con un directorio propio, un grupo y capacidad de utilizar el sistema. Tiene un nombre asignado dentro del sistema para hacer "login".

### Estructura del Sistema de Archivos

Dentro de UNIX, el sistema de archivos está organizado en forma jerárquica, parece un árbol invertido, el cuál tiene un directorio raíz ("root"). A continuación ilustramos un sistema de archivos:

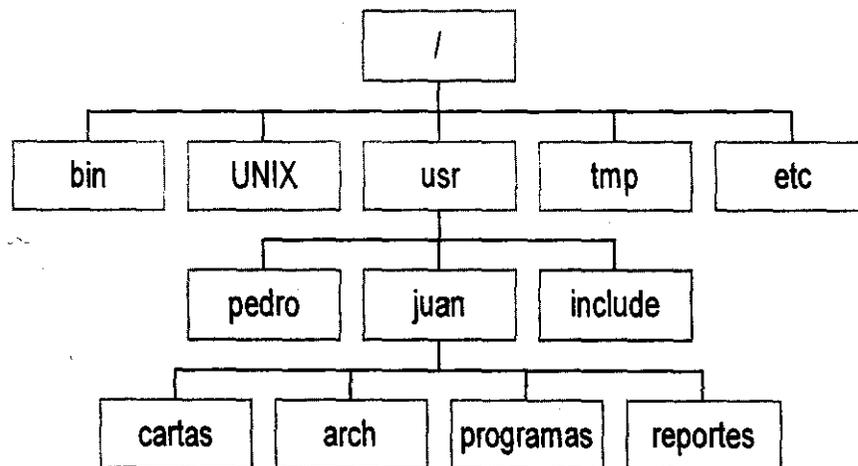


Figura 2.1.1.2 Sistema de Archivo de UNIX

## Tipos de Archivos

Desde el punto de vista del sistema operativo, existen cuatro tipos básicos de archivos:

*Archivos normales:* archivos que contienen datos. Incluyen archivos de datos propiamente dichos, ejecutables, texto, programas fuente, etc. UNIX no presupone estructura en los archivos, los considera un arreglo unidimensional de bytes.

*Directorios:* Contienen referencias a los archivos que se encuentran dentro de ellos.

*Dispositivos de carácter:* Todo tipo de periféricos que intercambian datos un byte a la vez. Incluyen terminales, impresoras, lectores ópticos, líneas de comunicación, etc.

*Dispositivos de bloques:* Periféricos que intercambian datos en bloques de varios bytes, normalmente 512. Incluyen discos duros, floppies, unidades de cinta, etc.

Los dos últimos tipos reciben el nombre genérico de "archivos especiales". Se encuentran en el directorio /dev. Un comando "ls -l" permite observar el tipo de archivo, con solo revisar el primer carácter de cada registro, correspondiendo:

- \_ Archivo normal
- d Directorio
- b Archivo especial de dispositivo de bloque
- c Archivo especial de dispositivo de carácter

## Directorios Importantes

Dentro de todo sistema de archivos UNIX se encuentran siempre una serie de directorios creados especialmente para dar una visión lógica. Estos son:

`/bin`

Este es el directorio donde se encuentra la mayoría de los comandos de UNIX

`/dev`

Este es el directorio de los archivos especiales como terminales, impresoras, memoria, discos flexibles, etc.

`/etc`

Contiene los programas de mantenimiento accesibles solo al super-usuario, así como archivos con información esencial para el sistema.

`/lib`

Contiene las bibliotecas objeto de los compiladores

`/usr/include`

Es para los archivos encabezado del compilador de "C".

`/tmp`

Este directorio se utiliza para archivos temporales creados durante la ejecución de los procesos.

`/usr`

Alberga generalmente a los usuarios (en algunos sistemas existe un directorio especial para ello, por ejemplo, `/usr/users`)

`/usr/bin`

Contiene utilerías de UNIX que son de nivel básico, así como los programas que no son propiamente de UNIX, como hojas de cálculo, procesadores de palabra, etc.

`/usr/lib`

En este directorio están los archivos con datos importantes para otros programas organizados en subdirectorios.

`/usr/spool`

Contiene una serie de directorios que albergan los archivos que van de la memoria a ser transmitidos a otros dispositivos y que se manejan en colas como la impresora, el correo electrónico, etc.

## 2.1.2 Windows NT

### La Familia de Sistemas Operativos de Microsoft

La familia de sistemas operativos de Microsoft incluye Windows 95, Windows NT *Workstation* (Estación de Trabajo) y Windows NT *Server* (Servidor). La mayoría de ambientes de trabajo pueden beneficiarse de uno o más de estos sistemas operativos. Existen ciertas similitudes entre estos sistemas operativos, así como también importantes diferencias:

#### Windows 95

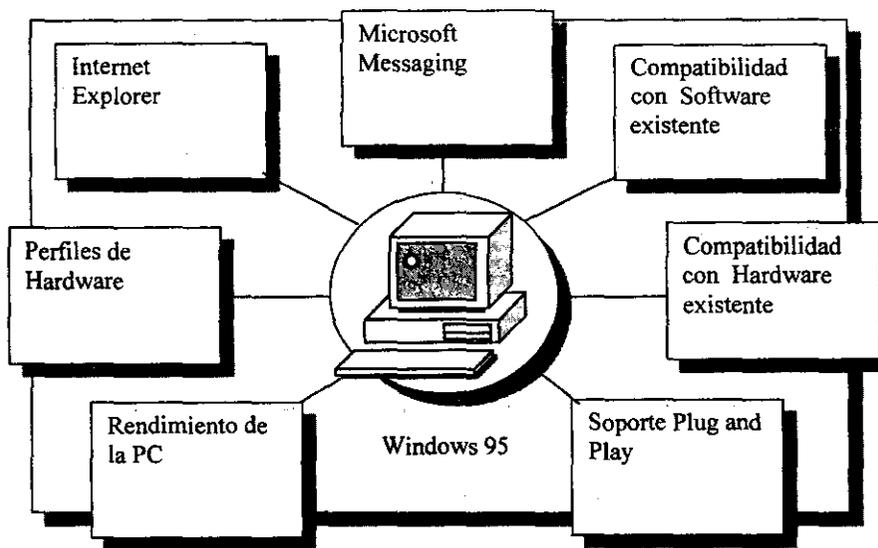


Figura 2.1.2.1 Sistema Operativo Windows 95

El Sistema Operativo Windows 95 está diseñado para proveer una plataforma de 32 bits. Windows 95 provee las siguientes características:

- Rendimiento de la PC: Soporta multitarea para aplicaciones basadas en Microsoft Win32 y Microsoft MS-DOS-. Windows 95 no soporta múltiples procesadores.
- Perfiles de hardware: Crea y mantiene una lista de configuraciones de hardware para satisfacer las necesidades específicas de cada computadora.
- Internet Explorer: Provee de un rápido y fácil de usar buscador que es compatible con los estándares actuales.
- Microsoft Messaging: Recibe y almacena correo electrónico, incluyendo archivos y objetos creados en otras aplicaciones.
- Compatibilidad con el Software existente: Corre aplicaciones de Windows de 32 y de 16 bits, así como aplicaciones basadas en MS-DOS. Esto incluye aplicaciones que accesan directamente el hardware, o aquellas que dependen de diseños no documentados de sistemas operativos previos.
- Compatibilidad con el hardware existente: Corre en la mayoría de computadoras personales de uso actual, incluyendo hardware que tiene varios años de antigüedad. Los requisitos mínimos de hardware para Windows 95 son los siguientes:
  1. Computadora personal con un procesador 386 DX/20 o mayor
  2. 4 Megabytes (MB) de RAM (se recomiendan 8 MB)
  3. 40 MB de disco duro libre (típicamente)
  4. Soporte de *Plug and Play* (Conectar y Usar): Provee poderoso soporte para laptops y para sistemas de escritorio.

### Windows NT Workstation (Windows NT para estaciones de trabajo)

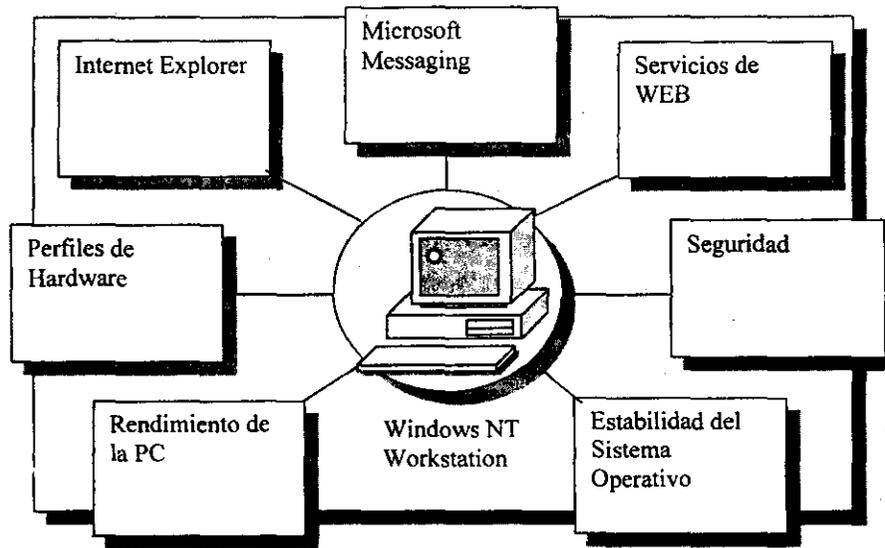


Figura 2.1.2.2 Windows NT Workstation

Windows NT Workstation puede ser usado solo como un sistema operativo de escritorio, en red en un ambiente de trabajo punto a punto, o utilizado como una estación de trabajo en un ambiente de Windows NT Server. Windows NT incluye las siguientes características, algunas de las cuales son también parte de Windows NT Server:

- Rendimiento de la PC: Soporta multitareas para todas las aplicaciones; soporta múltiples procesadores para lograr esto.
- Perfiles de Hardware: crea y mantiene una lista de configuraciones de hardware para satisfacer las necesidades específicas de cada computadora.

- Internet Explorer: provee de un rápido y fácil de usar buscador que es compatible con los estándares actuales.
- Microsoft Messaging: recibe y almacena correo electrónico, incluyendo archivos y objetos creados en otras aplicaciones.
- Servicios de Web: Provee de un servidor de Web personal, optimizado para correr en Windows NT Workstation.
- Seguridad: Provee seguridad local para archivos, carpetas, impresoras y otros recursos. El usuario debe estar autenticado por la estación de trabajo local o por controlador de dominio para acceder cualquier recurso en la computadora o en la red.
- Estabilidad del Sistema Operativo: Soporta cada aplicación en su propio espacio de dirección de memoria. Aplicaciones con un mal funcionamiento pueden no afectar otras aplicaciones o al sistema operativo.

Los requisitos mínimos de hardware para Windows NT Workstation son:

- Computadora personal con un procesador 486/DX/33 o mayor
- 12 MB de RAM (recomendados 16 MB)
- 120 MB de espacio de disco duro libre

### **Comparando Windows 95 con Windows NT Workstation**

Windows 95 y Windows NT Workstation comparten muchas características que los hacen muy efectivos para ser sistemas operativos de muchos ambientes de negocios. También existen diferencias importantes entre ellos: Windows 95 está diseñado para ser el camino más fácil para un sistema operativo de escritorio de 32 bits, mientras que Windows NT Workstation está diseñado para ser el más poderoso escritorio de 32 bits. Los consumidores necesitan decidir cuál es el sistema operativo más adecuado para sus necesidades. En muchos casos, una mezcla de ambos sistemas operativos puede ser lo mejor para los requerimientos de las compañías.

### **Características comunes**

Windows 95 y Windows NT Workstation tienen muchas características en común, que incluyen las siguientes:

- Interfaz de usuario común
- Utilidades comunes, tales como Internet Explorer y Microsoft Messaging
- Características comunes, como políticas del sistema y perfiles de hardware

### **Diferencias Técnicas**

Los siguientes puntos resumen estas diferencias:

- Windows 95 tiene menos requisitos de hardware demandantes que Windows NT Workstation, amplias aplicaciones y compatibilidad de dispositivos, y una instalación simplificada. Windows 95 ofrece multitarea robusta, y soporte móvil a través de APM (Manejador de Poder Avanzado) y Plug and Play.
- Windows NT Workstation tiene mayor desempeño, mayor confiabilidad y más seguridad que Windows 95.

## Windows NT Server

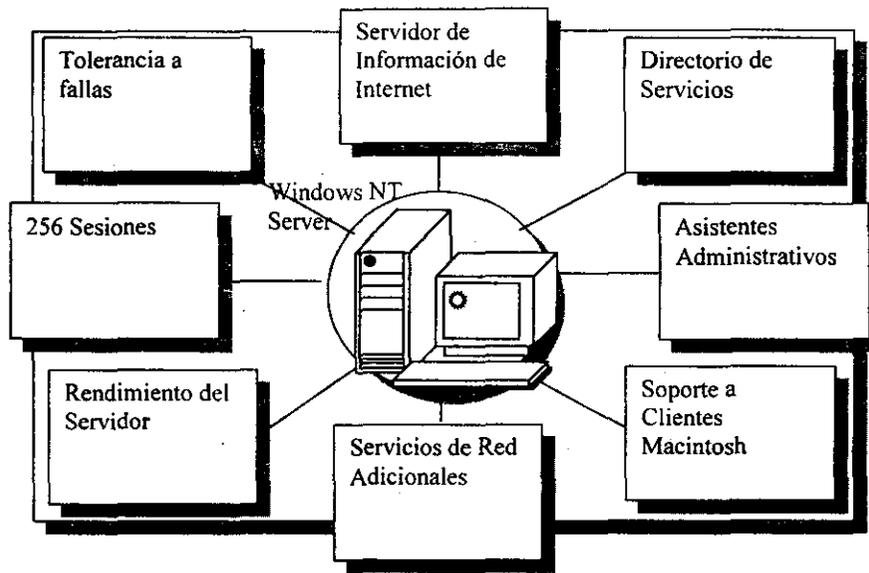


Figura 2.1.2.3 Windows NT Server

Windows NT Server ha sido optimizado para ser un servidor de archivos, impresoras y aplicaciones que puede manejar tareas para organizaciones muy diferentes, que van de pequeños grupos de trabajo hasta redes de empresas.

Windows NT Server es el componente preferido por muchas de las más poderosas herramientas de Microsoft, además ofrece las siguientes ventajas:

- **Desempeño del Servidor:** Puede ser configurado como servidor de archivos, de impresión o de aplicación. La menor versión de Windows NT Server soporta hasta 4 procesadores en paralelo en un ambiente de multiproceso simétrico. Las implementaciones de los Fabricantes de Equipo Original (OEM) de Windows NT Server soportan hasta 32 procesadores en un ambiente de multiproceso simétrico.

- Hasta 256 Sesiones de RAS: Provee soporte hasta para 256 sesiones de RAS.
- Tolerancia a fallas: Soporta tecnología RAID para protección de datos.
- Servidor de Información de Internet: Provee una plataforma rápida, poderosa y segura para ofrecer servicios HTTP, FTP y GOPHER.
- Ayudantes Administrativos: Incluye ayudantes que ayudan a los administradores a desempeñar tareas comunes.
- Soporte para Clientes Macintosh: Puede proveer servicios para compartir archivos o impresoras para clientes Macintosh.
- Servicios de Red Adicionales: Incluye ruteo Multi-protocolo, DNS, DHCP y WINS.
- Servicio de Directorio de Windows NT: Maneja un directorio de bases de datos seguro y distribuido y provee servicios para usuarios finales y para administradores de la red.

Los requisitos de hardware mínimos son:

- Procesador 486/DX2
- 16 MB de RAM
- 130 MB de espacio de disco duro libre

### **Arquitectura de Windows NT**

Para poder comprender el funcionamiento de Windows NT empezaremos con el entendimiento de la arquitectura del sistema. A continuación se explican los aspectos de las arquitecturas de Windows NT Workstation y Windows NT Server que afectan la funcionalidad y el desempeño de estos sistemas operativos.

**Modo Usuario vs. Modo Kernel**

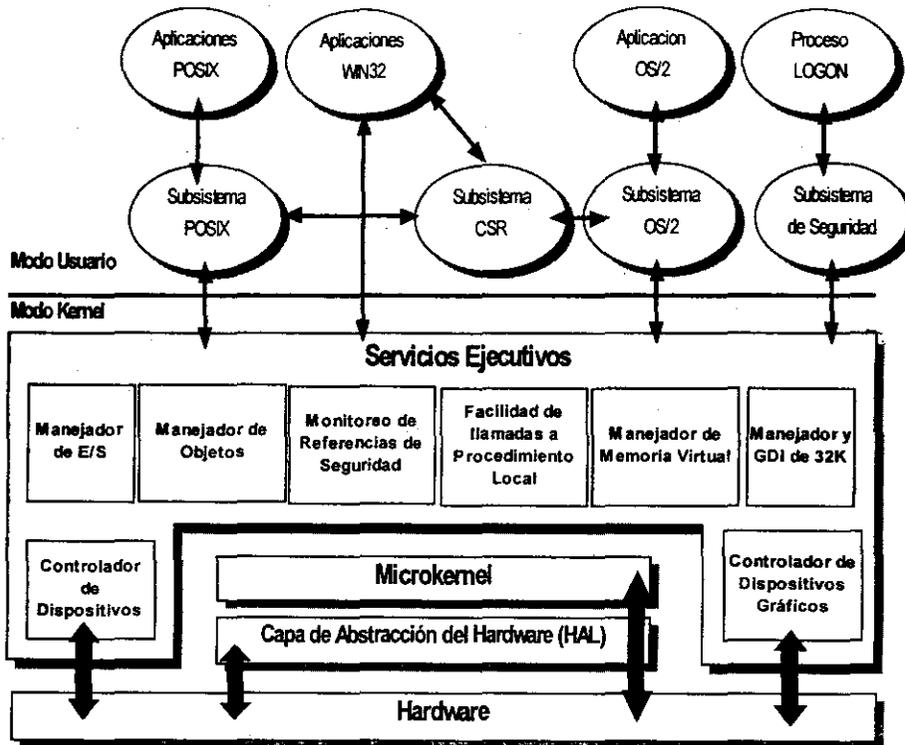


Figura 2.1:2.4 Modos de Operación de Windows NT

Windows NT utiliza dos modos de operación, que son: *Modo de Usuario* y *Modo Kernel* para mantener la eficiencia operativa y la integridad del sistema.

*Modo de Usuario*

Las aplicaciones, y los sistemas que los soportan, corren en modo de usuario. El proceso de este tipo de modo tiene las siguientes limitantes:

- No existe el acceso directo al hardware
- Limitado por un espacio de direcciones asignado
- Puede necesitar utilizar espacio de disco duro a modo de RAM virtual
- Procesa con una prioridad menor que los componentes de Modo Kernel

Este tipo de modo no procesa directamente los recursos. Las peticiones para acceder estos recursos deben ser concedidos por un componente del Modo Kernel. Esto provee protección contra aplicaciones en mal funcionamiento o acceso no autorizado de usuarios.

Los procesos tienen una prioridad baja, y generalmente tienen menos acceso a los ciclos del CPU que los que corren en modo Kernel.

#### *Modo Kernel*

Dado que los Servicios Ejecutivos de Windows NT centralizan a todas las funciones mayores del sistema operativo, es importante protegerlos de las aplicaciones del Modo de Usuario y sus subsistemas. Lo anterior se logra con los Servicios Ejecutivos, los cuales corren en Modo Kernel, el cual provee acceso a toda la memoria de la computadora y corre en un área de memoria reservada contra cualquier otra aplicación.

Únicamente los componentes en Modo Kernel pueden acceder recursos directamente.

#### **Servicios Ejecutivos**

Windows NT trabaja bajo las aplicaciones de usuario y el medio ambiente de los subsistemas para proveer soporte para las aplicaciones. La mayoría de esos componentes del sistema operativo son Servicios Ejecutivos.

**Componentes**

La siguiente tabla muestra los Servicios Ejecutivos y sus funciones:

<b>SERVICIO EJECUTIVO</b>	<b>FUNCIÓN</b>
Manejadores (Managers)	Varios módulos que manejan I/O, Objetos, Seguridad, Procesos, Comunicación Interprocesos (IPC), Memoria Virtual y Ventanas y Manejadores gráficos
Microkernel	Provee los servicios más básicos del sistema operativo, tales como la calendarización de tareas y el manejo de interrupciones.
Capa de Abstracción de Hardware (HAL)	Código que aísla la mayoría de las diferencias de interfaces de hardware de Windows NT, haciéndolo más portable
Manejadores de Dispositivos	Módulos que controlan el acceso al hardware

## El Modelo de Memoria de Windows NT

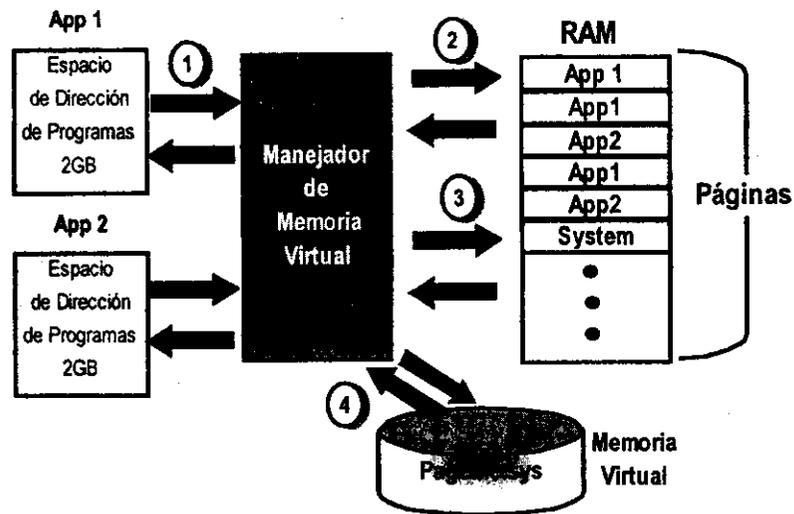


Figura 2.1.2.5 Modelo de Memoria de Windows NT

La arquitectura de memoria de Windows NT es un sistema de requerimiento de página virtual. Está basado en un espacio de direcciones de 32 bits lineal, que permite a cada proceso en Windows NT poder acceder hasta 4GB de memoria.

Con *Memoria Virtual*, todas las aplicaciones parecen tener todos los rangos de memoria completa disponible. Windows NT hace esto dando a cada aplicación un rango de memoria privada llamado *espacio de memoria virtual* y mapeando ésta hacia memoria física.

Windows NT mapea direcciones de memoria física y virtual en bloques de 4K llamados *páginas*. Cada segmento de memoria virtual tiene espacio para 4 GB de direcciones.

Dado que muy pocos sistemas contienen RAM suficiente para proveer 4GB para cada aplicación, el sistema operativo localiza las páginas de RAM físicas disponibles entre los espacios de memoria virtual. Mientras la aplicación opera como si tuviera su propio espacio de memoria de RAM, en realidad cada de las páginas en el espacio virtual de memoria puede tener una de las siguientes características:

- La mayoría de las páginas están actualmente vacías, porque la aplicación no las está utilizando
- Las páginas que están siendo utilizadas están redirigidas a un apuntador, invisible a la aplicación, hacia RAM física.
- Algunas páginas que no han sido usadas recientemente contienen otro apuntador, también invisible a la aplicación, a una sección de 4K de archivo de paginación en disco duro.

#### *Proceso de Memoria Virtual*

El proceso de manejar cuales páginas de memoria virtual son almacenadas en RAM, y cuales son almacenadas en el archivo de paginación es llamado *demanda de paginación*.

El procedimiento de demanda de paginación de Windows NT sigue los siguientes pasos:

- Una aplicación intenta almacenar datos en memoria
- El Manejador de Memoria intercepta la petición, determina qué tantas páginas son necesarias para completar la petición y mapea memoria física no usada a cualquier espacio de direcciones vacío en el espacio de memoria virtual de la aplicación. Este manejador esconde la organización de memoria física a los "ojos" de la aplicación. Cuando una aplicación llama para pedir una

localidad de memoria, mapea hacia una dirección de memoria que no cause conflicto.

- Si no existe suficiente espacio de memoria física sin usar, el manejador de memoria virtual utiliza demanda de paginación para encontrar páginas de 4K de RAM que no hayan sido utilizadas recientemente y entonces copia sus datos hacia el archivo de paginación (Pagefile.sys) en el disco duro.
- Cuando se necesitan los datos almacenados en el archivo de paginación, las páginas son copiadas a la RAM. La nueva localidad de RAM es mapeada de regreso a la misma dirección virtual requerida por la aplicación.

### **Ventajas de la Memoria Virtual**

Este esquema de direccionamiento lineal ayuda a hacer a Windows NT portable al ser compatible con el direccionamiento de memoria de procesadores tales como MIPS R4000 y DEC Alpha AXP.

Esta arquitectura contrasta en forma con el esquema de segmentación usado en los sistemas operativos de Windows de 16 bits, en ellos la memoria es rota en muchos segmentos de tamaño variable, cada uno con una longitud máxima de 64K. Este límite de 64K puede generar dificultades cuando se crean aplicaciones grandes, debido al uso ineficiente de los recursos del sistema.

Windows NT hace un uso de memoria fácil, más eficiente y más consistente. Como resultado de ello, permite a los programadores el escribir más grandes y más concisas aplicaciones. Ello también permite a los usuarios el correr más aplicaciones al mismo tiempo que no podría lograrse con tal solo la memoria física de un sistema.

## Grupos de Trabajo y Dominios

Una computadora basada en Windows NT opera en grupos de trabajo o en dominios.

### *Grupos de Trabajo*

Un grupo de trabajo es un agrupamiento lógico de computadoras y usuarios. Como parte de un grupo de trabajo, una computadora basada en Windows NT tiene su propio directorio de base de datos. Los recursos y las cuentas de usuario son manejadas en cada computadora. Los grupos de trabajo trabajan bien para un número relativamente pequeño de computadoras que comparten recursos con usuarios en otras computadoras.

### *Dominios*

Como un grupo de trabajo, un dominio es un agrupamiento lógico de computadoras y usuarios. Sin embargo, en un dominio, todas las computadoras comparten un directorio de bases de datos central que almacena información acerca de las cuentas de usuario y la seguridad para el dominio. El directorio de bases de datos es manejado por uno o más Servidores Windows NT Controladores de Dominio.

Como miembro de un dominio, una computadora basada en Windows NT puede compartir recursos, pero el dominio provee de un acceso centralizado para administración y mantenimiento de las cuentas de usuario.

### Modelo de trabajo en grupo

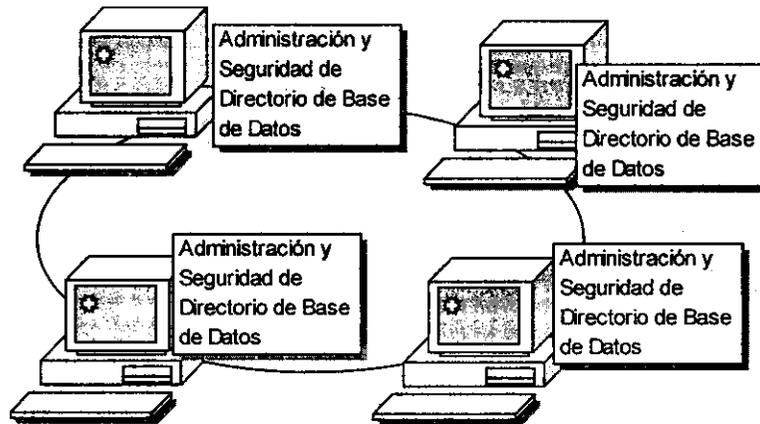


Figura 2.1.2.6 Modelo de Trabajo en Grupo

El modelo de trabajo en grupo es un arreglo de red de computadoras en el cuál los recursos, administración y seguridad están distribuidos a través de la red. Cada computadora tiene sus propias políticas de cuentas, administración y seguridad.

Un grupo de trabajo puede contener tanto Estaciones de Trabajo de Windows NT como Servidores Windows NT. La administración de cada computadora se encuentra separada de la administración de las otras computadoras en el grupo de trabajo. Cada computadora requiere a alguien para crear y mantener sus propias cuentas de usuario.

El modelo de trabajo en grupo tiene las siguientes ventajas y desventajas:

VENTAJAS	DESVENTAJAS
No requiere un servidor controlador de dominio de Windows NT	Manejo de cuentas no centralizado
Diseño e implementación simple	Ineficiente para redes que tienen muchas estaciones de trabajo
Conveniente para un número limitado de estaciones de trabajo en proximidad cercana	Control limitado de administración centralizado

**El Modelo de Dominio**

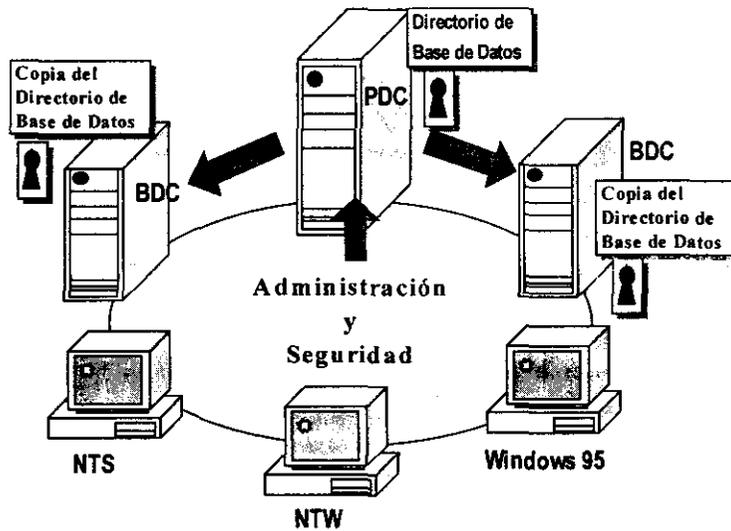


Figura 2.1.2.7 Modelo de Dominio

Un dominio es la unidad administrativa del Servicio de Directorio de Windows NT, en el cual la administración y la seguridad están centralizadas. Existe un directorio de base de datos común de la información de las cuentas y de la seguridad. En un dominio, las estaciones de trabajo y los servidores desempeñan las siguientes tareas:

- Obtener la validación de las cuentas del directorio de base de datos
- Permitir acceso a los recursos a los usuarios definidos en el directorio de base de datos
- Funciona como una parte de la administración centralizada de grupo

### **Servicios de Directorio**

El servicio de directorio de Windows NT (NTDS) maneja un directorio de base de datos seguro y provee servicios tanto para usuarios finales como para los administradores de la red. Para los usuarios, NTDS provee servicios de autenticación que permiten a cada uno de ellos tener un ID propio y un password que puede ser tecleado desde cualquier estación en la red para acceder a los servicios, aplicaciones y recursos localizados en cualquier lugar de la red. Para los administradores, NTDS provee manejo gráfico y servicios de seguridad que simplifican la creación y el mantenimiento de los usuarios y sus derechos para poblaciones de usuarios que van desde grupos de trabajo y hasta toda una empresa.

Contrariamente a los servicios de directorio con estructura de árbol, los cuales fuerzan a cada usuario en un grupo a colocarse al final de sus ramas, NTDS emplea una arquitectura de dominio más flexible que fácilmente acopla a las empresas, en las cuales cada persona es simultáneamente miembro de diferentes grupos, de acuerdo a su localidad, gerente, labores u otras funciones.

### Dominios

En un dominio, las computadoras basadas en Windows NT Server trabajan ya sea como controladores de dominio o como servidores stand-alone. Cada dominio mantiene su propio directorio de base de datos. Un administrador necesita crear una cuenta de usuario únicamente una vez en el directorio de base de datos en el Controlador de Dominio Primario (PDC). Cuando los usuarios se conectan al dominio, el controlador del dominio valida la conexión. El controlador de dominio verifica en el directorio de base de datos el nombre de usuario, password y sus restricciones.

Cuando un recurso se encuentra compartido en Windows NT, los permisos pueden ser asignados a las cuentas de usuarios que existen en la base de datos de cuentas de usuarios del dominio. La siguiente tabla resume las ventajas y desventajas de un dominio.

VENTAJAS	DESVENTAJAS
Cuentas centralizadas y seguridad en la administración	Requiere al menos un servidor controlador de dominio de Windows NT
Conexión de usuarios sencilla para acceso a los recursos de la red	Requiere planeación adicional por departamento o área

### Partición de Discos

El Setup de Windows NT puede ser utilizado para dividir discos duros en particiones. A las particiones les son asignados sus propias letras para denominarlos, además permiten la separación del sistema operativo y los datos de los usuarios en unidades lógicas.

### Partición del Sistema

Windows NT requiere de una *partición del sistema* para almacenar los archivos del hardware necesarios para bootear Windows NT. En un equipo basado en un procesador Intel, la partición activa, usualmente el drive C, es la partición del sistema. En un equipo basado en un procesador RISC, un programa de configuración de Firmware puede ser utilizado para diseñar la partición del sistema.

### Partición de Boot

Windows NT también requiere una *partición de boot* para almacenar los archivos del sistema operativo. La partición de boot debe cubrir el tamaño mínimo requerido. Si componentes de software adicionales son requeridos, el tamaño de esta partición puede ser mayor. Si Windows NT es instalado en la partición del sistema, esta partición también es la partición de boot.

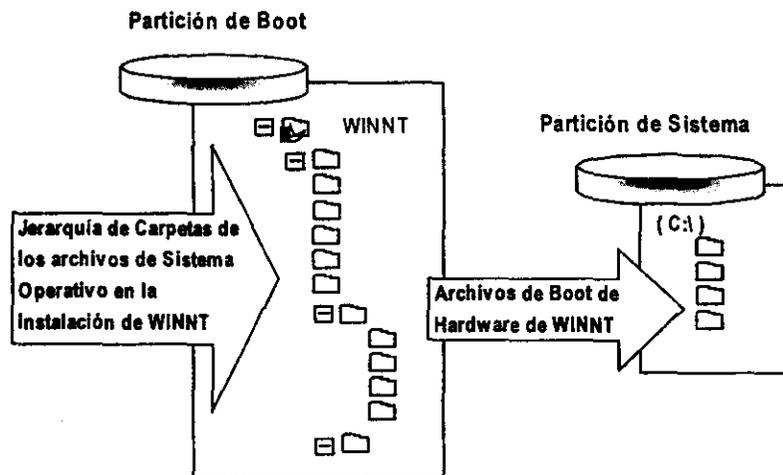


Figura 2.1.2.8 Particiones de Discos.

### Seleccionando un Sistema de Archivos (File System)

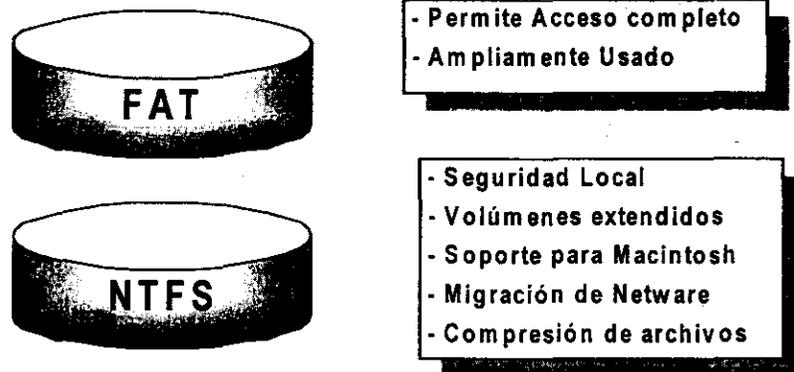


Figura 2.1.2.9 Sistema de Archivos

Antes de cualquier instalación de Windows NT hay que decidir el tipo de sistema de archivos a utilizar. Windows NT soporta los siguientes tipos:

#### *FAT (Tabla de Localización de Archivos)*

El sistema de archivos tipo FAT permite acceso de sistemas operativos como Windows NT, Windows 95, MS-DOS y OS/2. Para bootear entre Windows NT y otro sistema operativo, la partición del sistema debe estra formateda con este tipo de sistema de archivos.

Windows NT no soporta el sistema de archivos FAT32 de Windows 95.

#### *NTFS (Sistema de Archivos de Windows NT)*

Obviamente, NTFS requiere Windows NT, si la computadora es booteada bajo un sistema operativo diferente, este sistema no podrá ser capaz de acceder las particiones de NTFS.

Hay que escoger NTFS cuando:

- Windows NT sea el único sistema operativo en uso
- Cuando se utilicen servicios para Macintosh para compartición de archivos
- Se requiera seguridad de quinto nivel
- Los permisos tengan que ser preservados mientras se migran directorios y archivos de un servidor de NetWare
- Sea requerida compresión de archivos de Windows NT

El Setup formatea automáticamente con FAT, el cuál tiene un tamaño de partición máximo de 4GB. Si una partición tipo NTFS es creada desde el setup, la partición es formateada primero como FAT y es convertida a NTFS después de la instalación.

#### *Escogiendo el Rol del Servidor*

Los diferentes roles que puede jugar un sistema basado en Windows NT son:

- Controlador de Dominio Primario (PDC)
- Controlador de Dominio de Respaldo (BDC)
- Servidor Miembro o Stand-Alone

#### *Controlador de Dominio Primario (PDC)*

Una computadora en cada dominio, conocido como PDC, es responsable del mantenimiento de las cuentas de usuario. El PDC es definido durante la

instalación, y debe de estar en línea antes de que cualquier otra computadora pueda ser instalada en el dominio.

El PDC contiene la copia maestra del directorio de base de datos para el dominio. Cualquier cambio en la información de las cuentas es reflejado también en el PDC.

#### *Controlador de Dominio de Respaldo (BDC)*

El PDC copia periódicamente su directorio de base de datos a otras computadoras con Windows NT Server en el dominio. Estas computadoras son designadas durante la instalación como Controladores de Dominio de Respaldo. Un BDC puede autenticar y permitir acceso a usuarios del dominio. Es común tener más de un BDC.

Si el PDC falla, uno de estos BDC puede ser promovido como PDC. Los únicos datos de cuentas de usuarios que pueden ser perdidos, son aquellos cambios que aún no se han replicado en el BDC.

Cualquier BDC puede validar peticiones de acceso de los siguientes clientes:

- Windows NT
- Windows 95
- Windows para Trabajo en Grupos
- Microsoft LAN Manager
- Clientes de MS-DOS con características especiales

#### *Stand-Alone o Servidor Miembro*

Un servidor stand-alone tiene todas las características de Windows NT Server, incluyendo las siguientes:

- Soporte del servicio de RAS hasta para 256 estaciones simultáneas
- Tolerancia a fallas
- Servidor de Información de Internet
- Servicios de impresión y de archivos de Macintosh
- Servicio de Carga de Programa Remoto (RPL) que soporta clientes de Windows 3x, MS-DOS y Windows 95.

Contrario a los controladores de dominio, los servidores de stand-alone no participan en la replicación de cuentas o en la validación de acceso de los usuarios del dominio. Sin embargo, un servidor de dominio puede proveer servicios de archivo, impresión y de aplicación más eficientemente que un controlador de dominio.

Estos servidores pueden ser miembros de un grupo de trabajo o de un dominio. Es miembro de un dominio conocido como *servidor miembro*.

Similarmente a las computadoras corriendo Windows NT Workstation, los servidores stand-alone o miembros no pueden ser promovidos a BDC o a PDC sin reinstalar Windows NT Server.

## 2.2. COMUNICACIONES

### 2.2.1. Redes de Computadoras

#### Introducción

El siglo XVIII fue la etapa de los sistemas mecánicos que acompañaron a la Revolución Industrial; en el siglo XIX nos encontramos con la invención de la máquina de vapor; para este siglo, la tecnología clave ha sido la recolección, procesamiento y distribución de información. Entre otros muchos desarrollos, se ha observado la instalación de redes telefónicas en todo el mundo, la invención de la radio y la televisión, el nacimiento y crecimiento sin precedente de la industria de computadoras, así como la puesta en órbita de satélites de todo tipo y para muy variados fines. Conforme se acerca el final del siglo XX, se ha dado una rápida convergencia entre las áreas de computación y comunicaciones, logrando con esto, que las técnicas de captura, transporte, almacenamiento y procesamiento de información estén cambiando constantemente.

Organizaciones con centenares de oficinas dispersas en una amplia área geográfica esperan tener la posibilidad de examinar en forma habitual el estado actual de todas ellas, incluso las más alejadas, simplemente oprimiendo una tecla. A medida que crece la habilidad para recolectar, procesar y distribuir información, la demanda de más sistemas y procesamientos sofisticados de información crece aún con mayor rapidez.

Aunque la industria de computadoras es aún muy joven, comparada con otras industrias (por ejemplo la automotriz y la de transporte aéreo), las computadoras han mostrado un progreso espectacular en muy poco tiempo. Durante los primeros dos decenios de su existencia, los sistemas de cómputo estuvieron muy centralizados, usualmente en el interior de una habitación grande. Esta habitación, con frecuencia, tenía paredes de vidrio, a través de las cuales los visitantes se quedaban maravillados mirando el gran suceso electrónico del interior. Una compañía mediana o una universidad, podían contar con una o dos

computadoras, en tanto que las instituciones más grandes tenían a lo sumo una docena de ellas. La idea de que durante los siguientes 20 años sería posible producir, en forma masiva, computadoras más pequeñas que una tarjeta postal, pero igualmente poderosas, era como hablar de ciencia - ficción.

La función de las computadoras y las comunicaciones ha tenido una gran y marcada influencia en la forma en que estos sistemas están organizados. El concepto de "centro de cómputo" como una habitación con un servidor grande, al cual sus usuarios traían su trabajo para procesamiento, ha llegado a ser obsoleto. Este modelo no tiene uno, sino al menos dos aspectos deficientes: primero, el concepto de un solo servidor grande haciendo todo el trabajo y, segundo, la idea de que los usuarios traigan su trabajo al servidor en vez de llevar éste a donde se encuentren los usuarios.

El modelo de tener un solo servidor para satisfacer todas las necesidades de cómputo de una organización se está reemplazando con rapidez por otro que considera un número grande de servidores separados, pero interconectados, que efectúan el mismo trabajo. Estos sistemas, se conocen como **Redes**. A partir de este momento, se utilizará el concepto de "redes" para dar a entender una colección interconectada de servidores autónomos. Se dice que dos servidores están interconectados, si estos son capaces de intercambiar información. La conexión no necesita hacerse a través de un hilo de cobre; también puede hacerse mediante el uso de láser, microondas y satélites de comunicación. Al indicar que los servidores son autónomos, queremos excluir de nuestra definición a los sistemas en donde existe una clara relación maestro/esclavo. Si un servidor puede forzosamente arrancar, parar o controlar a otro, éstos no se consideran autónomos. Un sistema constituido por una unidad de control y muchos esclavos no es una red, ni tampoco lo es un servidor grande con lectoras de tarjetas de control remoto, impresoras y terminales.

### **2.2.1.1. Uso de Redes de Computadoras**

A medida que las computadoras personales (PC) se han venido convirtiendo en herramientas de trabajo en casi todas las actividades profesionales, las terminales independientes han dejado de ser la solución para muchos usuarios que necesitan información de diversas fuentes, en tal caso, se ha hecho indispensable utilizar una red de computadoras.

Las redes permiten aprovechar todas las capacidades de computación intercambiando y compartiendo recursos tanto físicos como lógicos.

El uso de una red de computadoras es tan variado debido a que se ajusta a las necesidades de cada aplicación, entre las más comunes se encuentran:

- Compartir información entre usuarios de una misma compañía.
- Correo electrónico.
- Acceso y comunicación remota a diferentes aplicaciones.
- Almacenamiento de grandes volúmenes de información.
- Intercambio de información entre usuarios ubicados en diferentes partes del mundo.

### **2.2.1.2. Objetivos**

Son muchas las organizaciones que ya cuentan con un número considerable de servidores en operación y, frecuentemente, alejados entre sí. Por ejemplo, una compañía con varias fábricas puede tener un servidor en cada una de ellas para mantener algún sistema, observar la productividad y llevar controles locales extra. Inicialmente, cada uno de los servidores puede haber estado trabajando en forma aislada de los demás, pero en algún momento, la administración puede decidir interconectarlos para tener así la capacidad de extraer y comparar información referente a toda la compañía.

Lo anterior va encaminado a compartir recursos, y el objetivo es hacer que todos los programas, datos y equipos estén disponibles para cualquiera de la red que así lo solicite, sin importar la localización física del recurso y del usuario mismo. En otras palabras, si el usuario se encuentra a 1000 Km. de distancia de la información, deberá acceder a la información necesaria como si se encontrara en el sitio mismo dónde reside la información. Otro aspecto relacionado con compartir recursos, es el de la compartición de la carga de trabajo.

Un segundo objetivo consiste en proporcionar una *alta fiabilidad* (datos veraces) al contar con fuentes alternativas de suministro y consulta. Por ejemplo, todos los archivos podrían duplicarse en dos o tres equipos, de tal manera que si una de ellas no se encuentra disponible (como consecuencia de un fallo de hardware) podría utilizarse alguna de las otras copias. Con ello, la presencia de múltiples CPU's involucra que si uno de ellos deja de funcionar, las otras pueden ser capaces de encargarse de su trabajo. Para aplicaciones militares, bancarias, control de tráfico aéreo, etc., es muy importante la capacidad de los sistemas para continuar funcionando a pesar de que se pudieran presentar algunos problemas.

Otro objetivo es el *ahorro*. Los equipos pequeños tienen una mejor relación costo/rendimiento, comparada con la ofrecida por los equipos grandes. Estas son a grandes riesgos, diez veces más rápidas que el más rápido de los microprocesadores, pero su costo es miles de veces mayor. Este desequilibrio ha ocasionado que muchos diseñadores de sistemas construyan sistemas constituidos por poderosos equipos personales, uno por usuario, con los datos guardados en una o más máquinas que funcionan como *servidor de archivo* compartido. Este objetivo conduce al concepto de redes con varios servidores localizados en el mismo edificio. A este tipo de red se le denomina LAN (Red de Área Local) en contraste con lo extenso de una WAN (Red de Área Extendida).

Un punto muy relacionado es la capacidad para aumentar el rendimiento del sistema en forma gradual a medida que aumenta la carga, simplemente añadiendo más procesadores. Con equipos grandes, cuando el sistema está lleno, deberá reemplazarse con uno más grande, operación que por lo general genera un gran gasto y una perturbación a los usuarios.

Otro objetivo del establecimiento de una red no tiene nada que ver con la tecnología. Una red puede proporcionar un poderoso medio de comunicación entre personas que se encuentren muy alejadas entre sí. Con el empleo de una red es relativamente sencillo para dos ó más personas, que viven en lugares separados, realizar alguna actividad juntos. Cuando un autor hace un cambio en un documento que se mantiene en línea, los otros pueden ver el cambio de inmediato, en lugar de esperar varios días para verlo reflejado.

### **2.2.1.3. Aplicaciones**

Con el avance de la tecnología vino una evolución de la computadora personal. A la PC se le añadieron capacidades de almacenamiento adicional y de procesamiento, con lo que cada vez se hizo más pequeña la diferencia entre la PC y las macro y microcomputadoras. De hecho, en algunos casos esto dio como resultado que la potencia de la PC superara a la de las mini o macrocomputadoras que entonces se utilizaban. Además éstas últimas tenían un alto costo de adquisición y mantenimiento, mientras que la PC podía comprarse por poco dinero y los costos de mantenimiento eran casi inexistentes.

Conectar las PC o las microcomputadoras para formar una red trae aparejados varios beneficios; por ejemplo, la posibilidad de obtener y utilizar información de una fuente común que, en algunos casos, es un requisito para operaciones como la contabilidad, en la cual diferentes personas necesitan aplicar los mismos datos al mismo tiempo. Otro beneficio inmediato de enlazar

en red a las PC es la posibilidad que se tiene de compartir impresoras y unidades de disco. De esta manera, todas las computadoras de una red suelen usar una sola impresora, una sola unidad grande de disco duro en donde se guarden datos que puedan ser utilizados por todos.

Existe un sin fin de aplicaciones para las redes de computadoras que van desde intercambiar correo entre usuarios hasta acceder una base de datos que se encuentra al otro lado del mundo.

#### 2.2.1.4. Estructura

Una LAN combina hardware y software especiales, lo que permite a las computadoras compartir programas, archivos, unidades de disco e impresoras. Los componentes de hardware para red se encuentran disponibles procedentes de diversos fabricantes. Por fortuna, los estándares bajo los que se fabrica el hardware de red están bien definidos, lo que da como resultado un alto nivel de compatibilidad con diferentes redes.

Los nodos de una red (computadoras) necesitan estar conectados para comunicarse. A la forma en que están conectados los nodos se le llama **Topología**. Una red tiene dos diferentes topologías: una física y una lógica. La topología física es la disposición física actual de la red, la manera en que los nodos están conectados unos con otros. La topología lógica es el método que se usa para comunicarse con los demás nodos, la ruta que toman los datos de la red entre los diferentes nodos. La topología física y lógica puede ser igual o diferente.

En una topología de BUS, cada computadora está conectada a un segmento común de cable de red (figura 2.2.1.1). El segmento de red se coloca como un bus lineal, es decir, un cable largo que va de un extremo a otro de la red, y al cual se conecta cada nodo.

El cable puede ir por el piso, por las paredes, por el techo, o puede ser una combinación de éstos, siempre y cuando el cable sea un segmento continuo.

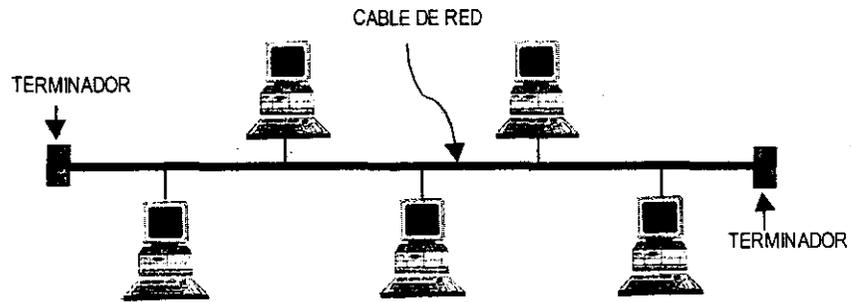


Figura 2.2.1.1 Topología BUS

En una topología de ESTRELLA, cada computadora está conectada a un concentrador (o HUB) ubicado centralmente (figura 2.2.1.2). El concentrador es un dispositivo de hardware con varios puertos, y se puede introducir un conector de cable de red en uno de ellos.

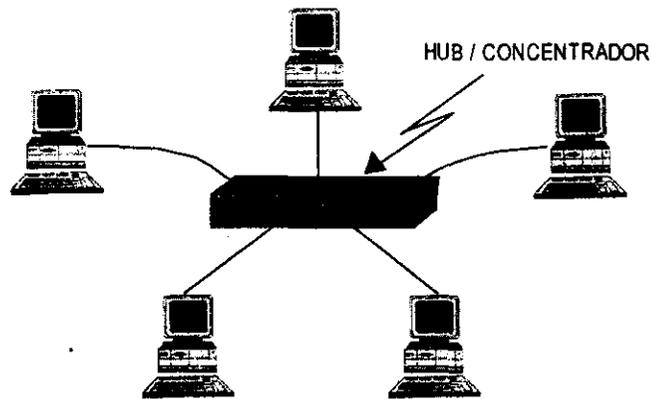


Figura 2.2.1.2. Topología de Estrella

En una topología de ANILLO, cada computadora se conecta en forma de anillo a la red. Las topologías de anillo casi siempre son lógicas con topología física de estrella. La figura 2.2.1.3 muestra la forma en la que fluyen los datos en una topología de anillo lógica conectada a una topología física de estrella. La topología física muestra que cada computadora se conecta a un dispositivo central y parece una estrella. La ruta seguida por los datos de una computadora a otra ilustra que la topología lógica es de anillo.

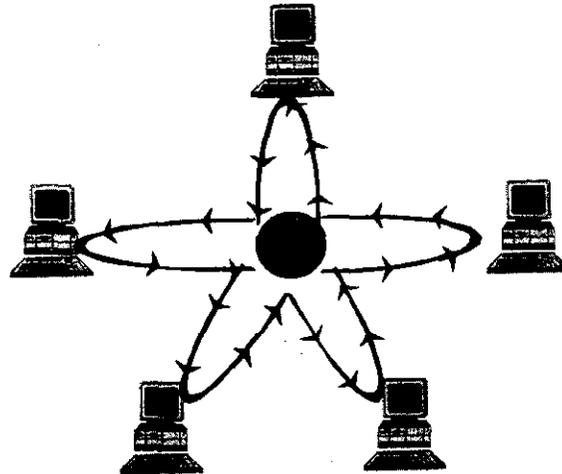


Figura 2.2.1.3. Topología física: Estrella  
Topología lógica: Anillo

Una topología de árbol es la combinación de las topologías de bus y de estrella. Muchos concentradores de las redes Ethernet con topología física de estrella tienen un conector en la parte trasera que enlaza al concentrador a una red de topología física de bus (figura 2.2.1.4).

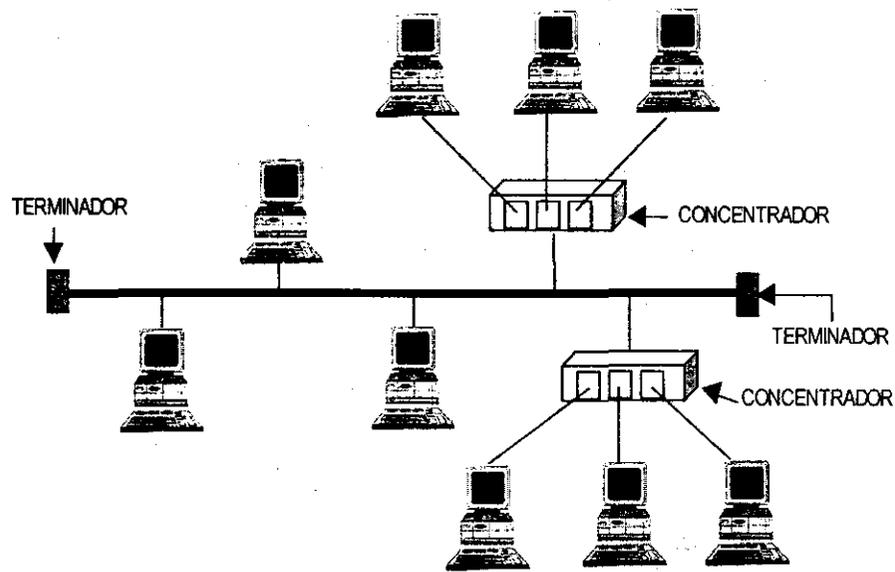


Figura 2.2.1.4. Topología de árbol

Las redes están compuestas por muchos componentes diferentes que deben trabajar juntos para crear una red funcional. Los componentes que comprenden las partes de hardware de la red incluyen tarjetas adaptadoras de red, los cables, conectores, concentradores y hasta la computadora misma.

Los componentes de red los fabrican, por lo general, varias compañías, por lo tanto, es necesario que haya entendimiento y comunicación entre los fabricantes en relación con la manera en que cada componente trabaja con los demás componentes de la red.

Para asegurar la compatibilidad entre los diferentes componentes se han creado estándares que definen la forma de conectar componentes de hardware en las redes y el protocolo (o reglas) de uso cuando se establecen

comunicaciones por red. Todos los datos que fluyen por el cable deben ir en secuencia y distinguirse, para que los diversos nodos puedan asegurarse de que los datos correctos lleguen al lugar pretendido.

Un **protocolo** es un juego de reglas que define la forma en que deben efectuarse las comunicaciones de las redes, incluyendo el formato, la temporización, la secuencia, la revisión y la corrección de errores.

Un **estándar** es la especificación de red (o serie de especificaciones) adoptada, e incluye guías y reglas que se refieren al tipo de componentes que deben usarse, a la manera de conectar los componentes, así como a los protocolos de comunicación que hay que emplear.

Los tres estándares más populares que se utilizan son: Arcnet, Ethernet y Token Ring. Los dos últimos son estándares respaldados por el Institute of Electrical and Electronic Engineers (IEEE); mientras que el primero es un estándar del American National Standards Institute (ANSI).

**Ethernet**, al que también se conoce como IEEE 802.3, es el estándar más popular para las LAN, este estándar emplea una topología lógica de bus y una topología física de estrella o de bus, también usa un método de transmisión de datos conocido como Acceso Múltiple con Detección de Portadora y Detección de Colisiones (CSMA/ CD). Antes de que un nodo envíe algún dato a través de una red Ethernet, primero escucha y se da cuenta si algún otro nodo está transfiriendo información. De no ser así, el nodo transferirá la información a través de la red. Todos los otros nodos escucharán y el nodo seleccionado recibirá la información. En caso de que dos nodos traten de enviar datos por la red al mismo tiempo, cada nodo se dará cuenta de la colisión y esperará una cantidad de tiempo aleatoria antes de volver a hacer el envío.

Existen cinco estándares de Ethernet 10BASE5, 10BASE2, 10BASE-T, 10BROAD36 y 100BASE-TX que definen el tipo de cable de red, las especificaciones de longitud y la topología física que debe utilizarse para conectar nodos en la red. El nombre está compuesto por tres partes (por ejemplo 10BASE5). El primer número indica la velocidad de datos, donde 10 significa 10 megabits por segundo. BASE especifica el funcionamiento en banda base y BROAD en banda ancha. La última parte indica el tipo de cable, 5 indica una configuración que admite cables de hasta 500 metros de longitud.

**10BASE5 (Thick Ethernet).** Es la configuración original de las redes Ethernet y utiliza un cable coaxial grueso de 50 ohmios. Los cables pueden alcanzar 500 metros sin necesidad de repetidores y cada sección de cable admite 100 estaciones conectadas. 10BASE5 tiene un estándar de topología física de bus que consiste en un segmento de cable de red con terminadores en los extremos (figura 2.2.1.5). Los terminadores incluyen una resistencia que disipa la señal de la red y no permite que se refleje de regreso. La tarjeta de interfaz de red en cada computadora es la interfaz de comunicaciones entre la computadora y el cable de red, y está conectada a un transmisor - receptor (transceiver) externo por medio de un cable de suspensión. El transceiver está conectado al segmento de cable de red y actúa para transmitir y recibir datos de la red entre la computadora y la red. El cable de la norma 10BASE5 es costoso y difícilmente manejable, por lo que se utiliza con menor frecuencia que otras opciones.

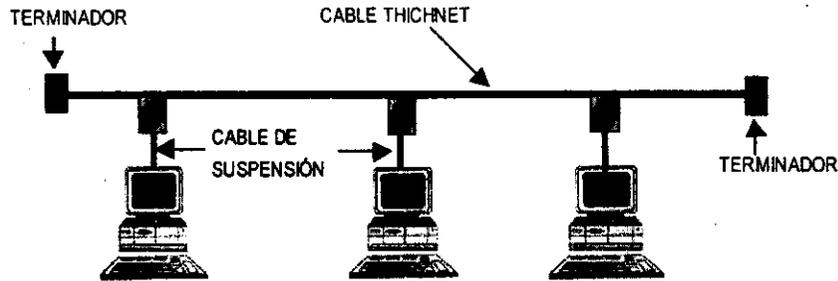


Figura 2.2.1.5 Topología física 10BASE5

**10BASE2 (Thin Ethernet).** El diseño de este sistema perseguía ofrecer una alternativa económica al sistema 10BASE5. Utiliza un cable coaxial más delgado que admite segmentos de hasta 185 mts. (el número 2 indica una longitud aproximada de 200 metros). 10BASE2 es económico y más fácil de instalar que 10BASE5, pero no se adapta bien a los sistemas de cableado estructurado que se configuran tirando un cable desde cada dispositivo hasta un concentrador (HUB) central. 10BASE2 se instala por medio de una topología física de bus, que consiste en segmentos de cable de red con terminaciones en cada extremo (figura 2.2.1.6). La tarjeta de interfaz de red de cada computadora está conectada directamente al segmento de cable de red, esto descarta la necesidad de un transceptor externo. Una desventaja de 10BASE2 es que, si llega a darse una ruptura en cualquier parte del cable, dejará de funcionar toda la red.

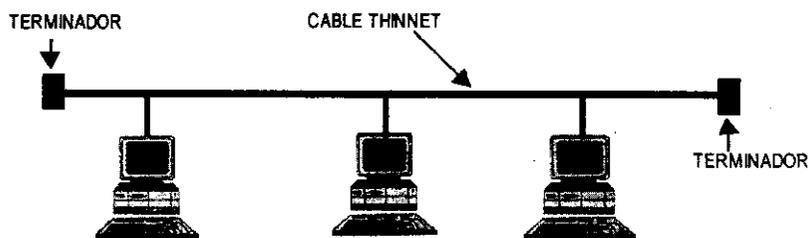


Figura 2.2.1.6. Topología Física 10BASE2

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

**10BASE-T(UTP).** La tendencia general consiste en reducir el uso de cables coaxiales y otros tipos de cable apantallado. Los diseñadores confían crecientemente en el cable de tipo par trenzado no apantallado (UTP Unshielded Twisted Pair) cuyo coste es ligeramente inferior al del cable coaxial. La letra T indica el uso de cable de tipo par trenzado. 10BASE-T utiliza un sistema de cableado basado en un concentrador y se adapta perfectamente al concepto de cableado estructurado. 10BASE-T se instala por medio de una topología física de estrella, cada nodo se conecta a un HUB (concentrador) (figura 2.2.1.7). La tarjeta de interfaz de red de cada computadora se conecta al concentrador por medio de un segmento de cable de red. Una ventaja de 10BASE-T es que la ruptura en el cable de red sólo desactivará a la computadora que esté al extremo de la línea rota, en vez de toda la red, como sucede con el 10BASE2.

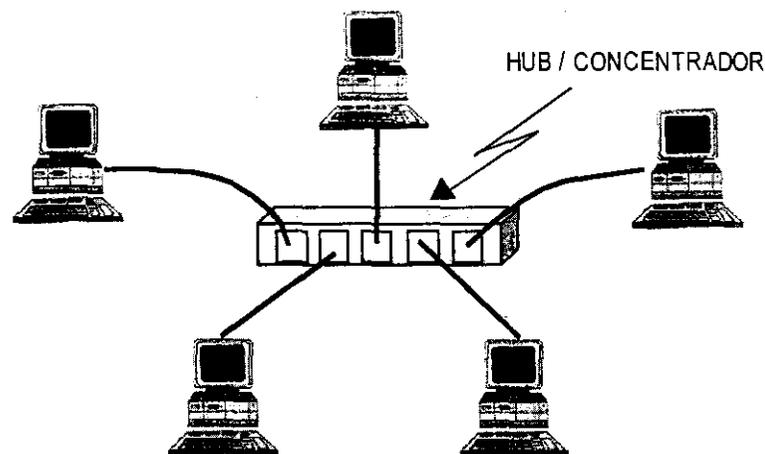


Figura 2.2.1.7. Topología física 10BASE-T (Par Trenzado)

**10BROAD36.** Es un sistema de cable de banda ancha que permite mantener varios canales de 10Mbps en un mismo cable coaxial.

**100BASE-TX.** Es una extensión del estándar Ethernet que opera a velocidades de 100 Mbp. Distintos comités IEEE 802 están evaluando varias normas de 100 Mbps. Todas utilizan cables UTP, pero difieren en la clase del cable y en el número de pares necesarios. 100BASE-TX utiliza un cable UTP de dos pares de clase alta. 100BASE-T4, una de las normas de 100 Mbps, utiliza un cable UTP de cuatro pares de datos estándar.

**Token Ring (IEEE 802.5).** Fue ideado por IBM y algunos otros fabricantes pero fue cedida a IEEE para su normalización. Con operación a una velocidad de 4 Mbps o 16 Mbps, Token Ring emplea una topología lógica de anillo y una topología física de estrella (figura 2.2.1.8). La tarjeta de interfaz de red de cada computadora se conecta a un cable que, a su vez, se enchufa a un HUB central llamado unidad de acceso a multiestaciones (MAU). Se pueden conectar las MAU de diferentes anillos en forma tal que los anillos que estaban en forma separada aparezcan como una sola red. IBM desarrolló la tecnología token ring (anillo de señales) para superar una deficiencia del método de acceso CSMA/CD. Existe la posibilidad de que la red esté ocupada cada vez que un dispositivo necesita transmitir. Incluso cuando el dispositivo inicia una transmisión, existe la probabilidad de que otro transmita y provoque una colisión obligando a ambos dispositivos a intentar una nueva transmisión. Dado que el acceso a una red CSMA/CD no está garantizado, se dice que el método esta basado en la probabilidad. El acceso por señales garantiza que cada dispositivo de la red reciba periódicamente la oportunidad de transmitir. La señal es un marco especial que circula de un dispositivo a otro a lo largo del anillo. El dispositivo propietario de la señal es el único que puede transmitir. Después de la transmisión, el dispositivo reinicia la señal permitiendo que otros puedan

transmitir. La implementación inicial token ring (4 Mbps) permitía la circulación de una sola señal por la red. Antes de liberar una señal para permitir que los dispositivos restantes transmitieran, el dispositivo emisor del marco debía esperar su retorno después de haber circulado por el anillo. Una nueva característica denominada liberación anticipada de señal, presentada con la tecnología token ring de 16 Mbps, permite que un dispositivo emisor libere una señal inmediatamente después de finalizar la transmisión de un marco. De este modo, una señal puede circular al mismo tiempo que un marco de datos.

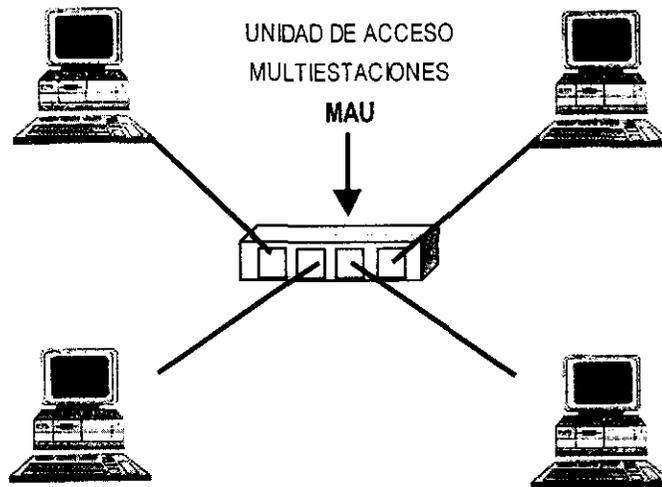


Figura 2.2.1.8. Topología Física Token Ring

**Arcnet**, Producida en los años setenta por Datapoint Corporation, la red de cómputo de recursos conectados (Arcnet) es un estándar aceptado por la industria, aunque no lleva número de estándar IEEE. En octubre de 1992, ANSI reconoció a Arcnet como estándar formal, lo que la hizo parte de su estándar de LAN ANSI 878.1. Como soporta una velocidad de transferencia de datos de 2.5 Mbps, Arcnet usa una topología lógica de bus y una ligera variación de la topología física de estrella (figura 2.2.1.9). Cada nodo de la red está conectado a un concentrador pasivo o a uno activo. Un concentrador pasivo no recibe potencia eléctrica y sirve para distribuir la señal de la red a cortas distancia. Un concentrador activo sí recibe potencia eléctrica y también amplifica la señal de la red para permitir que la red cubra distancias más largas. La tarjeta de interfaz de red de cada computadora está conectada a un cable que a su vez está conectado a un concentrador activo o pasivo.

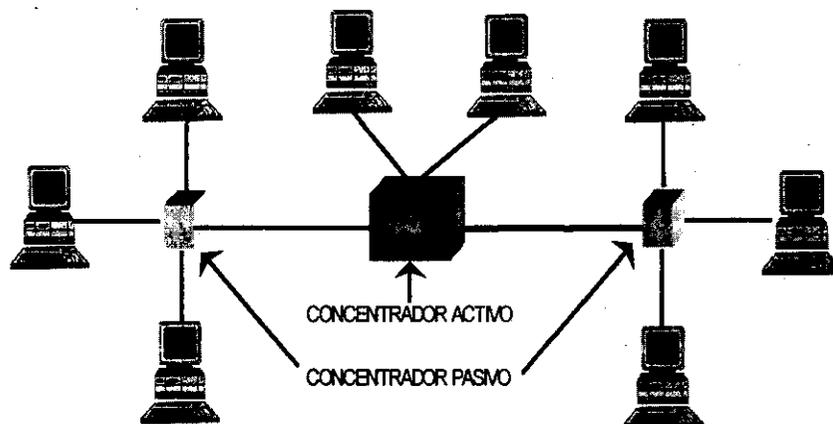


Figura 2.2.1.9 Topología física de ARCNET

A través de los años se han producido diversas variaciones del estándar Arcnet, incluyendo la Arcnet de bus lineal que usa una topología física de bus y la Arcnet de par trenzado que usa cableado UTP. Estas variaciones mejoran la flexible naturaleza del estándar Arcnet y permiten que los nodos de la red sean configurados en casi cualquier topología física.

Arcnet se base en un esquema de paso de señal (token passing) para administrar el flujo de datos entre los nodos de la red. Cuando un nodo está en posesión del token (señal) puede transmitir datos por la red. Todos los nodos, a excepción del receptor pretendido, pasan por alto los datos. Conforme se pasa el token a cada nodo, el nodo puede enviar datos. Debido a esto no suceden las colisiones que suelen darse en un esquema como el de CSMA/CD.

Conforme los usuarios transfieren más datos por la red y ésta a su vez debe alcanzar distancias más grandes, existen varias tecnologías nuevas que satisfacen las necesidades de las redes actuales, como las FDDI/CDDI y ATM.

**FDDI y CDDI.** La interfaz de distribución de datos por fibra óptica (FDDI) es un estándar para la transferencia de datos por cable de fibra óptica. El estándar ANSI X3T9.5 para FDDI especifica una velocidad de 100Mbps. Dado que el cable de fibra óptica no es susceptible a la interferencia eléctrica o tan susceptible a la degradación de la señal de red como sucede con los demás cables de red, FDDI permite el empleo de cables mucho más largos que otros estándares de red, además adopta una topología lógica de anillo con paso de token.

**ATM (Asynchronous Transfer Mode).** El modo de transferencia asíncrona (ATM), es un conjunto de estándares internacionales para la transferencia de datos, voz y video por medio de una red a muy altas velocidades. Puesto que opera a velocidades que van desde 1.5 Mbps hasta 1.5 Gbps, incorpora parte de los estándares Ethernet, Token Ring y FDDI para la transferencia de

datos. Tanto el vídeo como audio pueden representarse en forma digital, pero los requisitos de este tipo de datos suelen exceder la capacidad habitual de las computadoras. La digitalización de una señal de vídeo en movimiento puede requerir un ancho de banda mínimo de 6 Mbps aunque éste es tan solo uno de los problemas asociados con los datos de vídeo de una red. Las señales gráfica y de audio de una señal de vídeo constituyen dos corrientes de datos separadas que deben permanecer sincronizadas en tiempo real.

#### 2.2.1.5. Arquitectura

La comunicación de datos es sorprendentemente similar a la conversación humana. Tanto las personas como las computadoras utilizan una comunicación formal para realizar intercambios complejos de datos y procesos informales para objetivos específicos. Ambos respetan unos protocolos, reglas que logran que los sujetos intercambien información de manera ordenada y libre de errores. Los protocolos se siguen para establecer y finalizar la comunicación de modo que ninguna parte quede bloqueada en un estado no deseable.

El sistema de comunicación se desarrolla en capas. Cada una de ellas cuenta con su propia responsabilidad definida específicamente. Cuando se desea que alguien reciba una carta, sólo se tiene que colocar en un sobre con la dirección escrita, no es necesario preocuparse por las otras capas del proceso. Resulta indiferente si la carta se transporta en tren o en avión, esta responsabilidad es de otra capa. A este enfoque para diseñar un sistema de comunicaciones se *denomina Arquitectura de Capas*. Cada capa se responsabiliza de determinadas tareas y utiliza sus propias reglas para llevarlas a cabo sin conocer los procedimientos seguidos por otras capas (jerarquías de protocolos).

La mayoría de las redes se organizan en una serie de capas o niveles, con el objeto de reducir la complejidad de su diseño. Cada una de ellas se construye sobre su predecesora. El número de capas, el nombre, el contenido y función de cada una varían de una red a otra. Sin embargo, en cualquier red, el propósito de cada capa es ofrecer ciertos servicios a las capas superiores, liberándolas del conocimiento detallado sobre cómo se realizan dichos servicios.

La capa  $n$  en una máquina platica con la capa  $n$  de otra máquina. Las reglas y convenciones utilizadas en esa "charla" se conocen conjuntamente como **protocolo** de la capa  $n$ . A las entidades que forman las capas correspondientes en máquinas diferentes se les denomina *procesos pares*. En otras palabras, son estos procesos los que se comunican mediante el uso del protocolo.

Al conjunto de capas y protocolos se le denomina **Arquitectura de Red**. Las especificaciones de ésta deberán contener la información suficiente que le permita al diseñador escribir un programa o construir el hardware correspondiente a cada capa, y que siga en forma correcta el protocolo apropiado. Tanto los detalles de realización como las especificaciones de las interfaces no forman parte de la arquitectura, porque se encuentran escondidas en el interior de la máquina y no son visibles desde el exterior. Más aún, no es necesario que las interfaces de todas las máquinas de una red sean iguales, siempre y cuando cada una de las máquinas utilice correctamente todos los protocolos.

#### 2.2.1.6. Canales Punto a Punto

Una LAN punto a punto (también llamada "de igual a igual") permite que las computadoras de la red se configuren como servidores no dedicados de la red y de esta forma compartan los recursos de cada una de ellas (figura 2.2.1.10).

Una LAN punto a punto proporciona mucha más flexibilidad que una con servidor, puesto que aquella permite que cualquier computadora de la red comparta sus recursos con cualquier otra. La flexibilidad que proporciona una LAN punto a punto también puede hacer que la administración de ésta sea más confusa que la de una LAN basada en un servidor. Aunque las LAN punto a punto permiten poner a todos los nodos de la red como servidores no dedicados, casi todas ellas dan también la flexibilidad de colocar los nodos como estaciones de trabajo o como servidores dedicados.

Un aspecto importante de diseño, cuando se utiliza una red punto a punto, consiste en considerar cómo deberá ser la topología de interconexión. Las LAN que se diseñaron como tales, tiene por lo general una topología simétrica. A diferencia de éstas, las WAN tienen típicamente topologías irregulares.

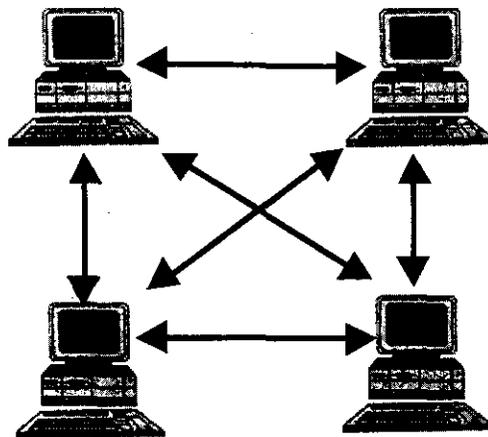


Figura 2.2.1.10 Red Punto a Punto

### 2.2.1.7. Canales Multipunto

Los canales multipunto son empleados como un segundo tipo de arquitectura de comunicación y la utilizan la mayoría de las LAN y un número reducido de WAN.

Este tipo de sistemas tiene un solo canal de comunicación que, a su vez, es compartido por todas las máquinas que constituyen la red. Los paquetes que una máquina envía son recibidos por todas las demás. El campo de dirección, localizado en el interior de un paquete, especifica a quién va dirigido. En el momento en que se recibe un paquete, se verifica el cambio de dirección y, si el paquete está destinado a otra máquina, simplemente es ignorado.

Por lo general, este tipo de sistemas también admite la posibilidad de dirigir un paquete a todos los destinos mediante el empleo de un código especial, incluido en el campo de la dirección. Cuando se transmite un paquete con dicho código, éste es recibido y procesado por todas las máquinas de la red.

Así mismo, las redes multipunto pueden dividirse en estáticas y dinámicas, dependiendo de cómo se haya asignado el canal. Una asignación estática típica sería dividir el tiempo en intervalos discretos y permitir así que cada máquina difunda su mensaje solo cuando le llegue su intervalo correspondiente. Las asignaciones estáticas desperdician la capacidad del canal cuando una de las máquinas no tiene nada que decir durante el tiempo que tiene asignado, así algunos sistemas asignan el canal en forma dinámica.

Los métodos de asignación dinámica para un canal común pueden ser centralizados o distribuidos. En el método de asignación de canal centralizado hay una entidad única, que determina quién es el siguiente. Esto podría hacerlo al aceptar solicitudes y tomar una decisión con base en algún algoritmo interno.

En el método de asignación de canal descentralizado no existe una entidad central; cada una de las máquinas deberá decidir por sí misma si transmite o no.

#### **2.2.1.8. Medios de Transmisión**

El propósito de un medio de transmisión consiste en transportar el flujo de datos de una máquina a otra. Normalmente se utilizan varios métodos físicos para realizar una transmisión. Cada estándar de red especifica el medio de transmisión (cableado) que se requiere y las especificaciones para la conexión de los nodos de la red. A continuación se presentan algunas características de algunos medios de transmisión.

##### **Medio Magnético**

Una de las formas más comunes para el transporte de datos, de una máquina a otra, consiste en escribir dicha información sobre una cinta magnética o en discos flexibles, y transportarlos físicamente hasta la máquina destino. Este método, aunque no tan sofisticado como aquellos en los que se utilizan satélites de comunicación, es bastante efectivo en costo, en especial en los casos en los que se necesitan anchos de banda grandes, o en donde el costo por bit transportado sea significativo.

Por ejemplo, una cinta magnética de 6250 bpi (bits por pulgada), que es un patrón industrial, puede almacenar 180 megabytes. Una camioneta o camión ligero llega a transportar 200 cintas a la vez. Supongamos que las máquinas fuente y destino se encuentran separadas por una distancia que se recorre en una hora de conducción aproximadamente. La velocidad efectiva de transmisión entre estas dos máquinas es por consiguiente de 288 000 megabits en 3600 segundos, o sea 80 Mbps. Ninguna WAN cae dentro del orden de magnitud de este ancho de banda, y muy pocas LAN pueden llegar a igualarlo.

Para un banco, por ejemplo, es probable que no exista ninguna otra tecnología de transmisión que pueda acercarse a la de cinta magnética en efectividad de funcionamiento - costo.

### **Par Trenzado**

Aunque las características del ancho de banda de una cinta magnética sean excelentes, las características de retardo son muy malas; su tiempo de transmisión se mide en minutos u horas, y no en milisegundos. En muchas aplicaciones resulta necesario tener conexiones en línea. El medio de transmisión más antiguo, y todavía el más ampliamente utilizado, es el par trenzado. Consiste en dos alambres de cobre aislados, en general de 1 mm de espesor. Los alambres se entrelazan en forma helicoidal, como en una molécula de DNA. La forma trenzada del cable se utiliza para reducir la interferencia eléctrica con respecto a los pares cercanos que se encuentran a su alrededor. Recordar que dos cables paralelos constituyen una antena simple, en tanto que un par trenzado no.

La distancia que se puede recorrer con este tipo de cables es de varios kilómetros, sin necesidad de amplificación, pero con la necesidad de repetidores para distancias más largas. Cuando hay muchos pares trenzados colocados paralelamente que recorren grandes distancias son agrupados y cubiertos con una malla protectora. Los pares dentro de estos grupos podrían sufrir interferencias mutuas si no estuvieran trenzados.

Los pares trenzados se pueden utilizar tanto como para transmisión analógica como digital y su ancho de banda depende del calibre del alambre y de la distancia que recorre; en algunos casos pueden obtenerse transmisiones de varios megabits/s, en distancias de pocos kilómetros. Debido a su buen comportamiento y su bajo costo los pares trenzados son ampliamente

utilizados y es muy probable que su presencia permanezca por muchos años más.

El UTP Ethernet es un cable que consiste en pares trenzados entre ellos (figura 2.2.1.11). Se emplea un total de cuatro conductores (o dos pares), para transmitir y recibir la señal de red. Puesto que los conectores estándar RJ-45 tienen ocho terminales de conexión, el cable que se instala tiene generalmente ocho conductores, aunque la red sólo use cuatro de ellos. Para la instalación y configuración del cable UTP Ethernet, se aplican las siguientes reglas:

- La longitud máxima de cable entre un nodo y un concentrador es de 100 metros.
- Las patas 1,2,3,y 6 del conector RJ-45 son conectadas de manera directa. Las patas 1,2 son transmisoras, y las 3 y 6 receptoras.
- Se pueden conectar hasta 12 concentradores a un concentrador central.
- Sin el uso de puentes, el cable Ethernet UTP puede acomodar un máximo de 1,024 estaciones de trabajo.

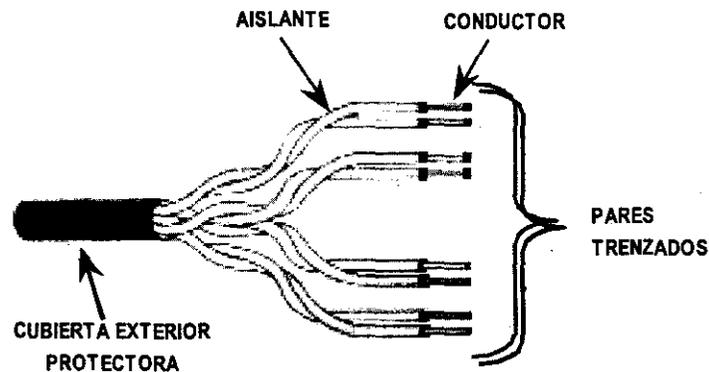


Figura 2.2.1.11 Cable UTP

### Cable Coaxial

El cable coaxial es otro medio típico de transmisión. Hay dos tipos de ellos que son utilizados con bastante frecuencia, uno de ellos es el de 50 ohms, que es utilizado en transmisión digital, en tanto que el otro tipo, el de 75 ohms es utilizado en transmisión analógica.

La construcción del cable coaxial produce una buena combinación de un gran ancho de banda y una excelente inmunidad al ruido. El ancho de banda que se puede obtener depende de la longitud del cable; para cables de 1 km., por ejemplo, es factible obtener velocidades de datos de hasta 10 Mbps. y, en cables de longitudes menores, es posible obtener velocidades superiores. Se pueden utilizar cables de mayor longitud, pero se obtienen velocidades más bajas. Los cables coaxiales son empleados ampliamente en redes LAN.

El cable empleado por el estándar 10BASE5 es un tipo especial de cable coaxial (figura 2.2.1.12). El conductor central está rodeado por un aislante dieléctrico al que, a su vez, lo rodea un blindaje de hoja de metal. Alrededor del blindaje de hoja de metal, hay un conductor tejido rodeado por otro blindaje de hoja de metal que, también, está cubierto por un conductor tejido. La parte externa del cable tiene una cubierta protectora.

Las reglas para la instalación y configuración de segmentos de este tipo de cable son:

- La longitud máxima de segmento de red es de 500 metros.
- Cada segmento de red debe tener una terminación de 50 Ohmios en cada extremo.
- No pueden conectarse en serie más de cinco segmentos de red y sólo tres de éstos pueden estar ocupados.
- La cantidad máxima de transmisores-receptores por segmento es de 100.

- La cantidad máxima de nodos en una red es de 1,024.
- Los transceptores no pueden instalarse a menos de 2.5 metros.
- Los cables de bajada no pueden ser más largos de 50 metros.
- La distancia máxima entre dos estaciones cualquiera es de 3,000 metros.

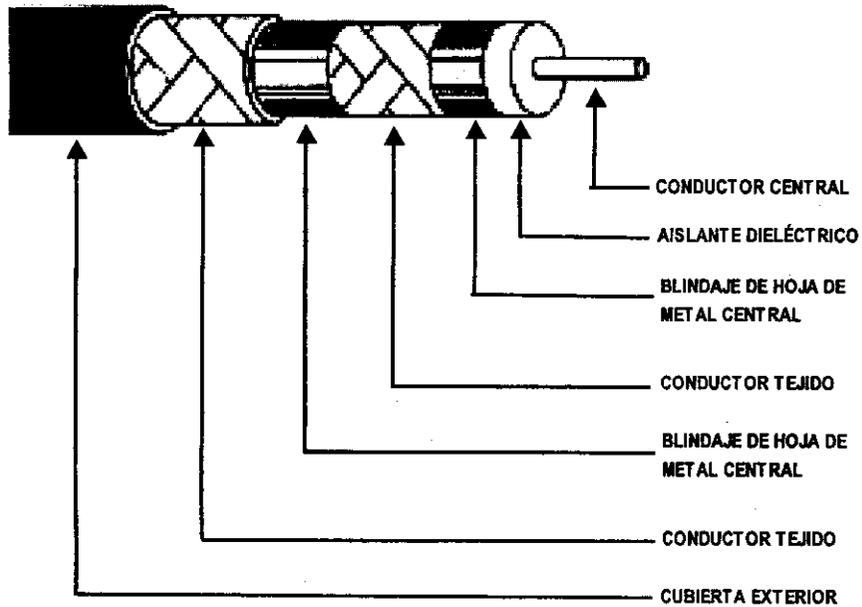


Figura 2.2.1.12 Cable empleado para el estándar 10BASE5

El cable empleado para el estándar 10BASE2, consiste en un conductor interno rodeado por un aislante dieléctrico, un blindaje de hoja de metal, un conductor tejido y una cubierta exterior protectora (figura 2.2.1.13). Para la instalación y configuración de este cable, se aplican las siguientes reglas:

- La longitud máxima del segmento debe ser de 185 metros.
- Cada segmento de red debe tener una terminación de 50 Ohms en cada extremo.
- No puede conectarse en serie más de cinco segmentos de red y sólo tres de éstos pueden estar ocupados.
- La cantidad máxima de nodos por segmento es 30.
- La distancia mínima de cable entre adaptadores de red es 0.5 metros.
- La cantidad máxima de nodos en una red es de 1,024.
- La distancia máxima entre nodos cualquiera es de 1,425 metros.

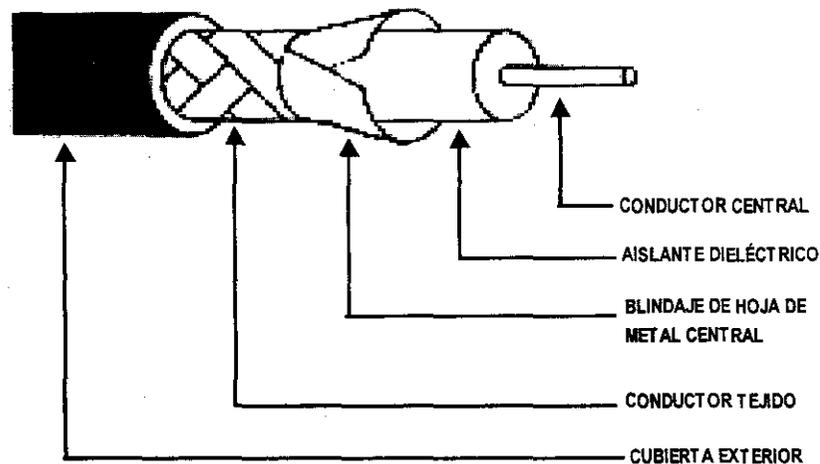


Figura 2.2.1.13 Cable empleado para el estándar 10BASE2

## Fibras Ópticas

Los desarrollos recientes en el campo de la tecnología óptica han hecho posible la transmisión de información mediante pulsos de luz. Un pulso de luz puede utilizarse para indicar un bit de valor 1; la ausencia de un pulso indicará la existencia de un bit de valor 0. La luz visible tiene una frecuencia de alrededor de 108 MHz., por lo que el ancho de banda de un sistema de transmisión óptica presenta un potencial enorme.

Un sistema de transmisión óptica tiene tres componentes: el medio de transmisión, la fuente de luz y el detector. El medio de transmisión es una fibra ultradelgada de vidrio o silicio fundido. La fuente de luz puede ser un LED, o un diodo láser; cualquiera de los dos emite pulsos de luz cuando se aplica una corriente eléctrica. El detector es un fotodiodo que genera un pulso eléctrico en el momento en el que recibe un rayo de luz. Al colocar un LED o un diodo láser en el extremo de una fibra óptica, y un fotodiodo en el otro se tiene una transmisión de datos unidireccional que acepta una señal eléctrica, la convierte y la transmite por medio de pulsos de luz y después reconvierte la salida en una señal eléctrica, en el extremo receptor.

En la actualidad los sistemas de fibras ópticas son capaces de hacer transmisiones de datos de 1000 Mbps en 1 kilómetro. Experimentalmente se ha demostrado que los láseres potentes pueden llegar a excitar fibras de 100 km de longitud sin necesidad de utilizar repetidores, aunque la velocidad es más lenta.

Un solo cable de fibra consiste normalmente en una fibra rodeada por un recubrimiento amortiguador (figura 2.2.1.14). El recubrimiento amortiguador está rodeado de Kevlar para una protección. La cubierta protectora exterior está compuesta de PVC o poliuretano negro. Con frecuencia, el cable de fibra contiene más de una de ellas.

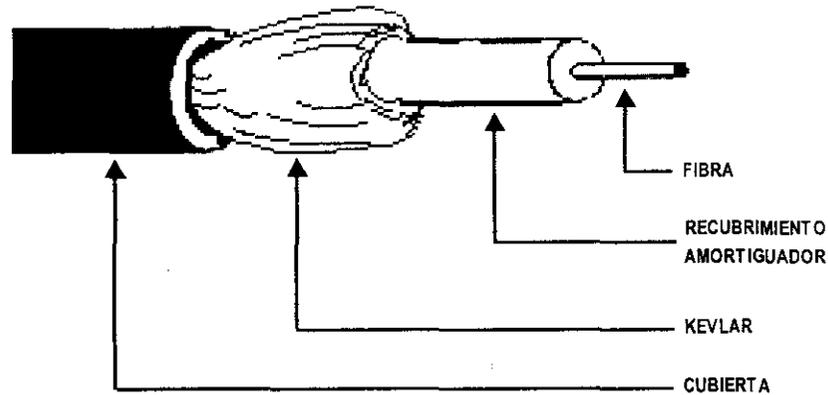


Figura 2.2.1.14 Fibra Optica

### Transmisión Inalámbrica

La transmisión de datos por rayos infrarrojos, láser, microondas o radio, no necesita de ningún medio físico, cada una de estas técnicas se adapta a la perfección a ciertas aplicaciones.

Una aplicación común en donde el recorrido de un cable o fibra resulta en general indeseable, es el caso del tendido de una LAN por varios edificios localizados en una escuela y oficinas de un centro empresarial, o bien en un complejo industrial. En el interior de cada edificio, la LAN puede utilizar cobre o fibra, pero para las conexiones que se hagan entre los edificios necesitarían hacerse excavaciones en las calles para construir una zanja adecuada en la que se pueda depositar el cable, esto genera un costo bastante considerable y si el trazado de dicha zanja debe cruzar una calle pública, dicho trabajo puede llegar a ser incluso ilegal. Por otra parte, el hecho de poner un transmisor y/o receptor láser o infrarrojo en el techo de un edificio o en una ventanilla resulta

muy económico, fácil de llevar a cabo y casi siempre estará permitida su realización.

Este tipo de diseño nos conduce a una jerarquía de redes, en donde la red dorsal, que vendría a ser la red de láser o infrarrojo, estaría localizada entre los edificios. La desventaja de este tipo de comunicación es que la lluvia y neblina pueden ocasionar interferencia dependiendo de la longitud de onda elegida.

Como una alternativa del cable coaxial, en aplicaciones para comunicaciones de larga distancia, se ha utilizado muy ampliamente la transmisión por radio de microondas. Las antenas parabólicas se pueden montar sobre torres para enviar un haz de señales a otra antena que se encuentre a decenas de kilómetros de distancia. Este sistema está siendo utilizado en transmisiones telefónicas y de video, cuanto mayor altura tenga la torre, más amplia será la cobertura.

La ventaja de las microondas es que la construcción de dos torres resulta, por lo general, más económica que abrir una zanja de 100 km de longitud sobre la cual se pueda depositar el cable o la fibra, y posteriormente volver a cubrirla. Por otra parte, las señales de una antena pueden dividirse y propagarse, siguiendo trayectorias ligeramente diferentes, hacia la antena receptora. Cuando estas señales, que se encuentran desfasadas, se recombinan, puede haber interferencia entre ellas, de tal manera que se reduce la intensidad de la señal. La propagación de las microondas también se ve afectada por las tormentas y otros fenómenos atmosféricos.

#### **Comunicación por satélites.**

La comunicación mediante satélite tiene algunas propiedades que la hacen atractiva en algunas aplicaciones. Este tipo de comunicación puede imaginarse como si un enorme repetidor de microondas estuviese localizado en el cielo; el

cual está construido por uno o más dispositivos receptor - transmisor, que a su vez cada uno de ellos escucha una parte del espectro, amplificando la señal de entrada y después, la retransmite a otra frecuencia, para evitar los efectos de interferencia con las señales de entrada. El flujo dirigido hacia abajo puede ser muy amplio y cubrir una parte significativa de la superficie de la tierra, o bien, puede ser estrecho y cubrir un área de cientos de kilómetros de diámetro.

La comparación entre la comunicación por medio de satélites y la realización por medio de fibras ópticas es interesante. Mientras que una simple fibra tiene, en principio, un ancho de banda potencial mayor que el de todos los satélites puestos en órbita, este ancho de banda no es accesible para todos los usuarios. Las fibras que actualmente se instalan, se emplean en el sistema telefónico con objeto de manejar muchas llamadas de larga distancia a la vez, y no proporcionar un ancho de banda grande para un usuario individual. Más aún, muy pocos usuarios son los que tienen realmente acceso a un canal de fibra óptica. Una llamada a la oficina de la compañía de local de teléfonos, a una velocidad de 9600 bps, nunca dará un ancho de banda mayor a 9600 bps, sin importar cual es el ancho de banda del enlace intermedio. Con los satélites, resulta muy práctico para los usuarios montar una antena en el techo de su edificio y con esto saltarse totalmente al sistema telefónico. Para los países del tercer mundo, con territorios totalmente hostiles y con muy poca infraestructura existente, los satélites son una idea atractiva. Indonesia, por ejemplo, ya cuenta con satélite propio para el tráfico telefónico doméstico.

### **2.2.2. Modelos de Interconexión.**

Cuando una red crece, se necesita una extensión de longitud de cable o exceder la cantidad de nodos especificado en el estándar empleado. Afortunadamente existen varios dispositivos diseñados para este fin.

**HUBS y CONCENTRADORES.** Son un punto central de conexión para nodos de red que están conectados a una topología física de estrella. Los concentradores son dispositivos que se encuentran físicamente separados de cualquier nodo de red, aunque algunos concentradores de hecho se conectan a un puerto de expansión en un nodo de red. El concentrador tiene varios puertos en la parte trasera de la tarjeta, a los que se conecta el cable de otros nodos de red. Pueden conectarse varios concentradores para permitir la conexión de nodos adicionales. En la figura 2.2.2.1 aparecen conectados dos concentradores de cuatro puertos. Ahí, ambos concentradores usan cable UTP y clavijas RJ-45 para la conexión. Se utiliza un puerto para cada concentrador para conectarse con el otro concentrador. El cable empleado para conectar a los concentradores es el mismo que se usa entre el concentrador y los nodos de red, a excepción de que los alambres están traslapados entre los dos conectores a cada extremo.

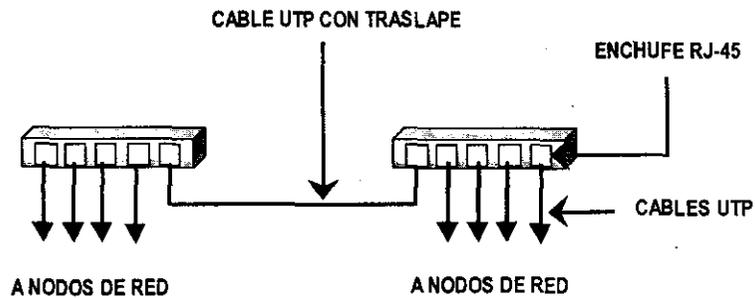


Figura 2.2.2.1 Conexión de dos HUB's

**REPETIDORES.** Un repetidor es un dispositivo que permite extender la longitud de la red, amplifica y retransmite la señal de red. Por ejemplo, la longitud máxima de segmento de cable para Ethernet 10BASE2 es de 607 pies. Si se coloca un repetidor al extremo del cable, se puede conectar otro segmento de cable de hasta 607 pies para dar un total de 1,214 pies (figura 2.2.2.2a). Los repetidores multipuertos permiten conectar más de dos segmentos de cable de red. Si se tiene un repetidor multipuerto, se pueden conectar varios segmentos de cable para formar una combinación de topologías física de bus y estrella (figura 2.2.2.2b).

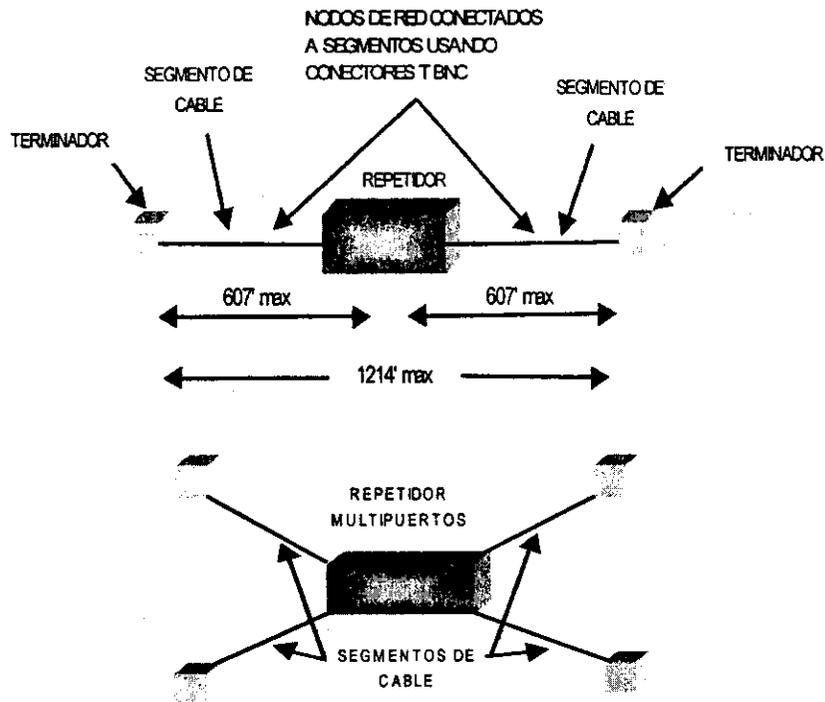


Figura 2.2.2.2 a) Longitud máxima permitida de un cable de red con un repetidor  
b) Uso de un repetidor multipuertos

**PUNTES.** Un puente es un dispositivo que conecta dos LAN separadas para crear lo que aparenta ser una sola LAN. Los puentes revisan la dirección asociada con cada paquete de información. Luego, si la dirección es la correspondiente al otro segmento de red, el puente pasará el paquete al segmento. Si el puente reconoce que la dirección es la correspondiente a un nodo del segmento de la red actual, no pasará el paquete al otro lado. Se puede considerar el caso de dos redes separadas, una que opera en 10BASE2 y la otra basada en un esquema de cableado propio con adaptadores de red propios, la función del puente es transmitir la información enviada por un nodo de una red al destino pretendido en otra red.

Los puentes también suelen emplearse para reducir la cantidad de tráfico en un segmento de red. Mediante la división de un solo segmento de red en dos segmentos y conectándolos por medio de un puente, se reduce el tráfico general en la red.

La unión de dos segmentos de red Ethernet 10BASE2 por medio de comunicaciones inalámbricas es otra aplicación que se le da a los puentes (figura 2.2.2.3). Los puentes vienen en todas formas y tamaños. En muchos casos, un puente es un dispositivo similar a una computadora con conectores a los que se conectan redes separadas. En otros casos, un puente es, de hecho, una computadora con un adaptador para cada red que va a conectarse. Un software especial permite el paso de la información adecuadamente a través de los adaptadores de un segmento de red a otro.

En los últimos años se han desplegado estratégicamente dispositivos de LAN denominados **CONMUTADORES** para mejorar el ancho de banda de las redes. Básicamente, un conmutador de LAN es un puente dotado de una gran cantidad de puertos. Además, utiliza técnicas que permiten aumentar la velocidad de envío de los marcos. Por ejemplo, no suelen perder tiempo guardando la totalidad de un paquete en la memoria intermedia antes de

enviarlo; una vez identificada la dirección física de destino, el conmutador puede iniciar el envío del paquete sin esperar a recibir el final del marco.

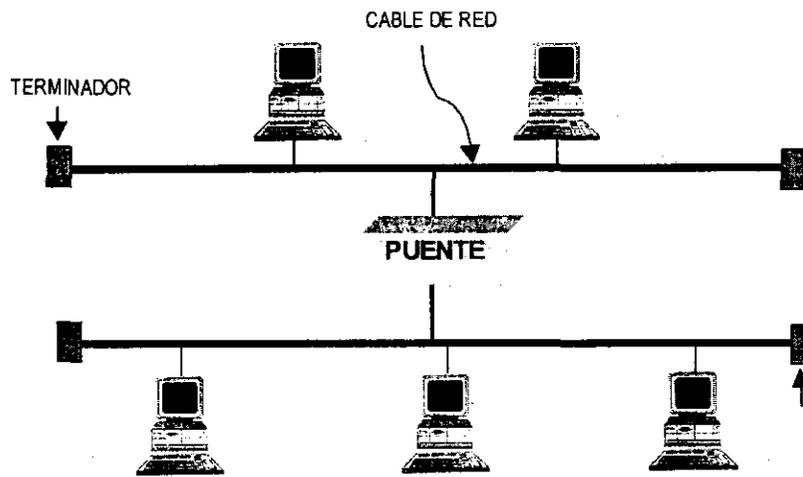


Figura 2.2.2.3 Uso de un puente para conectar redes similares

**RUTEADORES.** Los ruteadores son similares a los puentes, sólo que operan a un nivel diferente. Los ruteadores requieren por lo general que cada red tenga el mismo sistema operativo de red (NOS). Con un NOS común, el ruteador puede ejecutar funciones más avanzadas de las que podría permitir un puente, como conectar redes basadas en topologías lógicas completamente diferentes como Ethernet y Token Ring. Los ruteadores también suelen ser lo suficientemente inteligentes para determinar la ruta más eficiente para el envío de datos, en caso de haber más de una ruta. Sin embargo, junto con la complejidad y la capacidad adicional proporcionada por los ruteadores, el aumento de su costo y un rendimiento disminuido los hace menos utilizados.

**COMPUERTAS (GATEWAYS).** Un gateway permite que los nodos de una red se comuniquen con tipos diferentes de red o con otros dispositivos. Podrían tenerse por ejemplo, una LAN que consista en computadoras compatibles con IBM y otra que consista en computadoras Macintosh. En este caso, una compuerta permitiría que las computadoras IBM compartieran archivos con las Macintosh. Este tipo de compuertas también permite que se compartan impresoras entre las dos redes. Algunas otras permiten conectarse a otros tipos de dispositivos, como las máquinas de fax, por ejemplo.

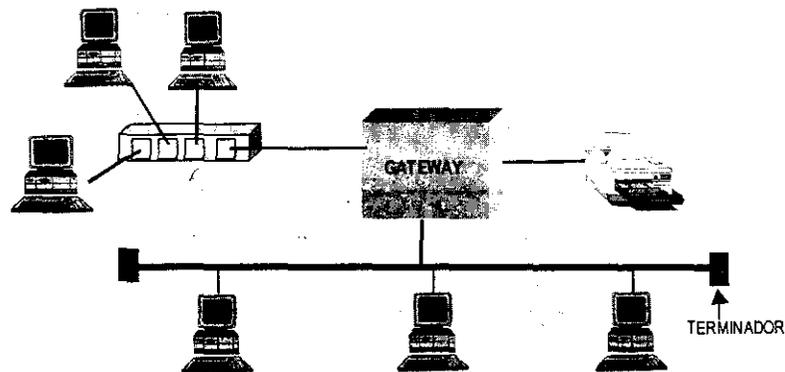


Figura 2.2.2.4 Uso de un GATEWAY para conectar redes no similares

### 2.2.3. Modelo OSI.

El paso más divulgado hacia la estandarización de las redes de computadoras fue la definición del modelo de referencia para la interconexión de sistemas abiertos OSI (Open Systems Interconnection) propuesto por la organización internacional de estándares ISO (International Standards Organization).

Es importante señalar que el modelo OSI, por sí mismo, no es una arquitectura de red, dado que no especifica, en forma exacta, los servicios y protocolos que se utilizarán en cada una de las capas. Sólo indica lo que cada capa deberá hacer.

Sin embargo, la ISO también ha generado normas para todas las capas, aunque éstas, estrictamente hablando, no forman parte del modelo. Cada una de ellas se han publicado como normas internacionales independientes. El modelo de referencia OSI esta compuesto por 7 capas o estratos (figura 2.2.3.1). Las primeras dos capas (física y enlace de datos) son el hardware que la LAN comprende, como los cables Ethernet y los adaptadores de red. Las capas 3,4,5 (de red, transporte y sesión) son protocolos de comunicación, como el sistema básico de entrada/salida de red (NETBIOS), TCP/IP, y el protocolo modular Netware (NCP) de Novell. Las capas 6 y 7 (de presentación y aplicación) son el Sistema operativo de Red (NOS) que proporciona servicios y funciones de red al software de aplicación.



Figura 2.2.3.1 Modelo OSI

### 1. Capa Física.

La capa física define la interfaz con el medio físico, incluyendo el cable de red, maneja temas como la intensidad de la señal de red, los voltajes indicados para cada señal y la distancia de los cables.

Esta capa se ocupa de la transmisión de bits a lo largo de un canal de comunicación. Su diseño debe asegurar que cuando un extremo envía un bit con valor 1, éste se reciba como un bit con ese valor en el otro extremo, y no como un bit con valor 0. Aquí se especifica cuantos voltios deberán utilizarse para representar un bit de valor 0 o 1; cuantos microsegundos deberá durar un bit; la posibilidad de realizar transmisiones bidireccionales en forma simultánea; la manera de establecer la conexión inicial y como interrumpirla cuando ambos extremos terminan su comunicación; o bien, cuantas puntas terminales tiene el conector de la red y cual es el uso de cada una de ellas (figura 2.2.3.2).

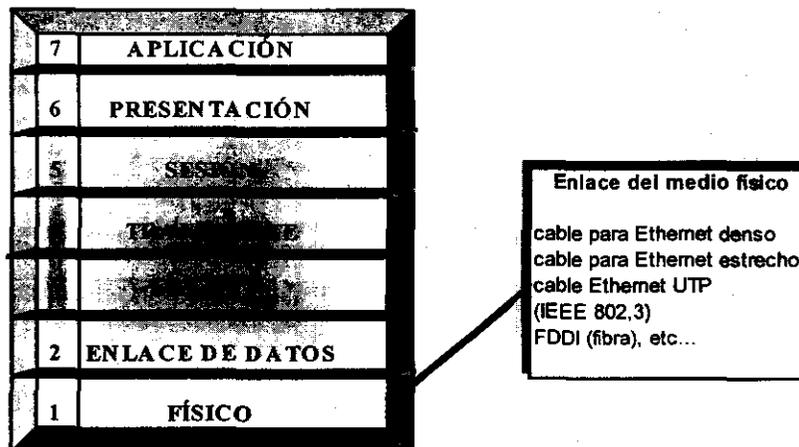


Figura 2.2.3.3 Capa Física

## 2. Capa de Enlace.

La capa de enlace de datos define el protocolo que detecta y corrige errores cometidos al transmitir datos por el cable de la red (figura 2.2.1.21). Esta capa es la causante del flujo de datos de la red, los cuales se dividen en paquetes o cuadros de información. Cuando un paquete de información es recibido incorrectamente, la capa de enlace de datos hace que se reenvíe. La capa de enlace está dividida en dos subcapas: la del control de acceso al medio (MAC) y el control de enlace lógico (LLC). Los puentes, ruteadores y gateways operan en la capa MAC. Cuando la capa de enlace de datos recibe un mensaje, le da formato para transformarlo en un marco de datos o paquete (figura 2.2.3.4).

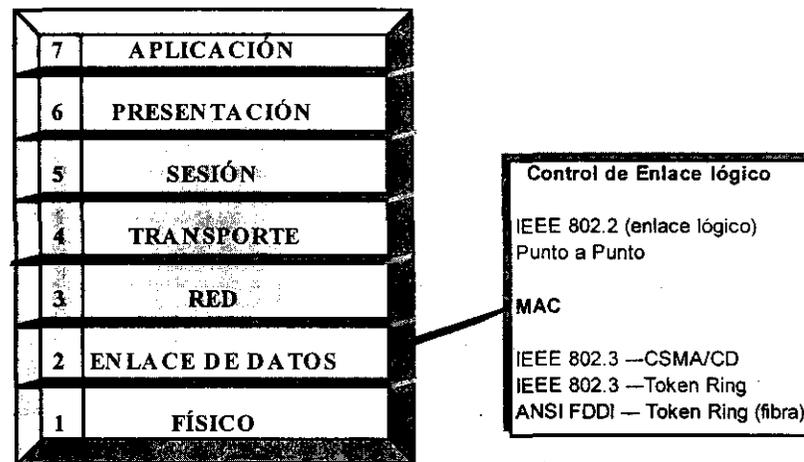


Figura 2.2.3.4 Capa de Enlace de Datos

**Indicador de inicio.** Un patrón de bits que indica el inicio de un marco de datos.

**Dirección de origen.** La dirección del nodo que realiza el envío se incluye para poder dirigir las respuestas al mensaje.

**Dirección de destino.** Cada nodo queda identificado por una dirección. La capa de enlace de datos del remitente añade la dirección de destino al marco. La capa de enlace del destinatario examina la dirección de destino para identificar los mensajes que debe recibir.

**Control.** En muchos casos es necesario incluir información adicional de control. Cada protocolo determina la información específica.

**Datos.** Este campo contiene todos los datos enviados a la capa de enlace de datos por las capas superiores del protocolo.

**Control de errores.** Este campo contiene información que permite que el nodo destinatario determine si se ha producido algún error durante la transmisión. El sistema habitual es la verificación de redundancia cíclica (CRC), que consiste en un valor calculado que resume todos los datos del marco, entiende que el marco se ha transmitido sin errores.

### **3. Capa de Red.**

Las redes más pequeñas consisten en una sola LAN, pero la mayoría de las redes deben subdividirse. Una red que consta de varios segmentos de red suele denominarse subred o interred.

La capa de red se ocupa del control de la operación de la subred. Un punto de suma importancia en su diseño, es la determinación sobre cómo encaminar los paquetes del origen al destino. Para entregar mensajes en una subred, cada una de ellas debe estar identificada de manera única por una dirección de red. Al recibir un mensaje de las capas superiores, la capa de red añade una cabecera al mensaje que incluye las direcciones de red de origen y destino.

La capa de red opera con independencia del medio físico, y dado que los encaminadores son dispositivos de la capa de red, pueden utilizarse para intercambiar paquetes entre distintas redes físicas. Por ejemplo, un encaminador puede enlazar una red Ethernet a una red token ring.

#### **4. Capa de Transporte.**

La función principal de esta capa, consiste en aceptar los datos de la capa de sesión, dividirlos, siempre que sea necesario, en unidades más pequeñas, pasarlos a la capa de red y asegurar que todos ellos lleguen correctamente al otro extremo. Bajo condiciones normales, la capa de transporte crea una conexión de red distinta para cada conexión de transporte solicitada por la capa de sesión. Si la conexión de transporte necesita un gran caudal, ésta podría crear múltiples conexiones de red, dividiendo los datos entre las conexiones de la red con objeto de mejorar dicho caudal. Según el modelo OSI, la capa de transporte asigna una identificación de punto de acceso a servicio (SAP) a cada paquete. La dirección de un SAP es una dirección que identifica el proceso que ha originado el mensaje, dicha dirección permite que la capa de transporte del nodo receptor encamine el mensaje al proceso adecuado. A la identificación de mensajes de distintos procesos para posibilitar su transmisión a través de un mismo medio de red se le denomina *multiplexión*. El procedimiento de recuperación de mensajes y de su encaminamiento a los procesos adecuados se le llama *demultiplexión* (figura 2.2.3.5).

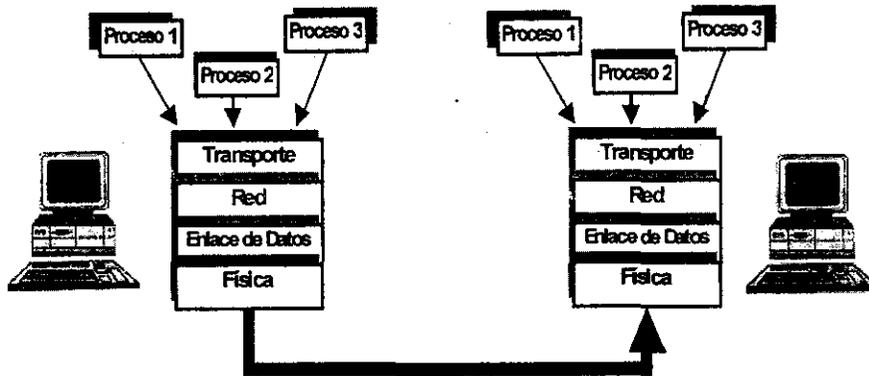


Figura 2.2.3.5 Capa de Transporte

### 5. Capa de Sesión.

El control de los diálogos entre distintos nodos es tarea de la capa de sesión. Un diálogo es una conversación formal en la que dos nodos acuerdan un intercambio de datos. La comunicación puede producirse en tres modos de diálogo (figura 2.2.3.6).

**Simple (Simplex).** Un nodo transmite de manera exclusiva mientras otro recibe de manera exclusiva.

**Semidúplex (Half-duplex).** Un solo nodo puede transmitir en un momento dado, y los nodos se turnan para transmitir.

**Dúplex total (Full-duplex).** Los nodos pueden transmitir y recibir simultáneamente. La comunicación dúplex total suele requerir un control de flujo que asegure que ninguno de los dispositivos envía datos a mayor velocidad de la que el otro dispositivo puede recibir.

Las sesiones permiten que los nodos se comuniquen de manera organizada. Cada sesión tiene tres frases:

1. **Establecimiento de la conexión.** Los nodos establecen contacto. Negocian las reglas de comunicación incluyendo los protocolos utilizados y los parámetros de comunicación.
2. **Transferencia de datos.** Los nodos inician un diálogo para intercambiar datos.
3. **Liberación de la conexión.** Cuando los nodos no necesitan seguir comunicados, inician la liberación ordenada de la sesión.

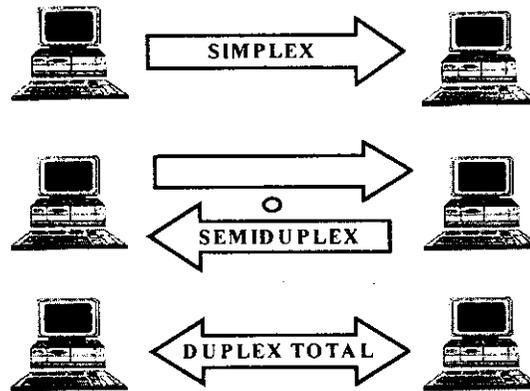


Figura 2.2.3.6 Tipos de Comunicación en la capa de Sesión

#### 6. *Capa de Presentación.*

La capa de presentación se responsabiliza de presentar los datos a la capa de aplicación. En ciertos casos, la capa de presentación traduce los datos directamente de un formato a otro. Por ejemplo, si se transmiten datos de una computadora EBCDIC a otra ASCII, la capa de presentación podría encargarse de traducir de un conjunto de caracteres al otro. Las normas OSI definen la Abstract Syntax Representation, Revision 1 (ASN.1 – Representación de sintaxis abstracta, revisión 1) como sintaxis estándar para los datos a nivel de la capa de presentación.

Otras funciones que pueden corresponder a la capa de presentación son, la encriptación / desencriptación y compresión / descompresión de datos.

### **7. Capa de Aplicación.**

La capa de aplicación proporciona los servicios utilizados por las aplicaciones para que los usuarios se comuniquen a través de la red. La siguiente lista enumera varios ejemplos de servicios.

**Transporte de correo electrónico.** Gran variedad de aplicaciones pueden utilizar un protocolo para gestionar el correo electrónico. Los diseñadores de aplicaciones que recurren al correo electrónico no necesitan desarrollar sus propios programas para gestionar el correo.

**Acceso a archivos remotos.** Las aplicaciones locales pueden iniciar y controlar procesos en otros nodos.

**Directorios.** La red puede ofrecer un directorio de recursos, incluyendo nombres de nodos lógicos. El directorio permite que las aplicaciones accedan a los recursos de la red utilizando nombres en lugar de identificaciones numéricas abstractas.

**Administración de la red.** Los protocolos de administración de la red permiten que varias aplicaciones puedan acceder a la información administrativa de la red.

Es frecuente encontrar el término interfaz de programa de aplicación (API) asociado a los servicios de la capa de aplicación. Un API es un conjunto de reglas que permiten que las aplicaciones escritas por los usuarios puedan acceder a los servicios de un sistema de software. Los diseñadores de programas y protocolos suelen proporcionar varias API's para que los programadores puedan adaptar fácilmente sus aplicaciones y utilizar los servicios disponibles en sus productos.

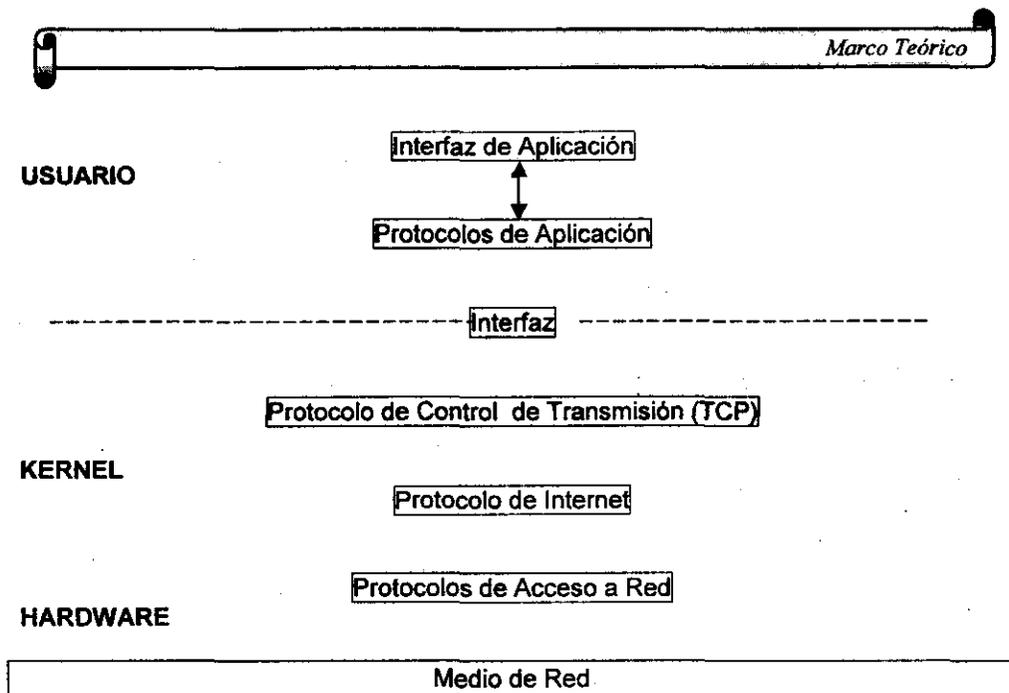
## 2.2.4 Arquitectura TCP/IP

El Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP), colectivamente conocido como TCP/IP, son los dos componentes primarios de un juego de protocolos que permiten comunicación entre *hosts* (servidores) en una misma o en redes diferentes. El juego completo, que también incluye un Protocolo de Datagramas (UDP) y un Protocolo de Resolución de Direcciones (ARP) es también conocido como *Suite* (Conjunto) del Protocolo TCP/IP.

Esta suite TCP/IP incluye un juego de protocolos de aplicación que soportan una gran variedad de servicios de usuario. Esas aplicaciones incluyen un Protocolo de Transferencia de Archivos (FTP), un Protocolo de Transferencia de Correo Simple (SMTP) y un Protocolo de Acceso a Terminal Remoto (TELNET).

Como otras arquitecturas de comunicaciones, los protocolos de TCP/IP pueden ser vistos como un grupo de capas. Cada capa recibe servicios de la capa inferior y provee servicios a la capa superior.

La figura 2.2.4.1 muestra la estructura de capas de TCP/IP.



**Figura 2.2.4.1 Estructura de Capas de TCP/IP**

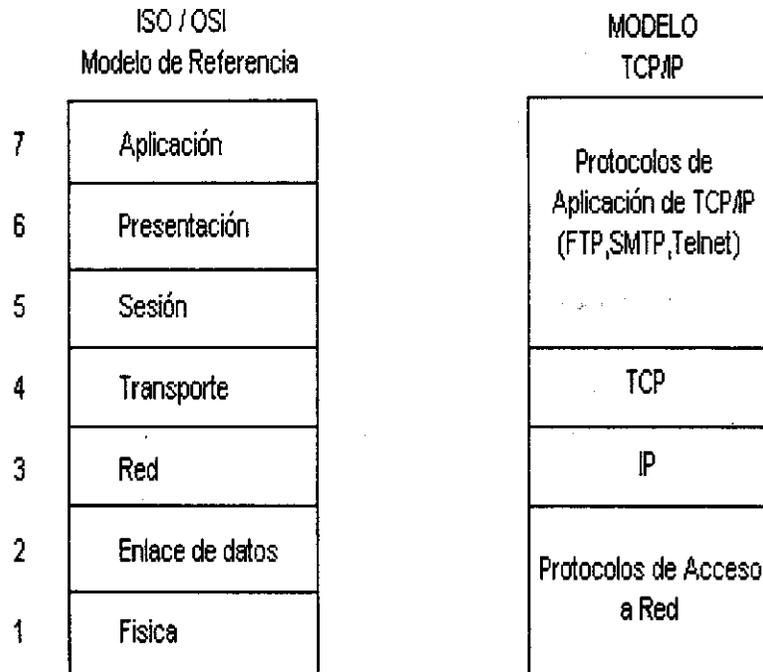
La siguiente lista describe brevemente las funciones de cada capa:

*Aplicación* Contiene aplicaciones, procesos o utilidades que inician la transferencia de datos sobre una red.

*TCP* Establece una comunicación dúplex y confiable entre dos hosts. TCP recibe mensajes de salida desde las capas de aplicación y añade información para asegurarse de que el mensaje llegue a la aplicación correcta en el host de destino. TCP también dirige mensajes de entrada a la aplicación correcta.

**IP** Toma los datos desde los protocolos de nivel superior y añade las direcciones de red y de host. La información añadida en el nivel de IP enruta los datos a través de una red hacia el host de destino.

**Acceso a Red** Da formato a los datos de TCP/IP para transmitirlos sobre una red física.



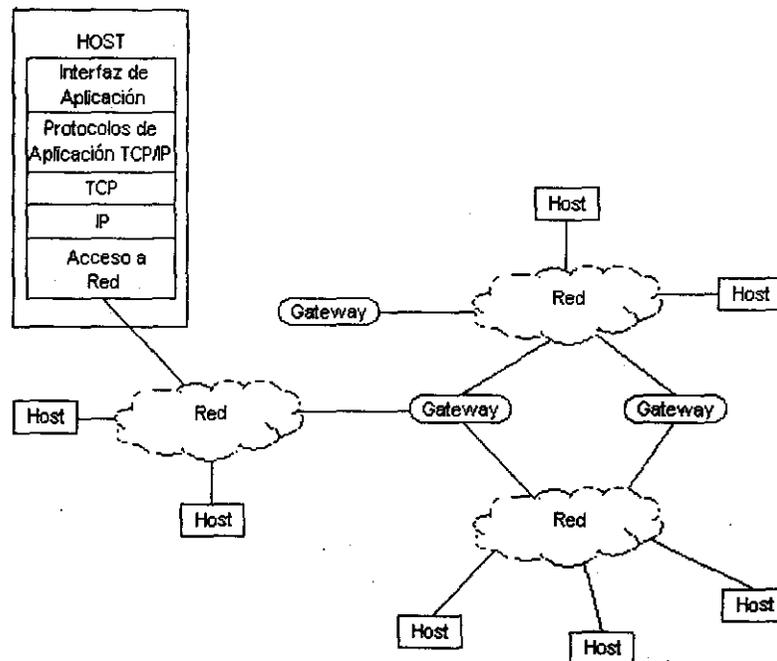
**Figura 2.2.4.2** Capas del Modelo ISO/OSI y de TCP/IP

Es muy común ver la suite del protocolo de TCP/IP mapeada al modelo de referencia de la Organización Internacional de Estándares/Interconexión de Sistemas Abiertos (ISO/OSI), aunque no exista una correlación directa entre protocolos. TCP/IP fue diseñado y utilizado varios años antes de que iniciaran los trabajos de los estándares de OSI.

Las figuras 2.2.4.2 y 2.2.4.3 revelan importantes aspectos de la arquitectura de TCP/IP: los protocolos de TCP/IP aíslan los procesos de aplicación de la red física. La red puede ser una Ethernet, una red X.25, o alguna otra red, pero desde el punto de vista del usuario, la comunicación toma lugar entre las aplicaciones.

Los protocolos de TCP/IP residen en cada host que se conecta a una red. Los **Gateways** (puentes) conectan redes separadas para formar lo que parece ser ante el usuario una red sencilla. Los gateways no requieren el juego completo de TCP/IP, únicamente aquellos elementos necesarios para rutear mensajes entre redes diferentes.

Una gran red como la Internet, contiene muchos gateways. Los mensajes se mueven de un gateway al siguiente hasta llegar al host destino.



**Figura 2.2.4.3 Red de ejemplo con varios gateways y hosts**

En este punto es necesario hacer una breve descripción del siguiente concepto, para con ello poder entender un poco mejor las ideas posteriores.

**Modelo Cliente-Servidor:** El modelo cliente-servidor nos ofrece un significado conveniente para el entendimiento de muchos procesos de comunicación. Muy sencillamente, un proceso en el cliente pide un servicio particular y el servidor de procesos responde. El proceso del cliente puede ser un programa de aplicación en una máquina; el proceso del servidor puede ser un programa de aplicación en otra. Ambos, el cliente y el servidor, transfieren datos sobre la misma red o sobre redes conectadas. En la mayoría de los casos, el proceso del servidor corre continuamente en la máquina servidor, esperando peticiones de los procesos del cliente. Algunos de los procesos más comunes del servidor incluyen servicios de archivos, servicios de impresión y servicios de cómputo.

### 2.2.5 Pasado y Presente

A principios de la década de los 60's, el Departamento de Defensa (DOD) se dio cuenta de que era necesario un protocolo estándar para soportar la comunicación entre una amplia variedad de computadoras y redes militares. Con los fundamentos provistos a través de la Agencia de Defensa y Proyectos de Investigación Avanzados (DARPA), varias universidades y corporaciones investigaron para crear los protocolos necesarios. El primer resultado de esta investigación fue una red de *switcheo* (intercambio) de paquetes conocido como ARPANET. Inicialmente, cada host conectado al ARPANET implementó un programa de control de red estándar (NCP), el cuál permitía que diferentes hosts se comunicaran en la misma red. Durante la década de los 70's, los esfuerzos fueron dirigidos a definir un estándar de redes que pudieran permitir a los hosts en diferentes redes comunicarse. TCP/IP fue el resultado de esos esfuerzos en 1983. TCP/IP reemplazó al NCP como el estándar de red para ARPANET, el cuál fue oficialmente delegado en 1990.

Hoy en día, muchas computadoras conectadas a una gran variedad de redes LAN y WAN utilizan la suite TCP/IP. Muchas de esas redes están conectadas para formar una gran red conocida como la Internet. Esas redes de Internet bajo el control de la DOD, tales como el MILNET, y varias otras redes clasificadas, son conocidas como la Red de Datos de la Defensa (DDN).

#### 2.2.5.1 Servicios en Modo Conexión y en Modo Sin Conexión

El servicio sin conexión, también conocido como servicio de datagrama, transfiere mensajes entre máquinas sin confirmación de entrega. En otras palabras, un mensaje puede ser descartado por alguna razón, sin notificación del emisor o del receptor.

La oficina postal provee una buena analogía de comunicación sin conexión. Una carta puede ser enviada desde un buzón local hacia cualquier buzón destino cuya dirección sea conocida. La oficina postal hace lo mejor para asegurar la entrega, pero ello no garantiza que la carta alcance su destino. Similarmente, no hay indicación explícita para el emisor o el receptor de que la transmisión sea exitosa o fallida.

La comunicación sin conexión algunas veces es clasificada como un servicio no confiable. Sin embargo, esto no es tan malo como podría parecer. El servicio sin conexión involucra un mínimo de **overhead** (encabezado extra redundante) y trabaja bien para intercambio de relativamente pequeñas cantidades de datos cuando el medio principal es confiable. Es muy bueno también para aplicaciones tales como voz digital, en donde la pérdida de pequeñas cantidades de datos es aceptable.

En contraste, el servicio en modo de conexión provee comunicación confiable a través de un proceso de tres fases: en la primera, se establece la conexión, en una segunda ocurre la transferencia de datos y en la tercera ocurre la desconexión. Durante la fase de establecimiento de conexión, las características de la misma son negociadas. Los datos se mueven sobre un canal de comunicación confiable, dúplex y de control de flujo durante la fase de transferencia de datos. Confiabilidad significa que la duplicidad, pérdida o secuencia inapropiada de unidades de datos son detectadas y, si es posible, corregidas por el prestador del servicio de red. La fase de desconexión, exitosa (sin pérdida de datos) o abortiva, cierra la conexión.

El teléfono provee una analogía de servicio de modo de conexión. Una llamada toma lugar desde un teléfono local hacia un teléfono destino. Una vez que la conexión se ha establecido, la conversación transfiere datos en ambas direcciones sobre la conexión. Si el dato se pierde, una o ambas partes son notificadas con la acción de recuperación apropiada. Finalmente, la

desconexión ocurre cuando ambas partes están de acuerdo en terminar la conexión.

La comunicación orientada a conexión requiere de mayor overhead que la comunicación sin conexión. El prestador de servicio básico de red debe mantener información adicional acerca del estado de cada conexión, y unidades de datos supervisores adicionales deben ser transmitidos para mantener la consistencia del estado de la información. La comunicación orientada a conexión también requiere de algoritmos sofisticados para asegurarse del uso más efectivo de los recursos de la red.

#### **2.2.5.2 Direccionamiento**

El direccionamiento es un aspecto integral de la arquitectura de TCP/IP. Cada red tiene su propia dirección. Cada host conectado a la red tiene una dirección que es única dentro de una red y una dirección física que es única en todas las redes. Cada aplicación corriendo en un host tiene una dirección única.

Las direcciones pueden ser divididas en tres tipos diferentes:

- Direcciones de aplicación (puertos)
- Direcciones de Internet
- Direcciones de red físicas

Las direcciones de aplicación, comúnmente referidas como puertos, especifican la última fuente o el destino del mensaje. El acceso a un programa de aplicación particular ocurre a través de un número de puerto. En la parte del cliente, los programas de aplicación corren a través de números de puerto aleatorios. El programa de aplicación del cliente debe especificar un número de aplicación de puerto específico en el host servidor. Generalmente, cada host

tiene varias aplicaciones con números de puerto únicos y universales. El protocolo de transferencia de archivos (FTP), por ejemplo, tiene un número de puerto 21. Las aplicaciones con números de puerto específicos, como FTP, esperan para proveer servicio a las peticiones del usuario.

Como una breve ilustración de puertos, podemos considerar a un usuario en la máquina Cliente A que busca transferir un archivo a la máquina Servidora B (ver la figura 2.2.5.1). El usuario utiliza un comando de transferencia de archivos en la máquina A. El comando establece comunicación con un número de puerto aleatorio en la máquina A, por ejemplo, el 1019. La máquina A envía una petición hacia un proceso de transferencia de archivos en el servidor, en un puerto bien conocido en la máquina B. (Un puerto bien conocido, también llamado reservado, es un puerto de TCP/IP asignado permanentemente a un servicio.) Cuando el proceso del servidor en la máquina B acepta la petición de transferencia de archivos, se crea una nueva conexión en la máquina B para procesar la transferencia de archivos, así, libremente, el proceso del servidor (el puerto bien conocido) está atento para peticiones adicionales.

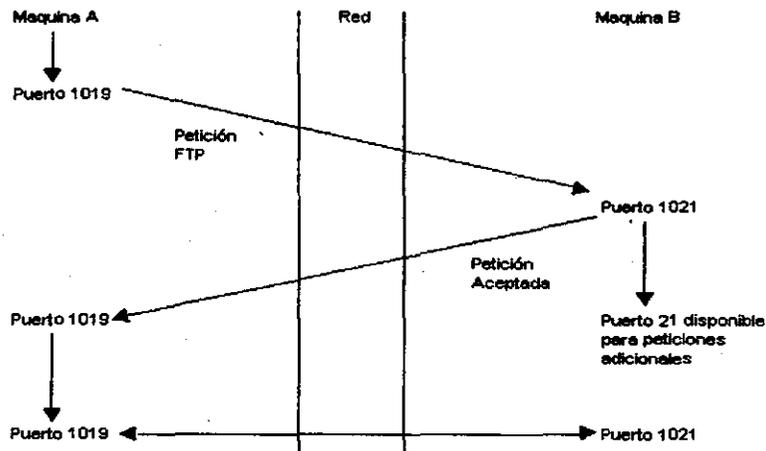


FIGURA 2.2.5.1 Ejemplo del proceso de utilización y asignación de puertos

La dirección de una Internet, también conocida como dirección IP, identifica una red particular y un host conectado a esa red. Las direcciones de Internet tienen dos componentes: una dirección de red y una dirección de host. Un campo de dirección de 32 bits en el encabezado de IP identifica cada host y su red. La dirección de Internet es expresada generalmente como cuatro números decimales separados por puntos decimales (notación de punto), por ejemplo 97.0.0.0. La combinación de una dirección de Internet y un número de puerto (dirección de la aplicación) es conocida como **socket** (ó conexión).

Las direcciones de Internet, definen tres clases de red: Clase A, Clase B y Clase C. Las redes clase A tienen direcciones de red de 8 bits, las de clase B tienen direcciones de red de 16 bits, y las de clase C tienen 24 bits de dirección. Los bits restantes de los 32 bits representan la parte de host de la dirección.

El tener en cero (0) el primer bit de orden mayor indica una red clase A. Por ejemplo, en la dirección 87.0.0.127, el primer bit es 0, lo cuál significa que el host está en una red clase A. El tener 10 en los primeros dos bits de orden mayor indican una red clase B. Finalmente, el tener 110 para los tres primeros bits de orden mayor indica una red clase C.

A continuación se muestran en decimal y en binario los rangos de cada clase de red.

Decimal	Binario	Clase
0 - 127	00000000 - 01111111	A
128 - 191	10000000 - 10111111	B
192 - 223	11000000 - 11011111	C

Las redes de clase A son utilizadas cuando se tienen muchos hosts conectados a la misma red. Los 24 bits en la parte de host de las redes clase A

proveen de un gran número de hosts diferentes. Contrariamente, una red clase C nos da un mayor número de direcciones de red únicas.

Las direcciones de red únicas especifican la dirección del hardware de red. Una dirección física es necesaria porque la transferencia de datos en una red toma lugar entre piezas de hardware. La dirección de Internet descrita con anterioridad debe estar mapeada a direcciones físicas antes de que los datos puedan ser transferidos entre aplicaciones.

### 2.2.5.3 Ruteo

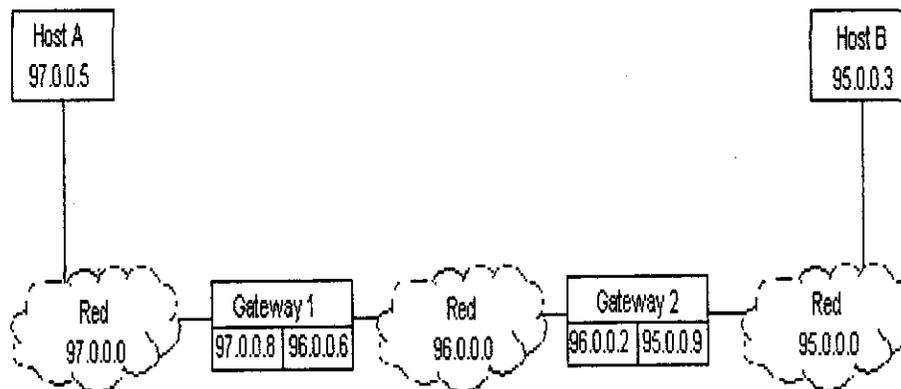
La comunicación de una Internet puede tomar lugar entre hosts en la misma o en redes separadas, pero siempre conectadas entre sí. En ambos casos, los datos viajan sobre la red como datagramas IP, encapsulados en *frames* (tramas). Los algoritmos y las tablas de ruteo contenidos en el software de IP de los hosts y de los gateways determinan la ruta que deben tomar los datagramas a través de la red.

El ruteo de los datagramas entre máquinas en la misma red es completamente directo. El software de IP en el host emisor compara la porción de red de su dirección de Internet con la del host receptor. Por ejemplo, dos hosts pueden tener la siguiente dirección de Internet:

97.0.0.3 (host emisor) y 97.0.0.5 (host receptor)

Como el 97 indica, ambos hosts están conectados directamente a la misma red. Consecuentemente, el host emisor coloca el datagrama en un frame y lo envía al host receptor. El ruteo de datagramas entre máquinas en redes separadas es más complejo, ya que los datagramas tienen que pasar por uno o más gateways hasta alcanzar su destino. Como se ilustra en la figura 2.2.5.2, los datagramas desde el host A hasta la dirección de Internet 97.0.0.5 deben

pasar a través de los gateways 1 y 2 para alcanzar el host B en la dirección de Internet 95.0.0.3. En este ejemplo, el software de IP del host A examina la dirección de red destino (95) y lo envía al gateway 1. Las tablas de ruteo en el gateway 1 determinan que el datagrama debe ser enviado hacia el gateway 2, donde podrá ser entregado a la red de destino.



**Figura 2.2.5.2 Ilustración del proceso de ruteo de datagramas**

#### 2.2.5.4 Servicio de Nombres

Cada máquina en una red TCP/IP tiene una dirección de Internet única. Utilizar un número para acceder a una máquina remota y usarla no es muy práctico. Un mejor método involucra el asignar a cada máquina un nombre simbólico. El protocolo de Servidor de Nombres de Dominio (DNS) convierte nombres simbólicos a su dirección de Internet correcta. DNS también convierte direcciones de Internet a nombres simbólicos.

### 2.2.5.5 Suite del Protocolo TCP/IP

A continuación ilustraremos la relación de las aplicaciones TCP/IP con las capas OSI.

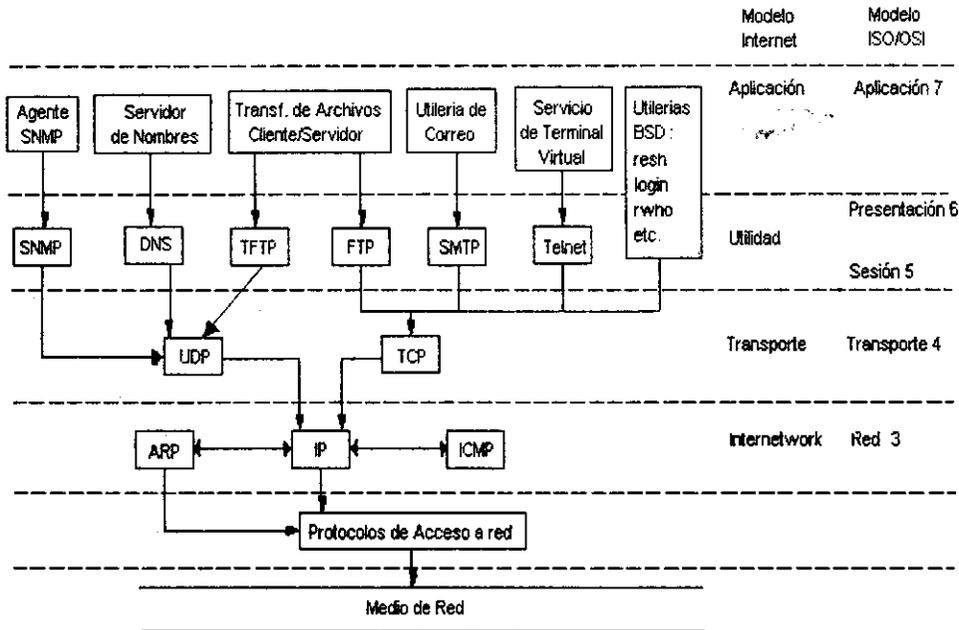


Figura 2.2.5.3 Relación de los Modelos ISO/OSI y TCP/IP

### 2.2.5.6 Aplicaciones y Utilidades

TCP/IP contiene utilerías que soportan transferencia de archivos, servicios de correo, servicio de nombres, emulación de terminal, etc. Esas utilerías incluyen los siguientes protocolos:

- Protocolo de Transferencia de Archivos (FTP)

- Protocolo de Transferencia de Archivos Trivial (TFTP)
- Protocolo de Transferencia de Correo Simple (SMTP)
- Protocolo de Terminal Virtual (TELNET)
- Protocolo de Manejo de Red Simple (SNMP)

La interacción con las utilerías anteriores ocurre a través de la interfaz del usuario.

#### 2.2.5.7 Protocolo de Control de Transferencia (FTP)

FTP permite a los archivos ser transferidos entre hosts conectados a una red común. El proceso de FTP es complejo en cuanto a que debe tratar con, entre otras cosas, diferentes sistemas operativos y estructuras de archivos. Esencialmente, el proceso de FTP puede ser descrito utilizando el modelo cliente - servidor. La máquina que inicia la conexión la llamaremos cliente. La segunda máquina involucrada en la conexión es el servidor. Una vez que la conexión ha sido establecida, la transferencia de archivos ocurre en ambas direcciones, servidor - cliente o cliente - servidor.

Los siguientes pasos describen el proceso de FTP a detalle:

1. Con el encendido del sistema, tanto el cliente como el servidor hacen que el servicio de FTP se encuentre disponible a través de un puerto bien conocido o reservado.
2. Un host inicia el proceso de FTP ya sea por un programa o por un usuario humano. Este host es llamado cliente.
3. El proceso de FTP establece una *conexión de control* entre el host cliente y el host servidor. El proceso puede requerir que el cliente provea un nombre de usuario y **password** (contraseña). Una vez que la conexión de acceso necesaria ha sido provista, el cliente puede desempeñar un

número limitado de operaciones, tales como listar los directorios y archivos del servidor.

4. El cliente solicita una transferencia de archivos. Esta petición abre una *conexión de datos*. La transferencia de datos ocurre sobre la conexión de datos.

5. Cuando la transferencia de datos se completa, la conexión de datos es desmantelada. La conexión de control, sin embargo, permanece intacta para peticiones de transferencia adicionales.

Como se indicó, ya sea un programa o una persona puede iniciar una transferencia de archivos utilizando procesos de FTP. En el siguiente ejemplo, el usuario Fred en la máquina A (el cliente) busca enviar el arch1 a la máquina B (el servidor). Fred ejecuta el comando **ftp maqB** en su máquina para establecer una conexión de control con la máquina B. Una vez que la conexión de control ha sido establecida, la máquina B solicita el *login* (nombre de usuario) de Fred y su password. Después de que Fred introduce su password correcto, su terminal despliega el prompt **ftp>**. Fred puede enviar el arch1 de la máquina A a la máquina B. También indica que el arch1 en la máquina A deberá ser llamado arch2 en la máquina B. Ejecutando el comando **send** (enviar), se establece una conexión de datos entre las máquinas. La transferencia de archivos ocurre sobre la conexión de datos. La reaparición del prompt **ftp>** indica que se ha completado la transferencia de archivos. Para salir del programa ftp, Fred teclea **quit** en el prompt de **ftp>**.

```
$ ftp maqB
connected to maqB
220 maqB FTP server ($Header: ch1b.doc 1.2 1991/06/06
18:33:29 $) ready.
Name (maqB:fred) : fred
331 Password required for fred.
Password (maqB:fred) :
230 User fred logged in.
ftp> send arch1 arch2
200 PORT command okay.
150 Opening data conection for arch2 (97.0.0.4 , 54795) .
226 Transfer complete.
152 bytes sent in 0.05 seconds (3 Kbytes/s )

ftp> quit
221 Goodbye.
$
```

#### 2.2.5.8 Protocolo de Transferencia de Archivos Simple (TFTP)

El protocolo de transferencia de archivo trivial (TFTP) provee un método simple de transferir archivos entre máquinas. Utiliza el protocolo de datagrama de usuario (UDP), el cuál es un servicio en modo sin conexión, como un mecanismo de transporte, y a menudo necesita un esquema de notificación/transmisión para asegurar una apropiada transferencia de datos. Los archivos son transferidos un bloque a la vez. Después de que cada bloque es transferido, la máquina emisora espera por una notificación de parte de la máquina receptora. TFTP no mantiene una conexión entre las máquinas, únicamente mantiene el rastro del host emisor.

Como con FTP, cualquier programa o usuario puede utilizar TFTP para transferencia de archivos. Sin embargo, TFTP no requiere una cuenta o password en la máquina remota, ya que únicamente archivos públicos son

accesibles para ser transferidos entre máquinas. Esto significa que un archivo en una máquina debe tener permisos de lectura y escritura para el grupo de **others** (otros) antes de poder ser transferidos a otra máquina que también tenga permisos de lectura y escritura para ese grupo.

Desafortunadamente, TFTP hace un uso ineficiente de la red. También presenta problemas de seguridad y debe ser utilizado solamente entre hosts confiables.

#### 2.2.5.9 Protocolo de Transferencia de Correo Simple (SMTP)

La transferencia de correo entre usuarios es una de las aplicaciones básicas y más ampliamente utilizadas en muchos sistemas de cómputo. Generalmente, cada usuario del sistema tiene su propio buzón. Utilizando algunas de las muchas interfaces de correo, los usuarios en la misma o en diferentes máquinas pueden fácilmente intercambiar mensajes o archivos entre buzones. El SMTP provee de un medio para transferir correo entre usuarios en el mismo sistema y, con los servicios ofrecidos por los protocolos de TCP/IP, a usuarios en sistemas separados. A continuación ejemplificamos una interfaz básica de correo. El emisor introduce el comando **mail** (correo) en el prompt, seguido del nombre de usuario del buzón receptor. El tema del mensaje es tecleado en el prompt **Subject:** . El texto del mensaje empieza en la siguiente línea. Al final de la porción del texto del correo, el emisor presiona la secuencia **Ctrl-D** o teclea un punto (.) en la primer columna de la línea siguiente. La interfaz de correo regresa la llave **Cc:** , en dónde los nombres que tecleemos recibirán una copia del mensaje. Al presionar **Return** después de introducir el último de estos nombres, envía el mensaje hacia el buzón de correo del receptor.

```
$ mail georgina
Subject: Cita
¿ Nos podemos ver a las 9:00 para discutir tu propuesta?
Cc: Alfredo
$
```

Para leer el contenido del correo, el receptor teclea únicamente el comando *mail*.

Cada mensaje de correo consiste de dos partes, un *encabezado* que contiene el nombre del emisor y del receptor, y el *cuerpo*, que contiene el texto del mensaje actual. El estándar RFC 822 especifica el formato exacto del mensaje de correo.

SMTP es un proceso complejo que maneja tanto correos de entrada como de salida. La siguiente secuencia describe el proceso de SMTP cuando el emisor y el receptor tienen buzones de correo en diferentes máquinas.

1. El emisor crea un mensaje de correo con una interfaz de usuario.
2. La interfaz del usuario coloca el mensaje de correo en una cola de salida. La cola de correo también contiene mensajes de correo de otros usuarios.
3. El servidor SMTP examina la cola a intervalos fijos.
4. El servidor SMTP selecciona un mensaje de la cola y establece una conexión a un puerto de TCP.
5. TCP establece una conexión con el receptor. Con una conexión establecida, los protocolos de SMTP en las máquinas emisoras y receptoras intercambian mensajes de identificación.
6. Una vez que las máquinas receptoras y emisoras se han identificado, la transferencia de mensajes de correo inicia. El receptor envía una

notificación al emisor después de que recibe cada una de las porciones del mensaje:

- Creador del Mensaje (From)
- Receptores del Correo (To: y Cc)
- Texto del Correo

7. El SMTP de la máquina emisora realiza un comando quit (q) (salir) al final del mensaje.

8. El SMTP de la máquina receptora notifica del comando de quit, y la máquina emisora cierra la conexión.

#### 2.2.5.10 Protocolo TELNET

Este protocolo permite a las aplicaciones y a las terminales en una máquina el establecer una conexión con aplicaciones y terminales en otra máquina. En otras palabras, TELNET actúa como una interfaz que maneja una variedad de características de aplicaciones y terminales. Con TELNET, ninguna máquina se tiene que conformar con las características de la otra máquina.

TELNET incluye las siguientes características:

*Terminal Virtual de Red (NVT):* TELNET define una terminal bien definida. Las máquinas en los extremos de la conexión de TELNET mapean la NVT en un formato apropiado para ellas mismas. Esto permite a máquinas disimilares comunicarse.

*Opciones de Negociación:* Proveen un juego expandido de opciones para procesos e interacción de terminales. Un proceso del cliente solicita una operación, y el servidor la acepta o la rechaza. De esta forma, el juego de opciones que utilizan las características de cada terminal o proceso se

establece. Esto permite la cooperación entre máquinas para proveer más servicios y características que las disponibles en la NVT.

*Conexión Simétrica* :Permite tanto a los procesos del cliente como del servidor operar como entidades diferentes. Ningún proceso se adapta a las características del otro proceso.

A continuación, mostramos un ejemplo de una sesión de TELNET.

```
$ telnet maqB
Trying ...
Connected to maqB.
Escape character is

Login: alf
Password:

$
```

#### 2.2.5.11 Protocolos de TCP/IP

Dentro de los principales protocolos de TCP/IP se encuentran los siguientes:

TCP,UDP,IP,ICMP,ARP y SNMP, los cuales describimos a continuación.

#### 2.2.5.12 Protocolo de Control De Transmisión (TCP)

El protocolo de control de transmisión (TCP) es el componente esencial de la comunicación en Internet. Es un protocolo complejo que define qué tan confiable es la transmisión, o si ocurre una transferencia orientada a conexión o de datos entre las aplicaciones del cliente y del servidor. El software que

implementa los servicios definidos por TCP generalmente residen en el sistema operativo de la computadora.

El software de TCP desempeña las siguientes tareas:

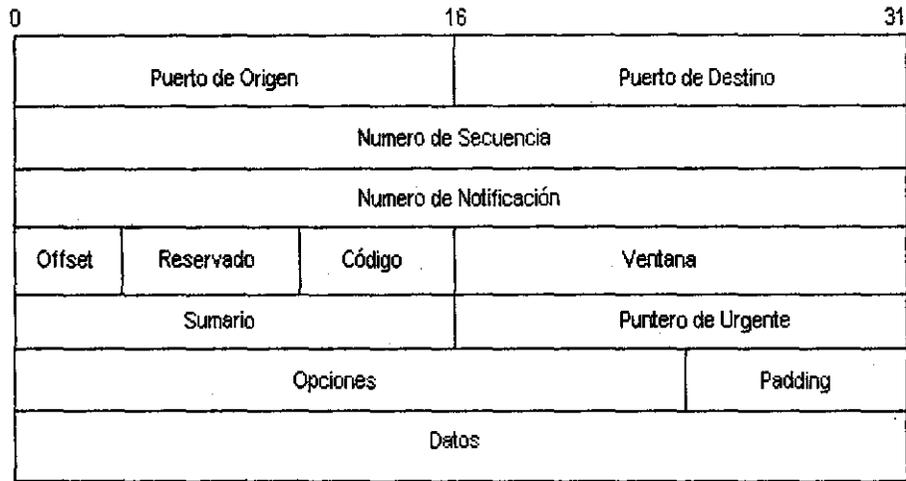
- Formatea los datos
- Establece y maneja una conexión entre el cliente y el proceso del servidor.
- Entrega datos y verifica errores.
- Direcciona datos hacia una aplicación particular (multiplexa)
- Desempeña control de flujo para asegurar el buen uso de los recursos de la máquina.
- Se adapta para cambiar el tráfico de la red y asegurar el mejor uso de la red.

Las utilerías de aplicación accesan el software de TCP en el sistema operativo a través de un mecanismo de interfaz. Los mensajes de o hacia una aplicación pasan a través de un puerto. Varias aplicaciones tienen una dirección de puerto bien conocida o reservada.

Los datos desde una aplicación pasan hacia TCP como una cadena continua de bits. Después, TCP divide la cadena de bits en piezas y añade un encabezado a cada pieza para formar segmentos. El software de TCP en el host destino reensambla los segmentos en cadenas de bits para su presentación en un puerto de aplicación.

Los segmentos también llevan la información necesaria para establecer, manejar y cerrar una conexión. Los segmentos de conexión no contienen un campo de datos.

A continuación se ilustra un segmento de TCP.



**Figura 2.2.5.4 Segmento de TCP/IP**

La siguiente lista describe cada campo de un segmento de TCP.

*Puerto de Origen:* Especifica la dirección del puerto de la aplicación cliente, en otras palabras, las direcciones de la aplicación que inicia la conexión o la transferencia de datos.

*Puerto de Destino:* Especifica la dirección del puerto de la aplicación del servidor.

*Número de Secuencia:* Especifica el número del primer byte (octeto) de datos en cada segmento de la cadena. Por ejemplo, el primer segmento puede contener bytes de datos del 1 al 100. El segundo segmento puede contener bytes de datos del 101 al 200. El primer segmento puede tener un número de secuencia de 1 y el segundo segmento puede tener un número de secuencia de 101. Cuando es activado el bit de sincronía

(SYN) en el campo de código, se convierte en el número de secuencia inicial (ISN). Con ello, el primer byte de datos toma el valor ISN +1.

*Número de Notificación:* Especifica el número del siguiente byte de datos que el cliente espera recibir.

*Offset:* Especifica el número de campos de 32 bits en el encabezado, con ello, la porción de encabezado de un segmento puede ser distinguida de la de datos. Esto es importante ya que el campo de opciones puede variar en longitud, variando el número de 32 bits.

*Reservado:* Reservado para uso futuro, con todos los bits puestos a cero.

*Código:* Especifica el uso y el contenido del segmento. El campo de código contiene los siguientes parámetros:

URG: Indica que el valor del campo de urgente es importante

ACK: Indica que el valor del campo de notificación es importante

PSH: Envía el segmento de datos desde el cliente inmediatamente hacia la aplicación del servidor.

RST: Resetea la conexión entre el cliente y el servidor

SYN: Sincroniza los números de secuencia

FIN: Indica que la aplicación emisora no tiene más datos

*Ventana:* Especifica el número de bytes de datos que el emisor espera aceptar. Este campo establece los parámetros de control de flujo.

*Sumario:* Utiliza un esquema de complemento a uno para asegurarse de la precisión de todas las palabras de 16 bits en el encabezado y en los datos, incluyendo un pseudoencabezado de 96 bits. Este pseudoencabezado protege al TCP de segmentos fuera de ruta.

*Puntero de Urgente:* Especifica el número del byte que sigue al dato urgente, informando al receptor cuántos datos urgentes por esperar. El valor de este campo es significativo únicamente cuando el campo de código URG ha sido activado.

*Opciones:* Especifica el tamaño de segmento máximo que el cliente o el servidor puede aceptar.

*Padding:* Un valor de cero asegura que el encabezado del datagrama de IP termina en un límite de 32 bits.

*Datos:* Contiene los datos del mensaje de aplicación.

Antes de que la transmisión de datos ocurra entre dos máquinas, una conexión debe ser establecida entre entidades de TCP en cada máquina. TCP utiliza un proceso de "3 pasos" para establecer la conexión. Primero, la entidad cliente de TCP envía un segmento con el campo de código SYN activo hacia la entidad TCP del servidor. La entidad del servidor responde con un segmento con los campos SYN y ACK activos. Finalmente, la entidad cliente de TCP envía un segmento con el bit ACK activo para informar al servidor que la conexión ha sido establecida.

### 2.2.5.13 Protocolo de Datagrama de Usuario (UDP)

Como TCP, UDP define un método para transferir datos de aplicación entre máquinas. El servicio de UDP sin embargo es en modo sin conexión, por ello, poco confiable. Así mismo, no provee ningún mensaje de secuencia o mecanismo de control de flujo, y no divide los mensajes en segmentos.

Como en TCP, las aplicaciones accesan al servicio de UDP a través de una interfaz. En contraste con TCP, el cuál acepta una cadena de bytes para transportarlos al destino, UDP trata cada bloque de los datos del usuario como una entidad separada. Añade un encabezado de UDP para permitir al destino saber qué aplicación deberá procesar los datos.

La siguiente lista describe los campos del UDP

*Puerto de Origen:* Especifica la dirección del puerto de aplicación del cliente; en otras palabras, la dirección de la aplicación que inicia la conexión o transferencia de datos. Este es un campo opcional utilizado solo cuando la máquina servidora o aplicación necesita una dirección de respuesta. El campo deberá ser puesto a cero cuando no sea utilizado.

*Puerto de destino:* Especifica la dirección de puerto del servidor de la aplicación.

*Longitud:* Especifica la longitud combinada del encabezado del UDP y de los datos.

*Sumario:* Utiliza un esquema de complemento a uno para asegurarse de la coherencia de todas las palabras de 16 bits en el

encabezado y en los datos, incluyendo un pseudoencabezado de 96 bits de IP, unido al encabezado del UDP.

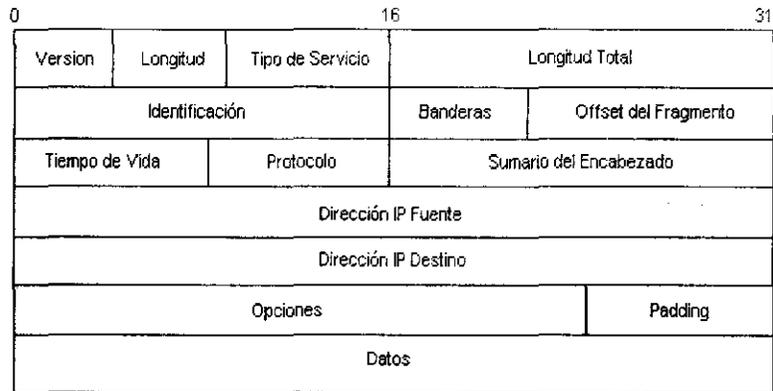
*Datos:* Contiene los datos del mensaje de la aplicación.

#### 2.2.5.14 Protocolo de Internet (IP)

Primeramente, IP se asegura de que el segmento de TCP o el datagrama del UDP llegue a la máquina correcta: lo cual no incluye el contenido del segmento o del datagrama. En otras palabras, IP lleva el datagrama a la máquina correcta y después, ya sea TCP o UDP se aseguran de que el datagrama llegue a la aplicación correcta en esa máquina. Como UDP, el servicio de IP es en modo sin conexión y poco confiable.

IP recibe el segmento de TCP o el datagrama de UDP y le añade su propio encabezado para crear un encabezado de IP. Los requerimientos para los protocolos de la capa de acceso a red requieren que un datagrama de IP sea dividido en *fragmentos*. Cada fragmento recibe su propio encabezado. El software de IP en la parte receptora reensambla los fragmentos para formar un datagrama completo.

A continuación ilustraremos un datagrama de IP:



**Figura 2.2.5.5 Datagrama de IP**

La siguiente lista describe los campos del datagrama IP:

*Versión:* Especifica el formato numérico del datagrama.

*Longitud:* Especifica la longitud del encabezado de IP en palabras de 32 bits.

*Tipo de Servicio:* Especifica la calidad del servicio que el datagrama deberá recibir mientras viaja a través de la red. Los bits en este campo pueden ser activados para especificar parámetros de **delay** (retardo), **throughput** (capacidad de la línea) y **reliability** (confiabilidad).

*Longitud Total:* Especifica la longitud total, encabezado y datos del datagrama de IP en bytes (octetos).

*Identificación:* Identifica el fragmento del datagrama. El emisor asigna un número a cada fragmento para ayudar al receptor en el reensamblado de los mismos en un datagrama sencillo.

*Banderas:* Especifica si el datagrama será fragmentado y si el fragmento es el último en una serie o si le seguirán más fragmentos.

*Offset del fragmento:* Especifica la posición del fragmento en el datagrama.

*Tiempo de vida:* Especifica el tiempo máximo que el datagrama permanece en el sistema antes de ser descartado.

*Protocolo:* Especifica qué protocolo de la capa superior recibirá los datos contenidos en el datagrama. Por ejemplo, un valor de 6 especifica que el dato será direccionado a TCP.

*Sumario del Encabezado:* Especifica el valor del sumario. Cada punto en la Internet que procesa el datagrama de IP recalcula y verifica el valor del sumario del encabezado.

*Dirección IP Fuente:* Especifica la dirección del host que inicia la transferencia del datagrama de IP.

*Dirección IP Destino:* Especifica la dirección del host que es el último receptor del datagrama de IP.

*Opciones:* Permite especificar una variedad de opciones, tales como seguridad, ruteo, etc.

*Padding:* Un valor de cero asegura que el encabezado del datagrama de IP termine en un límite de 32 bits.

*Datos:* Contiene el dato o fragmento de datos contenidos en el segmento de TCP o en el datagrama de UDP.

El software de IP entrega sus datagramas a la subcapa de acceso a red. El software de acceso a red encapsula el datagrama en un frame para transferirlo sobre la red física.

#### **2.2.5.15 Protocolo de Internet de Control de Mensaje (ICMP)**

Los datagramas que viajan sobre la red pueden experimentar varios problemas. Los errores pueden ser detectados mientras el software del gateway procesa el datagrama. Retardos en el gateway pueden deberse a congestiones en la red. En suma, un gateway puede ser incapaz por alguna razón de entregar el datagrama al host de destino o al siguiente gateway. Hay que recordar que, en el nivel de IP, el servicio provisto es sin conexión y poco confiable. Cada componente de la red, host o gateway provee su servicio sin informar al componente previo. A pesar de ello, algún método debe ser usado para pasar información entre componentes de red.

El ICMP, una parte necesaria del módulo de IP, especifica un mecanismo que permite a los gateways y hosts intercambiar información. El mensaje de ICMP se mueve a través de la red en porciones de datos de un datagrama de IP.

#### **2.2.5.16 Protocolo de Resolución de Direcciones (ARP)**

La transferencia de datos sobre una red ocurre entre dispositivos de hardware. Muchos de ellos emplean esquemas de direcciones diferentes al utilizado por las direcciones de Internet de 32 bits. Ethernet, por ejemplo, utiliza direcciones de 48 bits.

El protocolo de resolución de direcciones (ARP) provee de un mecanismo para mapear o convertir direcciones de Internet a direcciones físicas. El mecanismo de ARP en el host emisor censa otros hosts para determinar la dirección física del host destino. Los resultados del censo son almacenados en el caché del ARP. Cuando el host está listo para enviar sus datos, examina el caché para conocer la dirección física de la máquina de destino. Los datos de TCP/IP son colocados en un frame o paquete con la dirección de destino física. Finalmente, el paquete o frame viaja a través de la red hacia el host destino.

#### **2.2.5.17 Protocolo de Manejo de Red Simple (SNMP)**

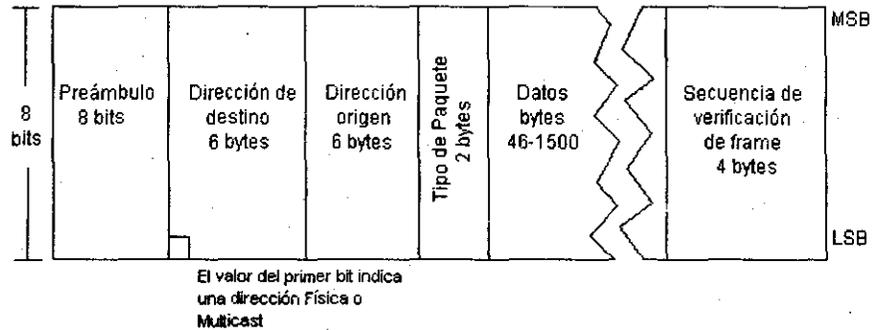
Las actividades de SNMP consisten en recopilar, analizar y ofrecer datos sobre el rendimiento de los componentes de la red. Los datos recopilados por SNMP incluyen estadísticas de rendimiento, informes rutinarios y avisos de problemas potenciales o existentes de la red. SNMP pertenece a la familia de protocolos Internet para la administración de la red.

#### **2.2.5.18 Protocolos de Acceso a Red**

Los protocolos de acceso a red especifican la forma en que los datos de las capas de TCP/IP deben ser procesados para su transmisión sobre una red física específica. La red puede ser una LAN o una WAN. Las LAN incluyen Ethernet, DECnet, etc. Muchas WAN implementan los servicios definidos por los protocolos de X.25.

El software que implementa los protocolos de acceso a red coloca los datos de TCP/IP en un *paquete* para su transmisión sobre la red. El paquete, o también llamado *frame*, es la unidad básica de transmisión entre los componentes de hardware en la red.

Ethernet provee un ejemplo de un protocolo de acceso a red común. Las capas físicas y de liga de datos de Ethernet proveen de servicios que dan formato a los datos desde la capa del cliente para formar los paquetes. A continuación ilustramos el formato de un paquete Ethernet.



**Figura 2.2.5.6 Un paquete Ethernet**

La longitud del paquete Ethernet puede variar de 72 a 1526 bytes. La variación en la longitud del paquete puede atribuirse al campo de datos, ya que los otros campos tienen valores fijos. A continuación describimos brevemente cada campo del paquete.

**Preámbulo:** El preámbulo consiste de una serie de unos (1) y ceros (0) que la estación receptora del paquete utiliza para sincronización.

**Dirección de Destino:** El campo de dirección de destino contiene la dirección de la estación o estaciones receptoras del paquete. Esto se determina a través del primer bit menos significativo (LSB) del campo de dirección, la estación transmisora del paquete puede seleccionar una dirección física o multicast (múltiple alcance). Un bit cero en la primera posición indica una

dirección física de una estación en la red. Cada estación en una LAN Ethernet tiene una dirección física única. ( $2^{47}$  permite hasta 147 trillones de direcciones físicas Ethernet).

Un bit uno en la primera posición indica una dirección multicast. Multicast significa que la dirección destino aplica a más de una estación en la red. En otras palabras, el paquete transmitido puede ser direccionado a más de una estación. Hay dos tipos de direcciones multicast: *direcciones de grupo multicast* y *direcciones broadcast*. Las estaciones con la misma dirección de grupo multicast pueden recibir y procesar cualquier paquete con esa dirección. Las direcciones broadcast permiten a todas las estaciones en una red simple recibir y procesar el mensaje transmitido. Al poner todos los bits en el campo de dirección de destino a 1 indica una dirección broadcast.

*Dirección Origen:* Este campo contiene la dirección física de la estación transmisora del paquete.

*Tipo de Paquete:* Una LAN Ethernet puede llevar datos de una gran variedad de protocolos de nivel superior: X.25, SNA, TCP/IP. Este campo especifica el formato del paquete de datos de acuerdo a los protocolos de nivel superior y le indica la forma de procesar los datos.

*Datos:* El campo de datos contiene los datos actuales que una estación desea transmitir a otra.

*Secuencia de Verificación del Frame:* Este frame de secuencia utiliza la puesta de la dirección de destino, dirección de origen, tipo de paquete y campos de datos para crear un valor cíclico de verificación de redundancia (CRC). La estación receptora del paquete utiliza el valor CRC para verificar errores de transmisión en el medio.

### 2.2.5.19 Protocolos de Acceso a Subred (SNAP)

El protocolo de acceso a subred (SNAP) es una parte de un estándar propuesto para la transmisión de datagramas del protocolo de IP sobre redes IEEE 802. Típicamente, los datagramas de IP son enviados directamente sobre una LAN Ethernet. Para aquellas instalaciones donde se tiene implementada una LAN IEEE 802, SNAP, en conjunción con la subcapa de liga de datos LLC (Control de liga lógica), proveen de un medio para que los datagramas de IP viajen a través del medio transparentemente.

Algunas implementaciones de SNAP vienen del RFC 1042. Este RFC especifica un protocolo estándar para la comunidad Internet. Describe un método estándar de encapsular los datagramas de IP y las peticiones del protocolo de resolución de direcciones ARP y da respuestas a redes IEEE 802. Este estándar de protocolo permite implementaciones de compatibilidad e interoperabilidad para transmitir datagramas de IP y peticiones y respuestas de ARP.

### 2.2.5.20 Una Red TCP/IP de Ejemplo

La transferencia de un archivo sobre una red ayudará a ilustrar cómo TCP/IP y los varios protocolos de acceso a red trabajan juntos. La siguiente figura ilustra dos hosts en dos redes separadas. Una tercera red separa las dos redes en que residen los hosts. Los gateways 1 y 2 proveen de un camino entre las redes individuales.

La siguiente secuencia de pasos describe la transferencia de un archivo del host cliente 1 al host server 2.

1. El usuario en la terminal del host cliente 1 ejecuta un comando **ftp** para establecer una conexión de control con el host server 2.

2. La aplicación FTP abre un puerto en el módulo del software de TCP. El puerto tiene un número único.
3. El software de TCP crea un encabezado de TCP con un valor de 21 en el campo de dirección destino y el número de puerto único en el campo del puerto origen. En este punto, el host cliente intenta únicamente establecer una conexión con el host server, por ello, ningún dato viaja con el encabezado de TCP.
4. El encabezado de TCP pasa hacia la capa de IP, donde un encabezado de IP es añadido para formar un datagrama de IP. El campo de dirección destino de IP en el encabezado contiene la dirección IP del host servidor 2. El software de IP examina la dirección de red destino del datagrama y determina que el datagrama de IP debe ser enviado hacia el gateway 1.
5. El datagrama de IP pasa hacia las capas del protocolo de acceso a red, donde es encapsulado en un frame en la red 1. El encabezado del frame contiene la dirección física del gateway 1.
6. El frame viaja a través de la red 1 hacia el gateway 1. El software de acceso a red en el gateway 1 remueve el encabezado del datagrama IP del frame y lo pasa a la capa de IP. El software de IP determina que el datagrama de IP debe ser ruteado hacia el gateway 2. Entonces el datagrama pasa hacia el software del protocolo de acceso a red para encapsularlo en un frame en la red 2. El encabezado del frame contiene la dirección física del gateway 2.
7. El frame viaja a través de la red 2 hacia el gateway 2. Siguiendo el procedimiento anteriormente explicado en el paso 6, el software en el gateway 2 determina que el datagrama IP debe ser ruteado hacia la

red 3. El software del protocolo de acceso a red encapsula el datagrama en un frame en la red 3. El encabezado del frame contiene la dirección física del host server 2.

8. El frame pasa a través de la red 3 hacia el host server 2. El software del protocolo de acceso a red remueve el encabezado del frame y pasa el datagrama de IP hacia el software de IP.
9. Una vez que la conexión de control ha sido establecida, el usuario se conecta con el host server. Con una conexión exitosa, la terminal del usuario del host cliente 1 despliega el prompt **ftp>**. Este prompt indica que una conexión de datos puede ser establecida entre los hosts servidor y cliente.
10. El usuario introduce el comando **send** seguido de dos nombres de archivos. El primer nombre es el del archivo que será transferido hacia el servidor. El segundo es el nombre que el archivo transferido tendrá en el servidor host.
11. El contenido del archivo del host cliente pasa hacia la aplicación del software de TCP como un mensaje en cadena de bits. El software de TCP rompe el mensaje en piezas y añade un encabezado de TCP para cada pieza y con ello poder formar un segmento de TCP.
12. Los segmentos individuales pasan a través de las capas inferiores y a través de redes y gateways hacia el host servidor en la forma usada para transferir el encabezado de TCP entre hosts (pasos 3 al 9).
13. En el nivel de TCP del servidor, los segmentos individuales son reensamblados y pasados como una cadena de bits hacia el nivel de aplicaciones para colocarlos en un archivo.

14. Una vez que se completó la transferencia del archivo, el usuario recibe una vez más el prompt **ftp>** en la terminal del host cliente. Tecleando **quit** en el prompt se termina la conexión entre las dos máquinas.

La secuencia arriba descrita presenta una vista de alto nivel de la transferencia de un archivo utilizando TCP/IP y sus aplicaciones.

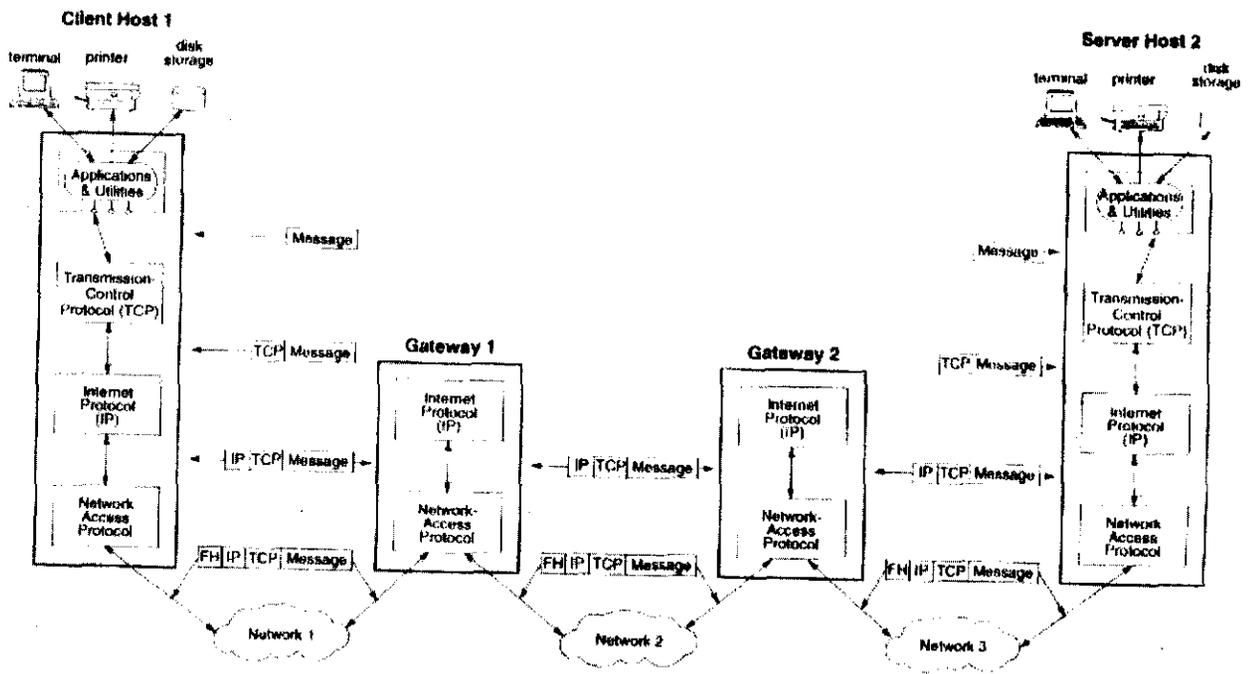


Figura 2.2.5.7 Red de Ejemplo. Envío de un mensaje

### 2.3. Bases de Datos Relacionales (Características, Ventajas y Desventajas)

Una base de datos relacional es aquella cuyos usuarios la perciben como un conjunto de tablas y nada más que tablas.

El modelo relacional es una forma de ver los datos; es decir, una receta para representar los datos (mediante tablas) y la receta para manipular esa representación (mediante operadores como el de unión). En términos más precisos, el modelo relacional se ocupa de tres aspectos de los datos: su estructura, su integridad y su manipulación. Cada una de las tres partes tiene sus propios términos especiales, y los más importantes en relación con la estructura de los datos se muestra en la siguiente figura, utilizando la relación S de una base de datos de ejemplo, llamada PROVEEDORES. Los términos en cuestión son *relación*, *tupla*, *cardinalidad*, *atributo*, *grado*, *dominio* y *llave primaria*. A continuación describiremos de forma informal estos términos, para pasar posteriormente a una más formal:

Una *relación* corresponde a lo que llamamos una tabla

Una *tupla* corresponde a una fila de esa tabla y un *atributo* a una columna. El número de tuplas se denomina *cardinalidad* y el número de atributos se llama *grado*.

La *llave primaria* es un identificador único para la tabla; es decir, una columna o combinación de columnas con la siguiente propiedad: nunca existen dos filas de la misma tabla con el mismo valor en esa columna o combinación de columnas.

Por último, un *dominio* es una colección de valores, de los cuales uno o más atributos (columnas) obtienen sus valores reales. Por ejemplo, el dominio marcado con S# en la figura es el conjunto de todos los números de proveedores legales, y el conjunto de valores que aparecen en el atributo S# de la relación S en cualquier momento es algún subconjunto de ese conjunto. De manera similar, el conjunto de

valores que aparecen en el atributo S# de la relación SP en cualquier momento dado es también subconjunto de ese conjunto.

### **Dominio**

Partiremos desde lo que es considerada como la menor unidad semántica de información, la cuál suponemos es el valor de un dato individual. Llamaremos a estos valores *escalares* (aunque este término no se utiliza mucho en la literatura relacional). Los valores escalares representan "la menor unidad semántica de información" en el sentido de que son atómicos, esto es, no poseen estructura interna (es decir, no se pueden descomponer) desde el punto de vista del modelo. Cabe subrayar que la carencia de estructura interna desde el punto de vista del modelo no implica la falta de estructura interna en términos absolutos. Por ejemplo, un nombre de ciudad tiene sin duda una estructura interna (formado por una secuencia de letras); sin embargo, si descomponemos un nombre de ciudad en sus letras componentes, perdemos significado. El significado solo se hace evidente si las letras aparecen todas juntas y en la secuencia correcta. Con lo anterior, los dominios son fondos de valores de los que se extraen los valores reales que aparecen en los atributos. Cada atributo debe estar definido sobre un dominio subyacente, y sólo uno, lo cual significa que los valores de ese atributo deben proceder de ese dominio. Cabe señalar que en cualquier momento será común encontrar en un dominio determinado valores que no aparecen de momento en ninguno de los atributos correspondientes a ese dominio.

Con lo anterior podemos responder a la pregunta "¿cuál es la importancia de los dominios?": los dominios restringen las comparaciones, como explicamos enseguida:

Consideremos las siguientes dos consultas en SQL sobre la base de datos de proveedores y partes:

```
SELECT P. *, SP.*  
FROM P, SP  
WHERE P.P# = SP.P# ;
```

```
SELECT P. *, SP. *  
FROM P, SP  
WHERE P.PESO = SP.CANT;
```

De estas dos consultas, la de la izquierda con toda probabilidad es razonable, pero la de la derecha probablemente no lo es. Desde un punto de vista formal, la consulta de la izquierda incluye una comparación entre dos atributos, P.P# y SP.P# definidos sobre el mismo dominio, en tanto que la de la derecha incluye una comparación entre dos atributos, P.PESO y SP.CANT definidos (al parecer) sobre dominios distintos, y decimos que "al parecer" porque, si bien es cierto que tanto los pesos como las cantidades son números, son diferentes tipos de números. No tiene mucho sentido comparar un peso y una cantidad. Es de suponer que los dominios de pesos y cantidades son distintos, aunque las dos consultas son legales en SQL.

Por tanto, si dos atributos toman sus valores dentro del mismo dominio, es probable que tengan sentido las comparaciones (y todas las demás operaciones) donde estén implicados esos atributos, porque se comparan cosas similares. En cambio, si los dos atributos obtienen sus valores de diferentes dominios, las operaciones donde estén implicados esos dos atributos carecerán de sentido. Con ello, una ventaja del manejo de dominios por partes de un sistema es que lo hace capaz de impedir errores de los usuarios. Si el sistema tratara de ejecutar una operación que incluyera una comparación entre dominios distintos, el sistema podría interrumpirla e indicar al usuario el posible error.

Los dominios pueden estar o no almacenados de manera explícita en la base de datos como conjuntos reales de valores, pero en la mayor parte de los casos no será así. Sin embargo, deberán al menos especificarse como parte de la definición de la base de datos; y después cada definición de atributo debería incluir una referencia al dominio correspondiente, con objeto de que el sistema pueda saber cuáles atributos son comparables entre sí y cuales no.

El nombre de un atributo dado puede ser igual al del dominio correspondiente o puede ser un nombre distinto. Es obvio que el nombre deberá ser distinto, ya que de no ser así, daría pie a una ambigüedad. Pero en términos generales es conveniente dar a los atributos nombres iguales a los del dominio subyacente cuando sea posible, o al menos incluir el nombre de ese dominio como la parte final (por ejemplo) de su propio nombre.

Por lo anterior, tener un recurso así implica mucho más que solo las comparaciones restringidas por los dominios. Es mucho más complejo de lo que parece a primera vista:

- En primer lugar, bien podrían presentarse situaciones en las cuales tuvieran sentido ciertas comparaciones entre atributos con diferentes dominios.
- En segundo lugar, no todas las operaciones de comparación tienen siempre sentido, aún dentro del mismo dominio. Consideremos, por ejemplo, el dominio SEXO, con los valores "Femenino" y "Masculino". Para nosotros es evidente que las únicas operaciones de comparación aplicables a este dominio son "igual a" y "diferente de"; "menor que" no tiene sentido.
- En tercer lugar, es obvio que debe ser posible (en los casos apropiados) combinar valores de diferentes dominios mediante diversos operadores aritméticos o de otro tipo. Por ejemplo, si multiplicamos un valor VELOCIDAD por un valor de TIEMPO, obtendremos un valor de DISTANCIA.

Por ello, un manejo de dominios requeriría por lo menos todo lo anterior, pero debe entenderse que ésta es una suposición muy amplia.

## Relaciones

Una *relación* (por ejemplo  $R$ ) sobre un conjunto de dominios  $D_1, D_2, \dots, D_n$  (no necesariamente todos distintos), se compone de dos partes, una *cabecera* y un *cuerpo*.

La *cabecera* está formada por un conjunto fijo de *atributos* o, en términos más precisos, de pares atributo - dominio,  $\{ (A_1:D_1), (A_2:D_2), \dots, (A_n:D_n) \}$  tales que cada atributo  $A_j$  corresponde a uno y solo uno de los dominios subyacentes  $D_j$  ( $j=1, 2, \dots, n$ ).

El *cuerpo* está formado por un conjunto de *tuplas*, el cual varía con el tiempo. Cada tupla a su vez está formada por un conjunto de pares atributo - valor:

$$\{ (A_1:v_{i1}), (A_2:v_{i2}), \dots, (A_n:v_{in}) \}$$

( $i=1, 2, \dots, m$ , donde  $m$  es el número de tuplas (filas o registros) del conjunto)

En cada una de estas tuplas hay uno de estos pares atributo - valor  $(A_j:v_{ij})$  para cada atributo  $A_j$  de la cabecera. Para cada par atributo - valor  $(A_j:v_{ij})$ ,  $v_{ij}$  es un valor del dominio único  $D_j$  asociado al atributo  $A_j$ .

Los valores  $m$  y  $n$  se llaman *cardinalidad* y *grado* respectivamente, de la relación  $R$ . La cardinalidad varía con el tiempo pero el grado no.

### Propiedades de las Relaciones

Las relaciones poseen ciertas propiedades, todas ellas consecuencia inmediata de la definición de relación presentada anteriormente, y todas ellas muy importantes. Primero expresaremos esas propiedades. Las propiedades son cuatro. Dentro de una relación dada:

- No existen tuplas repetidas
- Las tuplas no están ordenadas (de arriba hacia abajo)
- Los atributos no están ordenados (de izquierda a derecha)
- Todos los valores de los atributos son atómicos

#### 1. No existen tuplas repetidas

Esta propiedad es consecuencia del hecho de que el cuerpo de la relación es un conjunto matemático, y en matemáticas los conjuntos por definición no incluyen elementos repetidos.

Esta primera propiedad sirve para ilustrar la diferencia entre una relación y una tabla, porque una tabla podría contener filas repetidas, pero una relación no puede contener tuplas repetidas.

Un corolario importante de la ausencia de tuplas repetidas es que siempre existe una *clave primaria*. Como las tuplas son únicas, es evidente que por lo menos la combinación de todos los atributos de la relación tienen la propiedad de unicidad, de modo que al menos la combinación de todos los atributos puede servir (en caso necesario) como clave primaria. Desde luego, en la práctica casi nunca resulta necesario implicar a todos los atributos. En la definición formal de clave primaria se especificará la exclusión de cualquier atributo superfluo para el propósito de implicación única. La clave primaria sí puede ser compuesta, puede incluir varios atributos simples.

#### 2. Las tuplas no están ordenadas

Esta propiedad también se desprende del hecho de que el cuerpo de una relación es un conjunto matemático. Los conjuntos en matemáticas no son ordenados. Por lo anterior no es posible hablar de "la quinta tupla" o "la tupla 97" o "la primer tupla" de una relación; en otras palabras, no existe el concepto

de direccionamiento por posición. La referencia mencionada antes en relación con la propiedad de ninguna tupla repetida explica por qué es importante también la propiedad de ningún ordenamiento en las tuplas, de hecho las dos propiedades están relacionada entre sí.

### 3. Los atributos no están ordenados

Esta propiedad se desprende del hecho de que la cabecera de una relación se define también como un conjunto (es decir, un conjunto de atributos, o dicho de otra forma, de pares atributos – dominio). Tampoco puede hablarse de “*el primer atributo*”, o “*el segundo atributo*”, etc. Con ello, se reducen las probabilidades de errores y de programación incomprensible.

### 4. Todos los valores de los atributos son atómicos

Una forma más precisa de expresar esta última propiedad es: “*todos los valores de los atributos simples son atómicos*”. Se trata de una consecuencia del hecho de que todos los dominios subyacentes son a su vez simples; es decir, contienen solo valores atómicos. (Aún si existen atributos compuestos, estos no son sino una simple concatenación de atributos simples). Podemos expresar la propiedad de manera diferente: en cada posición de fila y columna dentro de la tabla, siempre existe un solo valor, nunca una lista de valores. O en otra forma equivalente: *las relaciones no contienen grupos repetitivos*. Si una relación satisface esta condición, se dice que está *normalizada* (que se definirá más adelante).

Lo anterior implica que todas las relaciones están normalizadas, en lo concerniente al modelo relacional, y en realidad así sucede. De hecho, el término “relación”, sin calificativo, se maneja como sinónimo de “relación normalizada” en el contexto del modelo relacional. Pero lo importante es que una relación *matemática* no necesita por fuerza estar normalizada. Un dominio valuado en relaciones es un

dominio cuyos elementos son en sí relaciones, en vez de escalares simples. No deben confundirse los dominios valuados en relaciones con los dominios compuestos. Los elementos de un dominio compuesto son sólo concatenaciones de valores escalares, uno de cada uno de los dominios simples constituyentes; dicho de otro modo, los elementos de un dominio compuesto son tuplas. Los elementos de un dominio valuado en relaciones son conjuntos de tuplas. Además esos conjuntos de tuplas, es decir, las relaciones que son los valores del dominio valuado en relaciones, pueden incluir a su vez atributos valuados en relaciones; esas relaciones a su vez pueden estar no normalizadas.

### Tipos de Relaciones

Algunos de los diferentes tipos de relaciones que pueden existir en un sistema relacional son:

*Relaciones base* (llamadas también relaciones reales): Una relación base corresponde al concepto de tabla base en SQL; es decir, es una relación autónoma, con nombre. Las relaciones base son aquellas cuya importancia (para la aplicación en cuestión) es tal que el diseñador de la base de datos ha decidido darles un nombre y hacerlas parte directa de la base de datos en sí, a diferencia de otras relaciones cuya naturaleza es más efímera, como por ejemplo el resultado de una consulta.

*Vistas* (llamadas también relaciones virtuales): Una vista es una relación derivada, con nombre, representada dentro del sistema exclusivamente mediante su definición en términos de otras relaciones con nombre; no posee datos almacenados propios, separados y distinguibles (a diferencia de las relaciones base).

*Instantáneas*: una instantánea (snapshot) es también una relación derivada, con nombre, como una vista, pero a diferencia de éstas últimas, las instantáneas

son reales, no virtuales; es decir, están representadas no sólo por su definición en términos de otras relaciones con nombre, sino también por sus propios datos almacenados.

La creación de una instantánea es muy similar a la ejecución de una consulta, excepto que a) el resultado de esa consulta se almacena en la base de datos con el nombre especificado, como una relación de sólo lectura, y b) en forma periódica la instantánea se "renueva" (REFRESH); es decir, se deshecha su valor actual, se ejecuta de nuevo la consulta y el resultado de esa nueva ejecución se convierte en el nuevo valor de la instantánea.

*Resultados de Consultas:* Un resultado de consulta es, como su nombre lo indica, tan solo la relación final resultante de alguna consulta especificada. Puede o no tener nombre. Los resultados de la consulta no tienen existencia persistente dentro de la base de datos propiamente dicha.

*Resultados Intermedios:* Un resultado intermedio es una relación (casi siempre sin nombre) resultante de alguna expresión relacional anidada dentro de alguna otra expresión relacional más grande. Por ejemplo, consideremos la siguiente operación en SQL:

```
SELECT DISTINCT S.CIUDAD
FROM S
WHERE S.S# IN
      ( SELECT SP.S#
        FROM SP
        WHERE SP.P# = 'P2' );
```

La relación resultante de la subconsulta en este caso es un resultado intermedio. A semejanza de los resultados finales de las consultas, no tiene existencia persistente dentro de la base de datos.

*Relaciones temporales:* Una relación temporal es una relación con nombre, similar a una relación base o vista instantánea, pero, a diferencia de las tres últimas, se destruye en forma automática en algún momento apropiado. Las relaciones base, vistas e instantáneas, son más permanentes, en cuanto a que solo se destruyen como resultado de alguna acción explícita del usuario.

### **Reglas de Integridad Relacional**

Cualquier base de datos, se compone de alguna configuración de valores de los datos, y se supone que esta configuración "refleja la realidad", es decir, suponemos que es un modelo o representación de algún fragmento del mundo real, tal como el momento real aparece en algún momento dado. Algunas configuraciones de valores no tienen sentido en cuanto a que no representan ningún posible estado del mundo real. Por lo anterior es necesario ampliar la definición de la base de datos a fin de incluir ciertas *reglas de integridad*, cuyo propósito es informar al manejador de la base de datos acerca de ciertas restricciones con el mundo real (como la restricción de que los pesos de un producto X no pueden ser negativos) para que pueda impedir la ocurrencia de tales configuraciones imposibles de valores.

La mayor parte de las bases de datos estarán sujetas a un gran número de tales reglas de integridad. Sin embargo, la mayor parte de estas reglas de integridad son *específicas*, en cuanto a que se aplican a una base de datos específica. El modelo relacional, en cambio, incluye dos reglas generales de integridad: generales en el sentido de que se aplican a no sólo una base de datos específica, sino más bien a todas las bases de datos (o por lo menos a todas las bases de datos que digan apegarse al modelo). Estas dos reglas generales se refieren, respectivamente, a las claves primarias y a las claves ajenas, que se analizarán posteriormente.

Cabe señalar que todas las reglas de integridad, tanto las específicas con respecto a una base de datos (como el ejemplo anterior acerca del requisito de pesos de partes positivos) como las dos reglas generales del modelo relacional, se

aplican a las relaciones base. Esto se debe a que son las relaciones base las que supuestamente reflejan la realidad, y por tanto son éstas las que tienen la restricción de contener sólo configuraciones correctas (o al menos factibles) de valores.

### Claves Primarias

En términos informales, la clave primaria de una relación es sólo un identificador único para esa relación. La clave primaria puede ser compuesta; de hecho es posible, aunque quizá poco usual, tener una relación en la cual el único identificador único (o sea, la clave primaria) sea el atributo compuesto formado por la combinación de *todos* los atributos de la relación; es decir, la relación es "toda la clave".

También es posible, aunque otra vez poco usual, tener una relación con *más de un* identificador único. Como ejemplo, supongamos que cada proveedor tiene siempre un número de proveedor único y un nombre de proveedor único. En un caso así, diríamos que la relación tiene varias claves *candidatas*, escogeríamos entonces una de esas claves candidatas como clave *primaria*, y a las demás les llamaríamos entonces claves *alternativas*.

Tenemos que definir el término clave candidata de la siguiente manera. El atributo  $K$ , probablemente compuesto, de la relación  $R$  es una clave candidata de  $R$  si y solo si satisface las siguientes dos propiedades, independientes del tiempo:

#### 1. Unicidad

En cualquier momento dado, no existen dos tuplas en  $R$  con el mismo valor de  $K$ .

#### 2. Minimalidad

Si  $K$  es compuesto, no será posible eliminar ningún componente de  $K$  sin destruir la propiedad de unicidad.

Toda relación tiene por lo menos una clave candidata, porque las relaciones no contienen tuplas repetidas. En la práctica, las relaciones tienden a tener una y solo una clave candidata, pero sin duda es posible que tengan más.

Supongamos que tenemos una relación llamada ELEMENTOS, la cuál representa la tabla de elementos químicos. Cada elemento tiene un nombre único, un símbolo único (por ejemplo, el símbolo del plomo es Pb), y un número atómico único. Es evidente que la relación tiene tres claves candidatas distintas.

Ahora bien, del conjunto de claves candidatas de una relación dada, se elige una y sólo una como clave primaria de esa relación; las demás, si existen se llamarán claves alternativas. Así, una clave alterna es una clave candidata que no es la clave primaria. En el caso de ELEMENTOS, por ejemplo, podríamos elegir "símbolo del elemento" como clave primaria; "nombre del elemento" y "número atómico" serían entonces claves alternativas.

Surgen los siguientes puntos:

Puesto que toda relación tiene por lo menos una clave candidata, toda relación tendrá por fuerza una clave primaria.

El razonamiento para elegir la clave primaria, en los casos donde hay varias claves candidatas, queda fuera del alcance del modelo relacional en sí. En la práctica la elección suele ser sencilla.

Debe entenderse que, en la práctica, la clave primaria es la que tiene verdadera importancia; las claves candidata y alternativa son sólo conceptos surgidos por fuerza durante el proceso de definir el concepto más importante de clave primaria.

Aunque es verdad que toda relación posee de hecho una clave primaria, en la práctica son las relaciones base a las que se aplica de manera más directa el concepto y para las cuales tiene mayor importancia.

Adviértase que en ningún momento se implica la necesidad de la existencia de un índice según la clave primaria. En la práctica desde luego, si habrá con toda seguridad un índice, pero el que exista o no queda fuera del alcance del modelo relacional en sí.

El manejo de las claves primarias constituye el mecanismo de direccionamiento a nivel de tuplas básico en un sistema relacional. Es decir, el único modo, garantizado por el sistema, de localizar alguna tupla específica es por el valor de su clave primaria. En consecuencia, las claves primarias son tan indispensables para el funcionamiento exitoso de un sistema relacional como las direcciones de memoria principal lo son para el funcionamiento exitoso de la máquina subyacente. De lo anterior se desprende también que un sistema sin conocimiento de claves primarias tenderá a mostrar en ocasiones un comportamiento que no es verdaderamente relacional. De manera similar, las relaciones sin clave primaria mostrarán con seguridad un comportamiento extraño y anómalo en ciertas circunstancias.

Muchas veces se asegura que el modelo relacional "requiere direccionamiento asociativo". Ahora bien, como es evidente que todo direccionamiento relacional, y en particular el direccionamiento por clave primaria, sí es asociativo (es decir, con base en los valores, no en las posiciones), también es evidente que tal aseveración es cierta en el nivel lógico. Pero desde luego no hay razón para asegurar que el modelo relacional requiere equipo asociativo. Como ya se ha dicho muchas veces el modelo relacional se ocupa del nivel lógico del sistema, no del nivel físico. En el nivel físico, los dispositivos de E/S y las estructuras de almacenamiento más convencionales son del todo adecuadas para manejar un sistema relacional con un desempeño bueno y hasta excelente.

## La Regla de Integridad de las Entidades

Llegamos ahora a la primera de las dos reglas generales de integridad del modelo relacional, la regla de integridad de las relaciones. Esta regla dice así.

### **Ningún componente de la clave primaria de una relación base puede aceptar nulos**

Con "nulos" queremos decir aquí información faltante por alguna razón, por ejemplo, si la propiedad no es aplicable o si el valor se desconoce. Entenderemos por nulo sencillamente un valor o representación que, por convención, no representa valor real alguno del atributo aplicable. La justificación de la regla de las entidades es la siguiente:

- Las relaciones base corresponden a entidades en el mundo real.
- Por definición, las entidades en el mundo real son distinguibles, es decir, se les puede identificar de alguna manera.
- Por tanto, los representantes de entidades dentro de la base de datos deben ser distinguibles (identificables) también.
- Las claves primarias realizan esta función de identificación única en el modelo relacional (es decir, sirven para representar a los identificadores de entidades requeridos).
- Si un "nulo" significa "la propiedad no es aplicable", es evidente que esa tupla no tiene sentido; como ya se explicó, las entidades deben tener identidades, y la propiedad deberá ser aplicable.
- Si significa el valor se desconoce, surge todo tipo de problemas. Por ejemplo, si queremos saber si la tupla representa a un proveedor que sí conocemos (porque no lo sabemos; esto es el significado de "se desconoce"). Entonces deberemos responder a la consulta "¿Cuántos proveedores hay?", con la respuesta "no se sabe".

- En conclusión, si una entidad es lo bastante importante en el mundo real como para requerir una representación explícita en la base de datos, tal entidad deberá ser definitivamente susceptible de identificación definida y sin ambigüedad, pues de lo contrario sería imposible siquiera hablar de ella en cualquier forma sensata: por esta razón, la regla de integridad de las entidades se expresa así:

**En una base de datos relacional, nunca registraremos información acerca de algo que no podemos identificar.**

Surgen los siguientes puntos:

1. Hacemos hincapié en que, en el caso de las claves primarias compuestas, la regla de integridad de entidades dice que cada valor individual de la clave primaria debe ser no nulo en su totalidad (no sólo en parte).
2. La regla de identidad de las entidades se aplica a las relaciones base. Otras relaciones bien podrían tener una clave primaria en la cual se permitieran nulos.
3. La regla de identidad de las entidades se aplica solo a las claves primarias. En las claves alternativas se podrán o no permitir nulos.

### **Claves Ajenas**

Una clave ajena es un atributo (quizá compuesto) de una relación R2 cuyos valores deben concordar con los de la clave primaria de alguna relación R1 (donde R1 y R2 no necesariamente son distintos).

Un valor de clave ajena representa una referencia a la tupla donde se encuentra el valor correspondiente de la clave primaria. Por tanto el problema de garantizar que

la base de datos no incluya valores no válidos de una clave ajena se conoce como el problema de la integridad referencial. La restricción según la cual los valores de una clave ajena determinada deben concordar con los valores de la clave primaria correspondiente se conoce como restricción referencial. La relación que contiene la clave ajena se conoce como relación referencial y la relación que contiene a la clave primaria correspondiente se denomina relación referida o relación objetivo.

He aquí una definición más formal del término clave ajena:

El atributo LF (quizá compuesto) de la relación base R2 es una clave ajena si y solo si satisface estas dos propiedades independientes del tiempo:

1. Cada valor de LF es nulo del todo o bien no nulo del todo (con "nulo del todo" o "no nulo del todo" queremos decir que si LF es compuesto todos sus componentes son nulos o bien todos sus componentes son no nulos, no una combinación).
2. Existe una relación base R1 con clave primaria LP tal que cada valor no nulo de LF es idéntico al valor de LP en alguna tupla de R1.

De donde surgen los siguientes puntos:

1. Una clave ajena dada y la clave primaria correspondiente deben definirse sobre el mismo dominio subyacente (el cual puede ser compuesto).
2. La clave ajena no necesita ser un componente de la clave primaria de la relación que la contiene. Veamos un ejemplo, con las relaciones departamentos y empleados:

DEPTO ( NUMDEP, ..., PRESUP, ...)

EMP ( NUMEMP, ..., NUMDEP, ..., SALARIOS, ...)

En estas base de datos, el atributo EMP.NUMDEP es una clave ajena en la relación EMP (y concuerda con la clave primaria DEPTO.NUMDEP de la relación DEPTO); sin embargo, no es componente de la clave primaria EMP.NUMEMP de esa relación EMP. En general, cualquier atributo (en una relación base) puede ser una clave ajena.

3. Una relación dada puede ser desde luego tanto una relación referida como una relación referencial.
4. Las relaciones R1 y R2 en la definición de la clave ajena *no son necesariamente distintas*. Es decir, una relación podría incluir una clave ajena cuyos valores (no nulos) deben concordar con los valores de la clave primaria de esa misma relación. Como ejemplo consideremos la relación:

EMP ( NUMEMP, ..., SALARIO, ..., NUMEMP\_GER, ... )

En la cual el atributo NUMEMP\_GER representa el número de empleado del gerente del empleado identificado mediante NUMEMP. Aquí, NUMEMP es la clave primaria, y NUMEMP\_GER es una clave ajena que hace referencia a ella. Las relaciones como EMP en este ejemplo se denominan en ocasiones *relaciones autoreferenciales*.

5. Las claves ajenas, a diferencia de las claves primarias (en relaciones base) deben aceptar nulos en ocasiones. Como ejemplo, consideremos la relación EMP recién analizada, ¿cuál es el valor de NUMEMP\_GER para el presidente de la compañía?

### **La Regla de Integridad Referencial**

La regla de Integridad Referencial es la segunda regla general del modelo relacional, y es:

#### **La base de datos no debe contener valores de clave ajena sin concordancia**

Con el término "valores de clave ajena sin concordancia" queremos decir aquí un valor no nulo de clave ajena para el cual no existe un valor concordante de la clave primaria en la relación objetivo pertinente.

Así como los valores de la clave primaria representan identificadores de entidades, así los valores de la clave ajena representan referencias a entidades (a menos que sean nulos). La regla de integridad referencial dice tan solo que si B hace referencia a A, entonces A debe existir.

Surgen los siguientes puntos:

La integridad referencial exige concordancia de las claves ajenas muy específicamente, con claves primarias, no con claves alternativas.

Los conceptos de clave ajena e integridad referencial se definen uno en términos del otro. No es posible explicar qué es una clave ajena sin mencionar la noción de integridad referencial, y de la misma manera es imposible explicar qué es la integridad referencial sin mencionar la noción de clave ajena.

## **2.4. Modelo Cliente/Servidor**

### **2.4.1. Introducción**

Cuando los profesionistas encargados de computadoras se refieren a cliente/servidor, tienden a pensar en una nueva tecnología que puede solucionar muchos problemas en el manejo de datos que enfrentan las organizaciones modernas. Los encargados se centran en la noción de que el cómputo cliente/servidor puede ser usado para crear nuevas oportunidades de negocio en un mundo manejado por la información. Los usuarios finales tienden a pensar acerca del software que está basado en la Interfaz de Usuario Gráfica (GUI) y que además tiene la capacidad para acceder diversos recursos en casi cualquier plataforma. En otras palabras, podemos descubrir que la descripción de este modelo depende de la persona a la que le preguntemos y del trabajo que desempeñe.

### **2.4.2. Conceptos**

#### **Definición:**

Cliente/Servidor es un término utilizado para describir un modelo de cómputo para el diseño de sistemas computarizados. Este modelo está basado en la distribución de funciones a través de dos tipos de procesos independientes y autónomos: servidores y clientes. Un cliente es cualquier proceso que solicita servicios específicos a los procesos del servidor. Un servidor es un proceso que provee servicios solicitados por los clientes. Tanto clientes como servidores pueden residir en la misma o en diferentes computadoras, conectadas por una red.

Un cliente puede solicitar servicios a varios servidores en la red sin necesidad de conocer la localización o las características de las computadoras en que residen los procesos servidores. La red sirve tanto a clientes como a servidores, proveyendo el medio a través del cual los clientes y los servidores se comunican.

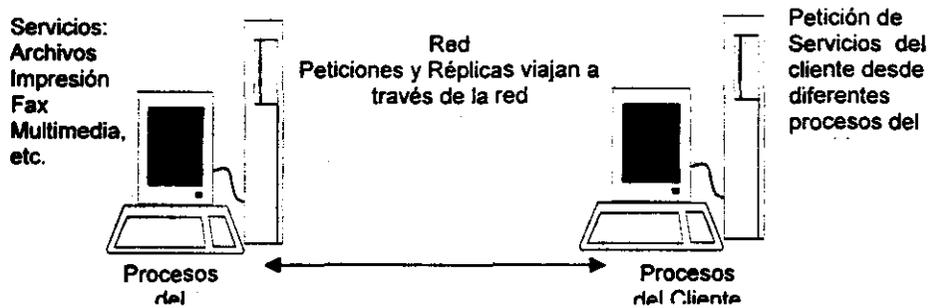


Figura 2.4.2.1 Modelo Cliente/Servidor Básico

La clave en el modelo cliente/servidor es el lugar en que se lleva a cabo el procesamiento de la solicitud. Por ejemplo, en una base de datos cliente/servidor el proceso cliente solicita datos de la base de datos servidora. El proceso actual de la solicitud (selección de registros) toma lugar en la máquina servidora de la base de datos. En otras palabras, el proceso del servidor selecciona los registros que concuerdan con los criterios de selección y los envía a través de la red hacia el proceso cliente. El procesamiento de la información puede ser dividida entre diferentes tipos de computadoras (servidores) tales como PC basadas en Pentium, PowerPC, una estación de trabajo RISC (Reduce Instruction Set Computing), una minicomputadora o un mainframe.

La separación de las tareas de procesamiento de datos es la clave que da la diferencia entre los sistemas cliente/servidor y los sistemas de mainframe. En los sistemas de mainframe, todo el procesamiento se lleva a cabo en el mainframe, y en el otro lado, la terminal (usualmente tonta) es solamente utilizada para desplegar los datos en pantalla. En este ambiente, no existe autonomía: La terminal tonta es simplemente una extensión del mainframe. En contraste, el ambiente cliente/servidor provee una clara separación de los procesos del cliente y los del servidor, y ambos procesos son autónomos. La relación entre servidores y clientes es de muchos a

muchos, donde un servidor puede proveer servicios a muchos clientes y un cliente puede solicitar servicios de muchos servidores.

### **Evolución de los Sistemas de Información basados en el Modelo Cliente/Servidor**

En la mitad de la década de 1970, los datos de una corporación residían seguramente en un grande y caro mainframe, que eran controlados por sistemas operativos complejos y propietarios. Las terminales tontas, conectadas como front-ends del procesador, se comunicaban con el mainframe para producir la información deseada. El mainframe y sus dispositivos eran celosamente resguardados, y el acceso estaba rigurosamente restringido al personal autorizado. Este estilo de cómputo, controlado por una parte por el hardware y el software disponible y por otra parte hecho posible por un ambiente de datos relativamente estático. Este estilo centralizado de cómputo imponía un control rígido de las aplicaciones, límites estrictos a los usuarios finales y un departamento complejo encargado de su mantenimiento.

Con la introducción de las microcomputadoras a principios de la década de los 80's, los usuarios fueron capaces de manipular datos localmente con la ayuda de software relativamente fácil-de-usar tales como hojas de cálculo y sistemas de bases de datos basados en microcomputadoras. Sin embargo, los datos en que el software trabajaba aún residían en el mainframe. Los usuarios a menudo reintroducían los datos necesarios manualmente para hacerlos accesibles a las aplicaciones locales. Este trabajo extra manual no era muy productivo y presentaba muchas anomalías en los datos.

El uso de las PC's creció grandemente a través de los años y eventualmente reemplazó a las terminales tontas en los escritorios de los usuarios finales. Los programas de comunicación y de emulación de terminales permitían a la PC's conectarse e integrarse al centro de datos. Las PC's conectadas al mainframe eran

llamadas "terminales inteligentes". Ya en estos años la adquisición de datos desde el mainframe hacia las PC's era el camino estándar de extraer los datos requeridos desde el mainframe para ser manipulados por las PC's locales. Dado este ambiente de acceso de datos, los datos del usuario final son actualmente un reflejo de los datos cambiantes de una compañía. Así mismo, los datos del mainframe deben ser frecuentemente adquiridos para evitar reportes viejos o resultados de búsquedas inadecuadas.

Usando sus PC's los usuarios finales fueron hábiles para crear sus propias bases de datos y reportes localmente, disminuyendo con ello el control centralizado de datos y servicios. Desafortunadamente esta nueva libertad del usuario final causa la proliferación de diferentes versiones de las bases de datos de las compañías. Este escenario creó "islas" de información que eran independientes de la base de datos del mainframe. La compartición de datos entre estas "islas" era una tarea poco sofisticada: Cuando los usuarios necesitaban compartir datos, ellos copiaban simplemente los datos a un diskette e iban hacia la oficina de sus compañeros, diskette en mano. Esta aproximación hacia la compartición de información fue etiquetada posteriormente como *sneaker net*.

La introducción de las PC's provocó problemas de seguridad de datos, replicación e integridad para los departamentos de las compañías. Sin embargo, desde la introducción de las PC's también trajeron muchos beneficios a los usuarios finales, pero su crecimiento no podía ser controlado fácilmente.

Consecuentemente, para retener algún balance en el control, los encargados de sistemas se encargaron de diseñar grupos de usuarios de PC's para compartir la información electrónicamente.

La nueva forma de compartir información electrónicamente fue hecha posible en gran parte por una compañía pequeña conocida como Novell Data Systems, la cuál presentó Netware/86 (originalmente llamado ShareNet) en 1983. El software de Novell y el hardware permitieron a los administradores controlar las PC's conectadas

a través de una LAN. La LAN permitió a los usuarios finales en las PC's compartir archivos vía una PC central que actuaba como un servidor de archivos de red. Netware/86, más tarde actualizado a Netware 2.15, 3.1x y 4.x, se convirtió en el primer Sistema Operativo de Red (NOS) ampliamente aceptado para computadoras IBM personales y compatibles.

Al avanzar la tecnología de los microprocesadores de las PC's, nuevas empezaron a rivalizar con el poder de procesamiento de los mainframe. El nuevo poder de las PC's lo hicieron posible y, basado en el costo relativo de cómputo, lo hicieron aún deseable para los administradores de sistemas para mover algunas de sus aplicaciones a sistemas basados en PC's, que utilizaban interfaces para comunicarse con el mainframe. Un resultado exitoso fue el cambio operacional en donde ahora tanto el usuario final como el administrador de sistemas trabajaban en una plataforma común.

El gran apogeo de las PC's en la década de los 90's está basado típicamente en una red que mantiene grandes cantidades de computadoras heterogéneas juntas. La PC se ha convertido en una estación común para el usuario final en las redes de las corporaciones, y generalmente se ha convertido en el punto de acceso de la base de datos de toda la compañía. Las aplicaciones permiten a los usuarios finales acceder directamente los datos sin importar la localización de los mismos, el modelo de datos utilizado o las características de comunicación de las otras computadoras en la red. La red permite a cada usuario compartir información con otros, para acceder los recursos del mainframe y para acceder alguna fuente de datos externos.

Hoy en día, los usuarios finales requieren y demandan acceso para el análisis y la toma de decisiones, y, quizás lo más importante, ellos incrementan la productividad rediseñando los procesos básicos del negocio dentro de sus propios departamentos. La cada vez más creciente y demandante solicitud para compartir recursos ha creado el ambiente para el cómputo basado en el modelo cliente/servidor.

La evolución desde el cómputo en mainframe hasta el cómputo en PC's del modelo cliente/servidor han generado muchos cambios en algunos de los aspectos clave del manejo de información. Algunos de estos cambios se presentan en la siguiente tabla.

**Contraste entre los sistemas de Información Cliente/Servidor y de Mainframe**

	Mainframe	Cliente/Servidor
Manejo	Centralizado	Distribuido/Descentralizado
Vendedor	Solución de vendedor único	Solución de vendedores múltiples
Hardware y Software	Propietario	Múltiples vendedores
Seguridad	Alta/Centralizada	Normal/Descentralizada
Capacidad de Manipular Datos	Muy limitada	Muy flexible
Manejo del Sistema	Integrada	Algunas utilerías disponibles
Diseño de Aplicaciones	Sobreestructurado Consume tiempo	Flexible Diseño rápido de aplicaciones
Plataforma de usuario final	Terminales tontas Basado en caracteres Tareas sencillas	PC's inteligentes Interfaz de usuario gráfica Multitareas

**Figura 2.4.2.2 Evolución del modelo Cliente/Servidor**

### 2.4.3. VENTAJAS

El cómputo cliente/servidor se ha incrementado debido a su potencial, capaz de hacer buen uso de los recursos de hardware y software, en una palabra, la utilización eficiente de recursos es la clave. Tal eficiencia, es una función de las siguientes características:

- Cliente/Servidor nos permite enfocarnos en sistemas que son independientes de cualquier plataforma de hardware o de software.

- Cliente/Servidor nos permite optimizar la distribución de las actividades entre las diferentes plataformas de cómputo utilizando las mejores características de cada una de ellas.
- Cliente/Servidor nos permite usar una combinación de técnicas, metodologías y herramientas especializadas para diseñar sistemas que sean amigables, de buen costo y que se puedan comunicar a través de cualquier hardware y software.

Una vez mencionados los beneficios potenciales del modelo cliente/servidor, es apropiado tener en mente las siguientes observaciones:

- Aunque el modelo cliente/servidor soluciona potencialmente los problemas del manejo de información, no es su principal fin. En su lugar, esto es una herramienta que puede ser utilizada para alcanzar el propósito principal que es el manejo de información eficiente.
- El modelo cliente/servidor no es la respuesta para todos los problemas de manejo de datos, por ejemplo: ningún diseño pobre de una base de datos o aplicaciones mal escritas son capaces de correr en el modelo cliente/servidor.
- El modelo cliente/servidor aún está en una etapa intermedia, está basado en una tecnología compleja. La combinación de estas dos características genera sus propios problemas de manejo. A menos que la tecnología sea bien entendida y manejada, las expectativas en su manejo no podrán tener los beneficios concretos. De hecho si no es cuidadosamente cuidada e implementada, algunos modelos podrían tener resultados adversos.

Dada la abundancia de sistemas cliente/servidor que se encuentran en artículos de revistas, conferencias, etc., no es sorprendente que las expectativas en su manejo o beneficios sean grandes. Sin embargo, es útil recordar que los beneficios esperados dependen de la naturaleza del negocio y del tipo de implementación.

#### 2.4.4. Arquitectura Cliente/ Servidor

Una infraestructura conocida como arquitectura cliente/servidor, es un requisito para el diseño adecuado de este tipo de sistemas. Está basada en componentes de hardware y software que interactúan para formar un sistema. El sistema comprende tres componentes principales: clientes, servidores y medio de comunicación.

- El cliente es cualquier proceso que solicita servicios del servidor. El cliente, también conocido como front-end (interfaz de usuario) de la aplicación, refleja el hecho de que el usuario final generalmente interactúa con el proceso cliente.
- El servidor es cualquier proceso que provee servicios a los clientes. El servidor también conocido como back-end (fondo) de la aplicación, refleja el hecho de que el proceso de la aplicación provee los servicios de fondo para el proceso del cliente.
- El medio de comunicación es cualquier proceso a través del cual, los clientes y los servidores se comunican. El medio de comunicación, también conocido como medio o capa de comunicación, esta compuesto de varias capas de software que se añaden en la transmisión de datos y en la información de control entre clientes y servidores. El medio se asocia generalmente con una red. Todas las solicitudes del cliente y respuestas del servidor viajan a través de la red en forma de mensajes que llevan información de control y datos.

#### Forma de Interactuar de los Componentes

Para ilustrar cómo interactúan los componentes, examinaremos como un proceso de cliente hace peticiones desde un proceso de base de datos. Tal como se ilustra en la figura 2.4.4.1, el proceso de la aplicación ha sido dividida en dos procesos principales e independientes: un cliente y un servidor. El medio de comunicación hace posible que los procesos del cliente y del servidor trabajen juntos. También se observa que el medio de comunicación es la plataforma en la cual los clientes y los servidores se encuentran. El medio de comunicación es un componente clave en el

sistema, su presencia añade valor al crear over-head adicional, además de puntos contra fallas del sistema y, en general, suma complejidad a la implementación del sistema.

Como se mencionó al principio, el servidor y el cliente no necesitan estar en diferentes computadoras, a menudo pueden residir en la misma computadora para compartir el mismo procesador, asumiendo que el sistema operativo lo permita, esto es asumiendo el uso de un sistema operativo multitarea. Sin embargo, la mayoría de las implementaciones cliente/servidor colocan a los procesos en diferentes computadoras, la figura 1.5.4 ilustra un sistema cliente/servidor con dos servidores y tres clientes.

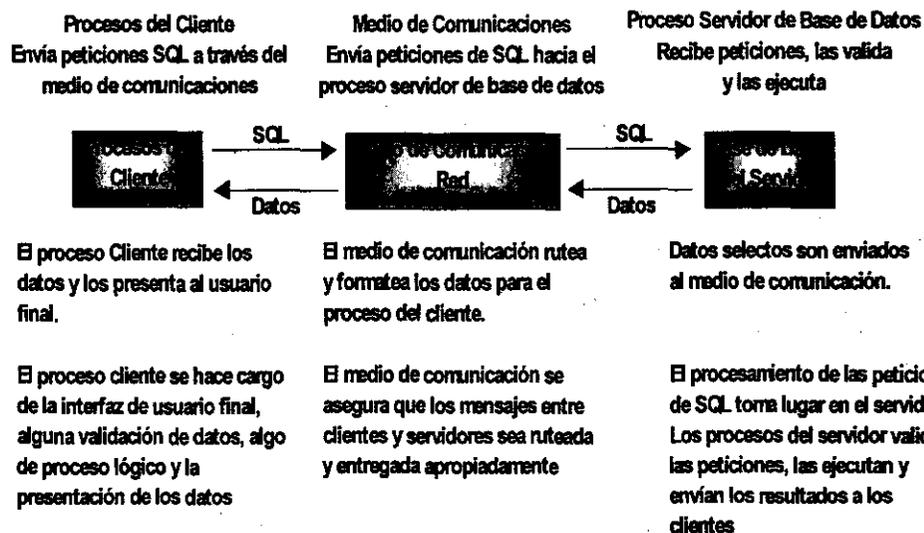


Figura 2.4.4.1 Interacción de los componentes del Cliente y del Servidor

Dado el ambiente mostrado en la figura 2.4.4.2 se tiene un proceso servidor de base de datos que utiliza una computadora basada en RISC, y un proceso servidor imaginario que corre en una computadora tipo PS/2. Los tres procesos cliente pueden correr bajo tres diferentes sistemas operativos: DOS/Windows, OS/2 y Apple Mac System 7. Los procesos del cliente y del servidor están conectados a través de una red token-ring. Las aplicaciones de front-end en la computadora del cliente solicitan datos e imágenes de los procesos de back-end del servidor (servidores de base de datos y de imágenes). La red y el software que la soporta forman el medio de comunicación a través del cual el cliente y el servidor se comunican. La comunicación puede darse tanto entre clientes y servidores como entre servidores.

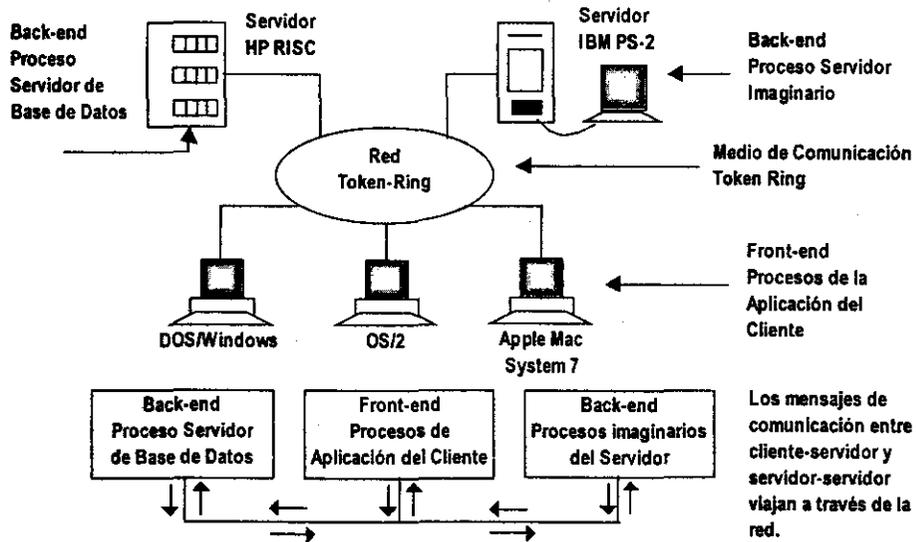


Figura 2.4.4.2 Ejemplo de la Arquitectura Cliente-Servidor

La figura anterior ilustra un ambiente cliente/servidor complejo, en la cual los procesos del servidor corren en dos diferentes sistemas operativos, los procesos del cliente corren en tres diferentes sistemas operativos y el sistema contiene tres

plataformas de hardware diferentes. En este escenario, el medio de comunicación se convierte en la plataforma integrada para todos los componentes.

### **Principios del Modelo Cliente/Servidor**

Los componentes de la arquitectura cliente/servidor deben cumplir con algunos principios básicos si se desea que actúen adecuadamente. Esos principios deben ser uniformemente aplicables tanto a clientes, a servidores y al medio de comunicación. Algunos de los principios se describen a continuación:

- Independencia de hardware
- Independencia del software
  - Sistema Operativo
  - Sistema de red
  - Aplicaciones
- Acceso abierto a los servicios
- Distribución de procesos
  - Autonomía de procesos
  - Maximización de recursos locales
  - Escalabilidad y flexibilidad
  - Interoperabilidad e integración
- Estándares

#### **Independencia de hardware:**

El principio de la independencia de hardware requiere que los procesos del cliente, los del servidor y los del medio de comunicación sean capaces de correr en múltiples plataformas (IBM, DEC, APPLE, etc.) sin diferencias funcionales.

**Independencia del software:**

El principio de independencia del software requiere que los procesos del cliente, los del servidor y los del medio de comunicación sean capaces de soportar múltiples sistemas operativos (tales como DOS, OS/2, UNIX, etc.), múltiples protocolos (tales como IPX ó TCP/IP) y múltiples aplicaciones (hojas de cálculo, bases de datos, correo electrónico, etc.).

**Acceso abierto a servicios:**

Todos los clientes en el sistema deben tener acceso abierto (sin restricciones) a todos los servicios que la red provee, y esos servicios no deben ser dependientes de la localización del cliente o del servidor. Un punto clave es que los servicios deben ser provistos de acuerdo a las demandas del cliente. De hecho, la provisión de servicios es uno de los objetivos principales del modelo cliente/servidor.

**Distribución de procesos:**

Una de las primeras características identificables de los sistemas cliente/servidor es que el procesamiento de la información se hace de forma distribuida entre los clientes y los servidores. La división del procesamiento de la información debe realizarse de acuerdo con las siguientes reglas:

Los procesos del cliente y del servidor deben ser entidades autónomas con límites y funciones claramente definidas. Esta propiedad nos permite definir claramente la funcionalidad de cada uno de los procesos, lo cual permite la modularidad y flexibilidad de los sistemas.

La utilización local de recursos (tanto por parte del cliente como del servidor) se maximiza. Los procesos del cliente y del servidor deben hacer uso completo del poder de cómputo de las computadoras. Esta propiedad permite al sistema el

asignar funcionalmente los mejores recursos a las tareas apropiadas. En otras palabras, para lograr la mejor utilización de recursos el proceso del servidor debe estar compartido entre todos los procesos del cliente, esto es, un proceso del servidor debe atender múltiples peticiones de múltiples clientes.

La escalabilidad y la flexibilidad requieren que tanto los procesos del cliente como del servidor sean fácilmente actualizables para correr en plataformas de software o de hardware más poderosas. Esta propiedad extiende la funcionalidad de los procesos cliente/servidor cuando son ejecutados para proveer de capacidades adicionales o para obtener un mejor desempeño.

La interoperabilidad y la integración requieren que tanto los procesos del cliente como del servidor sean integrados para formar en conjunto un sistema. El cambio entre procesos del servidor debe ser transparente para los procesos del cliente.

#### **Estándares:**

Finalmente, todos los principios deben estar basados en estándares aplicados en la arquitectura cliente/servidor. Por ejemplo, los estándares deben gobernar las interfaces de usuario, el acceso a los datos, protocolos de red, comunicación de interprocesos, etc. Los estándares aseguran que todos los componentes sean capaces de interactuar de forma ordenada para obtener los resultados esperados.

Estándares universales no existen para todos los componentes aún. El hecho es que existen diferentes estándares de dónde escoger. Por ejemplo, una aplicación puede estar basada en un ODBC (Open Database Connectivity) en lugar de IDAPI (Integrated Database Application Programming Interface) para acceder a los datos, y puede utilizar como protocolo de red a IPX (Internet Packet Exchange) en lugar de TCP/IP. Si la aplicación no utiliza un estándar sencillo no significa que no sea una aplicación del tipo cliente/servidor. La idea es asegurar que si una aplicación soporta

un estándar, todos sus componentes (servidor, clientes y medio de comunicación) sean capaces de interactuar al utilizar el mismo estándar.

#### 2.4.5. El Cliente

El cliente es cualquier proceso que solicita servicios de un proceso del servidor. El cliente es proactivo y puede siempre iniciar la conversación con el servidor. El cliente está compuesto por componentes de hardware y de software. Las características deseables de un cliente son las siguientes:

- Hardware poderoso
- Un sistema operativo que permita la multitarea
- Una interfaz de usuario gráfica (GUI)
- Capacidades de comunicación

Dado que los procesos del cliente tienden a requerir muchos recursos de hardware, deben de existir en computadoras con suficiente poder de cómputo. Los avances en la tecnología del microprocesador hacen posible el equipar PC's con CPU's poderosos (como Pentium, diseñado por Intel y PowerPC, conjuntamente diseñado entre IBM, APPLE y MOTOROLA). Tales facilidades en el procesamiento de información facilitan la creación de sistemas con capacidades de multimedia. Los sistemas multimedia manejan diferentes tipos de datos, tales como voz, imágenes, video, etc. Los procesos del cliente también requieren gran cantidad de espacio de disco duro y de memoria física.

El cliente debe tener acceso a un sistema operativo con al menos algunas características de multitarea. La combinación de DOS y Windows es actualmente la plataforma más común para clientes. Mientras que DOS tiene varias limitaciones en cuanto a memoria y soporte de múltiples aplicaciones, Windows provee acceso a memoria, capacidades de multitarea y una interfaz de usuario gráfica. Estas capacidades sumadas a la abundancia de aplicaciones diseñadas para la interfaz de

Windows, permiten el escoger la combinación de DOS/Windows como la mejor implementación del modelo cliente/servidor. Sin embargo existen muchas otras plataformas igual de populares, como Windows NT, OS/2 y las diferentes versiones de UNIX.

Para tener un ambiente cliente/servidor que interactúe eficientemente, la computadora cliente debe ser capaz de conectarse y comunicarse con otras computadoras en un ambiente de red. Sin embargo, la combinación de hardware y sistemas operativos deben proveer también de la conectividad adecuada para múltiples sistemas operativos de red (NOS). La razón para requerir que una computadora cliente sea capaz de conectar y acceder múltiples sistemas operativos es simple: algunos servicios pueden estar en diferentes redes.

La aplicación del cliente o el front-end corre en la parte "alta" del sistema operativo y se conecta con el medio de comunicación para acceder los servicios disponibles en la red. Varios lenguajes de programación de tercera generación pueden ser utilizados para crear aplicaciones de front-end. La figura 2.4.5.1 describe los componentes del cliente básicos.

En esta figura, se nota que la aplicación de front-end interactúa con el sistema operativo para acceder las multitareas y las capacidades GUI que el sistema operativo tiene. Las aplicaciones de front-end también interactúan con los componentes del software de la red para acceder los servicios localizados en la misma. Los componentes de hardware del medio de comunicación, transportan físicamente las solicitudes y las respuestas entre los clientes y los servidores. Mientras que la solicitud es procesada por el servidor, el cliente está libre para realizar otras tareas.

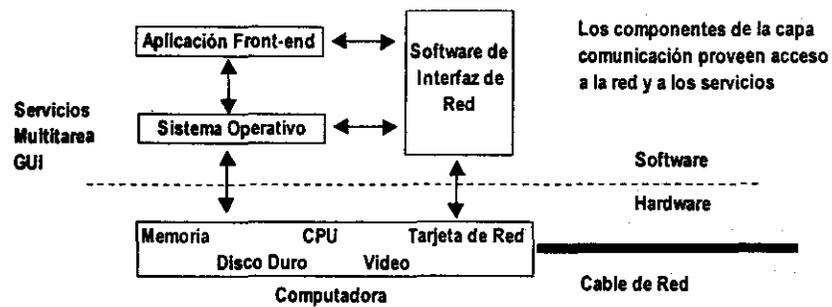


Figura 2.4.5.1 Componentes del Cliente

#### 2.4.6. El Servidor

El servidor es cualquier proceso que provee servicios a los procesos del cliente. El servidor es *reactivo* porque siempre se encuentra en espera de las solicitudes del cliente. Los servidores proveen comúnmente de:

*Servicios de Archivos* para el ambiente de LAN, en el cuál una computadora con un disco duro grande y rápido que es compartido entre varios usuarios. Un cliente conectado a la red puede almacenar archivos en el servidor de archivos como si se tratara de otro disco duro local.

*Servicios de Impresión* para un ambiente tipo LAN, en el cual una PC con una o más impresoras conectadas es compartida entre varios clientes. Un cliente puede acceder cualquiera de las impresoras como si se encontraran directamente conectadas a su computadora. Los datos a imprimir viajan desde la PC del cliente hasta la PC servidora de impresión, en dónde son almacenados temporalmente en el disco duro. Cuando el cliente termina su impresión, los datos son llevados del disco duro del servidor de impresión hacia la impresora apropiada.

*Servicios de Fax*, requieren al menos un servidor equipado (interna o externamente) con un dispositivo de fax. La computadora cliente no necesita tener un fax ni una línea telefónica. En vez de ello, el cliente envía los datos a ser enviados por fax hacia el servidor de fax con la información necesaria, como número de fax o nombre del receptor. El servidor de fax puede programar el fax, marcar el número de fax y transmitirlo.

*Servicios de comunicación* que permitan a la PC cliente acceder otras computadoras o recursos de la red, a los cuales no se encuentra directamente conectado.

*Servicios de base de datos*, que constituye la implementación cliente/servidor más común y de mayor uso. Con la existencia de un servidor de base de datos, el cliente puede enviar una solicitud SQL hacia el servidor, el cuál recibe tal petición, la valida, la ejecuta y envía únicamente el resultado al cliente. Los datos y la base de datos se localizan en la máquina servidora de base de datos. El cliente solo necesita tener la aplicación front-end para acceder el servidor de base de datos.

*Servicios de transacciones* son provistos por servidores de transacción que son una variante de los servidores de base de datos. Un servidor de transacción contiene una base de datos, su estructura y los procedimientos necesarios para manejar los datos. Una aplicación front-end en la máquina cliente envía una petición al servidor de base de datos para ejecutar un procedimiento específico. Ningún código de SQL viaja a través de la red. Los servidores de transacción reducen el tráfico de la red y proveen de un mejor desempeño que las bases de datos servidoras.

*Miscelánea de Servicios*, que incluyen CD-ROM, video, respaldo, etc.

Una idea equivocada es que el proceso del servidor debe ejecutarse en la computadora que contiene el sistema operativo de red. Actualmente, a menos que las circunstancias dicten otra cosa, la separación del proceso del servidor y del

sistema operativo de red se recomienda ampliamente. Tal separación permite al proceso del servidor ser colocado en cualquiera de las computadoras de la red y estar disponible para todos los usuarios de la misma. Por ejemplo, supongamos que tenemos un servidor de CD-ROM en una red de Novell. Si el software del servidor requiere que los procesos del servidor de CD-ROM se ejecuten en la misma computadora en donde reside el sistema operativo, provoca mayor exigencia a la computadora host, la cuál tendrá entonces una doble tarea, como servidor de archivos y como servidor de CD-ROM.

Como en el caso del cliente, el servidor también tiene componentes de hardware y de software. Los componentes de hardware incluyen la computadora, el CPU, memoria, disco duro, video, tarjeta de red, etc. La computadora que mantiene el proceso servidor debe ser una computadora más poderosa que la del cliente, porque el proceso del servidor debe ser capaz de procesar solicitudes concurrentes de los múltiples clientes. Algunos otros componentes se muestran en la figura 2.4.6.1.

Las aplicaciones del servidor o back-end se ejecutan en la parte "alta" del sistema operativo e interactúa con los componentes del medio de comunicación para "escuchar" las solicitudes de los clientes.

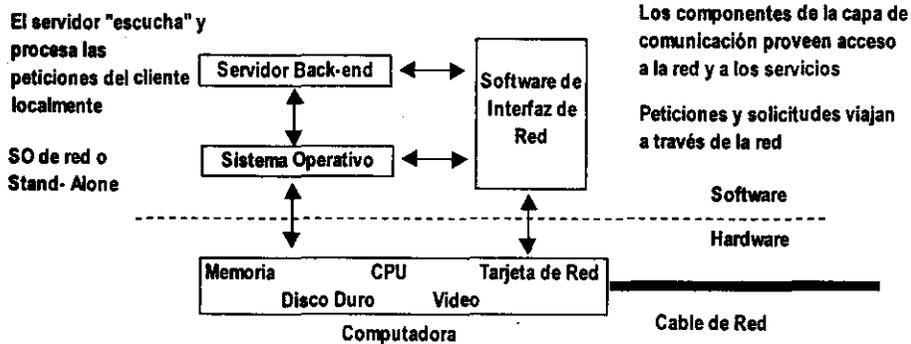


Figura 2.4.6.1 Componentes del Servidor

Contrario a los procesos del cliente, los procesos del servidor no necesitan de una GUI. Hay que tener claro que las aplicaciones del servidor interactúan con el sistema operativo para acceder los recursos locales. El servidor constantemente "escucha" las solicitudes de los clientes. Una vez que recibe una solicitud, la procesa localmente. El servidor sabe cómo procesar las solicitudes, el cliente solicita solamente al servidor lo que necesita, y no cómo hacerlo. Cuando la solicitud ha sido procesada, la respuesta es enviada de regreso hacia el cliente a través del medio de comunicación.

Las características de hardware especiales dependen de la magnitud de las solicitudes de servicios. Por ejemplo, un servidor de base de datos que fuera a ser usado por 50 clientes, podría requerir como mínimo un equipo con las siguientes características:

- CPU veloz
  - RISC, Pentium, PowerPC o multiprocesador
- Capacidad de tolerancia a fallas
  - Doble alimentación de energía, para evitar problemas eléctricos
  - Fuente de poder regulada para proteger contra fallas en la línea
  - Memoria de verificación y corrección de errores (ECC) para proveer protección contra fallas en el disco duro
- Expansión de CPU, memoria, disco y periféricos
- Bus capaz de soportar múltiples tarjetas
- Múltiples opciones de comunicación

En teoría, cualquier proceso de cómputo que pueda ser claramente dividido en componentes de cliente y de servidor puede ser implementado a través del modelo cliente/servidor. Si se implementa adecuadamente, los procesos de la arquitectura cliente/servidor se reflejan en los siguientes beneficios:

- *Independencia de la Localización:* Los procesos del servidor pueden encontrarse en cualquier parte de la red.
- *Optimización de los recursos:* Los procesos del servidor pueden ser compartidos por varios procesos cliente.
- *Escalabilidad:* Los procesos del servidor pueden ser actualizados para ejecutarse en plataformas más poderosas.
- *Interoperabilidad e Integración:* Los procesos del servidor pueden ser capaces de trabajar en ambientes "plug and play" (conectar y usar).

Estos beneficios, añadidos a los principios de independencia de hardware y software del modelo cliente/servidor, facilitan la integración de las PC's, de las microcomputadoras y de los mainframes en un mismo ambiente de trabajo.

#### **El medio de comunicación**

El software del medio de comunicación provee el medio a través del cual los clientes y los servidores se comunican para realizar tareas específicas. Este software también provee de los servicios especializados para los procesos del cliente que permiten programar las aplicaciones de front-end desde puntos internos del servidor de base de datos y de los protocolos de red. En el pasado, los programadores de aplicaciones debían escribir código que sirviera de interfaz con el lenguaje específico de cada base de datos y con el protocolo de red que utiliza, si era necesario intentar comunicación con otras bases de datos, este código debía escribirse nuevamente, lo cuál era completamente indeseable. La definición de medio de comunicación se basa en sus funciones y en sus propósitos:

El medio de comunicación describe el software especial que:

- a) se encuentra lógicamente colocado entre el cliente y el servidor, y
- b) provee de servicios especializados para garantizar que el cliente esté enterado de todos los detalles tanto del protocolo de red como de los procesos del servidor.

Así mismo el software del medio de comunicación puede ser utilizado en diferentes escenarios, tales como e-mail, fax, etc., la mayoría del software de primera generación utilizado en las aplicaciones cliente/servidor está orientado a proveer acceso a datos transparente para varios servidores de base de datos. El uso de este software permite:

Independencia de la red, permitiendo a las aplicaciones de front-end acceder datos sin preocuparse de los protocolos de red

Independencia del servidor de base de datos permitiendo a las aplicaciones de front-end acceder a los datos de múltiples servidores de base de datos sin tener que escribir código específico para cada servidor.

El uso de este software hace posible al programador generar código genérico para acceder diferentes bases de datos.

Para realizar su labor, el software de comunicaciones opera en dos niveles:

El nivel físico que se encarga de las comunicaciones entre el cliente y el servidor (computadora a computadora). En otras palabras, direcciona la forma en que las computadoras están ligadas. Las ligas físicas incluyen tanto hardware como software. El software de red incluye los protocolos de red. Los protocolos de red son las reglas que gobiernan la forma de actuar entre computadoras para asegurar que

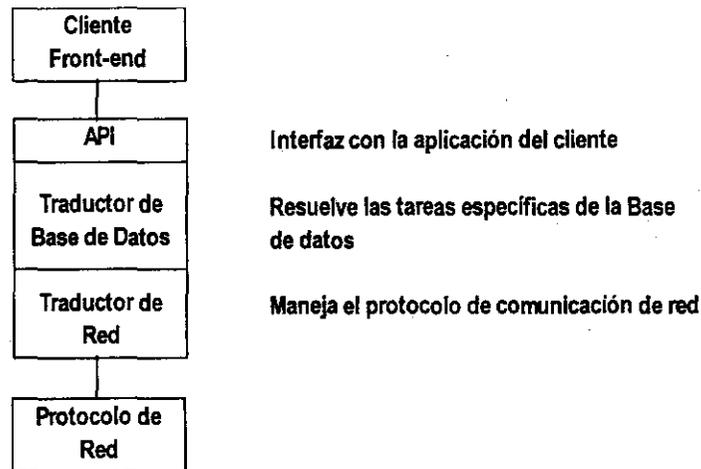
sean capaces de enviar y recibir señales de unas hacia otras. Físicamente, el software de comunicación es, en la mayoría de los casos, la red misma. Como el modelo cliente/servidor permiten al cliente y al servidor residir en la misma computadora, puede existir sin el beneficio de una computadora de red.

El nivel lógico se encarga de la comunicación entre los procesos cliente y servidor (proceso a proceso), esto es, la forma en que los procesos se comunican. Las características lógicas son gobernadas por protocolos de comunicación de proceso a proceso (o interprocesos). Es el nivel en donde la mayor parte de la conversación entre cliente y servidor toma lugar.

Para ilustrar la forma en que estas dos capas trabajan, tomemos un ejemplo: supongamos que ordenamos una pizza por teléfono. Lo primero que hacemos es tomar el teléfono, marcar el número de la pizzeria y esperamos la respuesta. Cuando el teléfono es contestado, nos identificamos, le decimos al vendedor la pizza que queremos, cuántas pizzas queremos y otros detalles. Después, el vendedor nos pide la dirección, nos da el precio y nos da un estimado de tiempo de entrega. Esta operación sencilla requiere la acción tanto del nivel físico como el lógico:

- Las acciones del nivel físico incluyen cambiar nuestra voz a señal analógica y los movimientos subsecuentes que esas señales toman hasta la central telefónica, y de ahí hacia el establecimiento.
  
- Las acciones del nivel lógico son controladas por el vendedor y por uno mismo. Dado que hablamos el mismo lenguaje, solicitamos el servicio en un formato que el vendedor entiende y los detalles de la plática son discutidos exitosamente.

### Componentes del Software del Medio de Comunicación



**Figura 2.4.6.2 Componentes del Software del Medio de Comunicación**

Este software se encuentra dividido en tres componentes principales:

- Interfaz del programa de aplicación (API)
- Traductor de la base de datos
- Traductor de la red

Estos componentes (o sus funciones) están distribuidas generalmente entre varias capas de software que son intercambiables entre sí.

La interfaz del programa de aplicación (API) es pública para las aplicaciones del cliente. El programador interactúa con el software del medio a través de las API's provistas. Las API's permiten a los procesos del cliente ser independientes del servidor de base de datos. Tal independencia que el servidor puede ser cambiado sin requerir que las aplicaciones del cliente sean reescritas por completo.

El traductor de base de datos traduce las llamadas de, por ejemplo, SQL hacia la sintaxis del servidor de base de datos. La capa traductora toma la petición genérica de SQL y la mapea al protocolo del servidor de base de datos. Dado que un servidor de base de datos puede no tener un estándar común, el traductor puede optar por transferir las peticiones genéricas de SQL al formato específico utilizado por la base de datos. Si la solicitud de SQL requiere utilizar datos de dos diferentes servidores de bases de datos, el traductor puede encargarse de la comunicación con cada servidor, extrayendo los datos utilizando el formato común esperado por la aplicación del cliente.

El traductor de red maneja los protocolos de comunicación de la red. Hay que recordar que los servidores de base de datos pueden utilizar cualquier protocolo existente. Sin embargo, si una aplicación del cliente requiere dos bases de datos, una que utiliza TCP/IP y otra que utiliza SPX/IP, la capa de red puede manejar todos los detalles de comunicación entre las bases de datos, de forma transparente para la aplicación cliente. La figura 2.4.6.3 ilustra la interacción entre los protocolos de red.

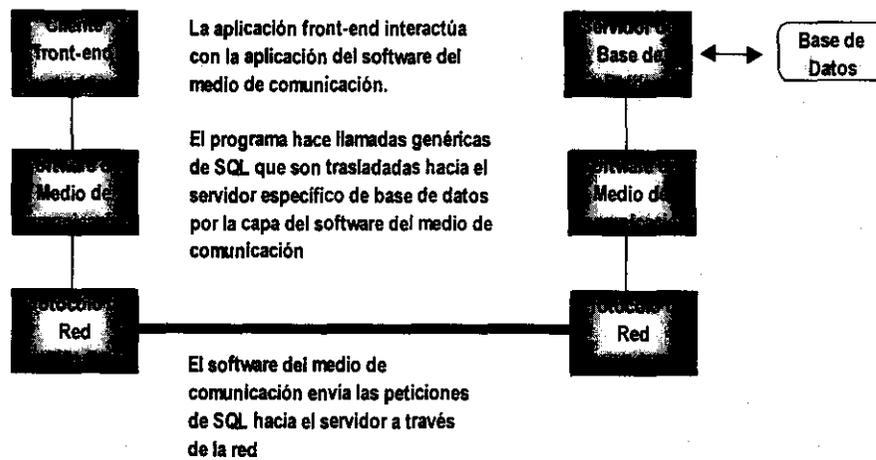


Figura 2.4.6.3 Interacción de los protocolos de red

Es posible identificar los tres beneficios principales del uso de este software:

- Los clientes pueden acceder múltiples bases de datos (aunque sean diferentes)
- Los clientes pueden ser independientes del servidor de base de datos
- Los clientes pueden ser independientes del protocolo de red

Para ilustrar la forma en que todas las piezas trabajan juntas, veremos cómo un cliente acceda dos servidores de bases de datos diferentes. La figura 2.4.6.4 muestra una solicitud de datos de un servidor de base de datos ORACLE y una del tipo SQL. El servidor de base de datos de ORACLE utiliza SQL-Net como su protocolo de comunicación; el servidor de SQL utiliza Named Pipes como su protocolo de comunicación. SQL-Net, una solución propietaria limitada a bases de datos de ORACLE, es utilizada por ORACLE para enviar solicitudes SQL a través de una red. Named Pipes es un protocolo de comunicación interprocesos común en sistemas operativos multitarea, como UNIX y OS/2, y es utilizado en un servidor SQL para manejar las comunicaciones tanto del cliente como del servidor a través de la red.

Hay que notar también que el servidor de ORACLE corre bajo sistema operativo UNIX y utiliza TCP/IP como su protocolo de red. El servidor de SQL corre bajo sistema operativo OS/2 y utiliza NetBIOS como su protocolo de red. En este caso, la aplicación del cliente utiliza una búsqueda (query) de SQL genérica para acceder datos en dos tablas: una de ORACLE y otra de SQL. El traductor de base de datos del software del medio de comunicación contiene dos módulos, uno para cada servidor de base de datos.

Cada módulo maneja los detalles de cada uno de sus protocolos de acceso a base de datos. La capa de traducción de la red se encarga del uso correcto del protocolo de acceso a red para cada base de datos.

Cuando los datos de la búsqueda son regresados por uno de los servidores, son presentados en un formato común para la aplicación del cliente. El usuario final o programador no necesita preocuparse por los detalles de la extracción de información de los servidores.

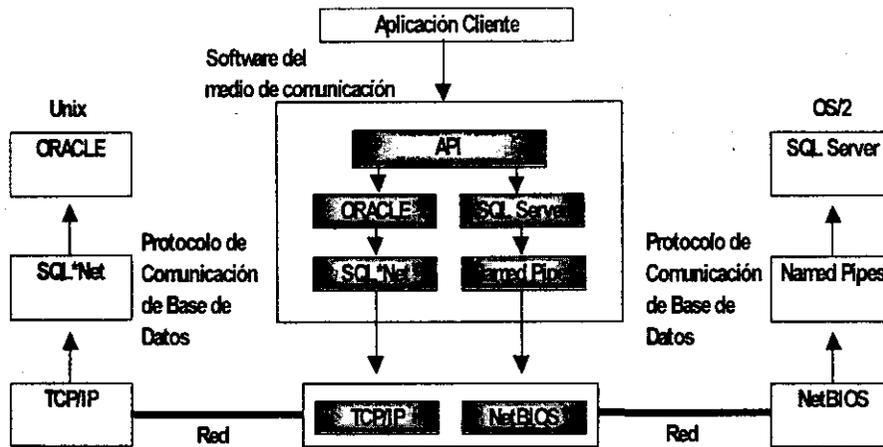


Figura 2.4.6.4 Componentes del Medio de red y sus funciones principales

#### Tareas de Implementación del Modelo Cliente/Servidor

El implementar sistemas cliente/servidor es mejor descrito con una palabra: reto. El diseño de sistemas cliente/servidor difiere en gran forma de los procesos y estilos de los métodos de los sistemas tradicionales. Por ejemplo, los sistemas diseñados con anterioridad, orientados hacia el ambiente de mainframe centralizado y basado en lenguajes de programación tradicionales, difícilmente van a funcionar bien en un ambiente cliente/servidor que está basado en gran diversidad de hardware y de software. En suma, los usuarios modernos son más demandantes y mucho más conocedores de tecnología de computadoras que en años anteriores. Así mismo, los departamentos de sistemas se encuentran en una constante carrera con el reloj del conocimiento para asimilar las nuevas tecnologías que están basadas en diferentes

plataformas, múltiples GUI's, múltiples protocolos de red, etc. En suma, los departamentos de sistemas se deben acoplar con el diseño de aplicaciones rápidamente, para poder satisfacer la autonomía de los usuarios finales.

### **Cliente/Servidor vs. Procesamiento de Datos Tradicional**

Ya conocemos que el modelo cliente/servidor tiende a ser más complejo técnicamente hablando que el modelo tradicional, porque el primero se basa en múltiples plataformas, sistemas operativos y redes. Lo más importante es observar la forma en que el modelo cliente/servidor cambia la forma de ver las tareas más fundamentales del procesamiento de datos.

El cómputo cliente/servidor expande el alcance de los sistemas de información, cambiando la forma en que las cosas se hacían y creando gran responsabilidad por parte de los usuarios finales hacia la información. Los usuarios finales, tradicionalmente, recurrían al departamento de sistemas para adquirir información, hoy en día, ellos crean su propia información creando en una "alberca" común de información y produciendo sus propios reportes y búsquedas para soportar la toma de decisiones.

Esta nueva visión de la información crea una paradoja. Por una parte, los usuarios finales han declarado su independencia del departamento de sistemas, y por otra parte, los usuarios finales se han independizado tanto en la infraestructura cliente/servidor (servidores, redes, software de comunicación y front-end's) que es manejado por el departamento de sistemas. Claramente, el cómputo cliente/servidor ha introducido algunos cambios para bien al procesamiento tradicional de datos:

**Cambio de sistemas propietarios a sistemas abiertos:** La arquitectura tradicional del procesamiento de datos está basada típicamente en soluciones sencillas de un solo vendedor. El integrar diferentes productos con esta arquitectura es una tarea muy difícil y a menudo hasta imposible. El nuevo ambiente cliente/servidor requiere

sistemas que sean fácilmente integrables, esto es, que estén abiertos a otros sistemas.

Cambio de codificación orientada a mantenimiento hacia codificación de análisis, diseño y servicio: Dado el ambiente tradicional de mainframe centralizado, la mayoría de los departamentos de sistemas se enfocan en el mantenimiento de la aplicación. Mientras que los sistemas de cliente/servidor requieren de substancial infraestructura de mantenimiento, por ello un departamento de sistemas que maneje tales sistemas debe gastar gran porción de su tiempo en el soporte a las funciones de los usuarios finales. Los sistemas tradicionales diseñaron ciclos de vida dedicados la mayoría de las veces al código y mantenimiento de las aplicaciones de usuario. El ambiente cliente/servidor cambia el papel de los programadores dejándolos hacer uso de sofisticadas herramientas para liberarlos del código (4GL, CASE, etc.). El precio de utilizar estas herramientas es que los programadores deben gastar más tiempo en el diseño y en el análisis de sistemas, ya que los errores tienden a ser muy costosos. (Por una parte, la autonomía de los usuarios significa que los errores se encontrarán más dispersos). El punto importante es el cambio de codificación a diseño.

Cambio de colección de datos a despliegue de datos: En vez de enfocarse en el almacenamiento de datos centralizado y manejo de datos, un departamento de sistemas basado en cliente/servidor debe concentrarse en mantener los datos disponibles de manera más eficiente y fácil para los usuarios finales.

Cambio de estilo en el manejo de datos centralizados a distribuidos: El manejo de datos típico en el ambiente de mainframe tradicional estaba estructurado muy rígidamente hacia el tipo procedural. El nuevo ambiente de cliente/servidor fuerza a un estilo más flexible en el manejo de los datos. Caracterizado por un manejo descentralizado y enfocado a la toma de decisiones, el cómputo cliente/servidor se enfoca más a intentar resolver los problemas de los usuarios finales y las necesidades del consumidor.

Cambio de estilo de organización vertical, inflexible hacia organizaciones flexibles y horizontales: La estructura de un departamento de sistemas típico era aplanada. El acceso directo a los datos habilitaba a los usuarios individuales para ser más dependientes de la información del departamento de sistemas. En consecuencia, los departamentos de sistemas deben retomar sus fines para acomodarse a las posibilidades de la gente, como PC's, GUI's, etc.

El cambio en el procesamiento de datos introducido por el ambiente cliente/servidor trae consigo el evaluar los componentes de los sistemas de información:

*Hardware:* Los sistemas de información ya no son dependientes de vendedores únicos; en su lugar, existen muchas opciones para escoger una plataforma de hardware.

*Software:* Los sistemas tradicionales se basaban en rutinas de lenguajes tradicionales escritas en algunos 3GL, tales como COBOL o FORTRAN. Todo el procesamiento era realizado en el mainframe central. Los sistemas cliente/servidor son el resultado de la integración de muchas rutinas creadas por y soportadas por GUI's, bases de datos, redes y comunicaciones. Estos nuevos sistemas se acomodan al procesamiento de información mediante muchos componentes que se integran para formar uno solo.

*Datos:* Tradicionalmente, los datos se encontraban centralizados en repositorios únicos. Los nuevos sistemas distribuyen los datos a través de múltiples computadoras, colocando los datos muy cerca de los usuarios. Con ello, los múltiples formatos de datos (sonido, imagen, video, texto, etc.) se encuentran disponibles.

*Procedimientos:* Los sistemas tradicionales se basaban en procedimientos que eran muy rígidos y complejos. Los sistemas actuales intercambian los procedimientos existentes para hacerlos más flexibles y descentralizados.

*Gente:* El cómputo cliente/servidor cambia los papeles de la gente y sus funciones. Son necesarias nuevas habilidades para soportar y utilizar la nueva tecnología, lo cual conlleva entrenamiento y reentrenamiento para mantenerse al día. Tales cambios no tienen límite para el departamento de sistemas.

La implementación de sistemas cliente/servidor representa un gran cambio en cualquier organización. Tal cambio debe ser manejado apropiadamente dentro de la misma organización. Dado que el cambio se refleja tanto en la información como en la filosofía de acceso a la misma, los departamentos de sistemas deben trabajar para adecuarse y cambiar del estilo tradicional al estilo cliente/servidor. Por lo anterior es esencial una implementación coherente e integrada.

#### **2.4.7. Modelos Cliente / Servidor de Dos y Tres Niveles**

La mayoría de las configuraciones Cliente/Servidor se basan en el modelo de dos capas, que consiste en un nivel *cliente* que solicita los servicios de un *servidor*. También es factible que un servidor actúe como cliente de otro servidor, en una especie de arquitectura jerárquica, pero que sigue siendo un modelo de dos capas encadenado.

Este tipo de arquitectura presenta ciertos problemas en el caso de aplicaciones complejas con muchos clientes, bases de datos y entradas heterogéneas, así como a transacciones que abarcan varias bases de datos bajo diferentes DBMS's (Data Base Management Systems). Un DBMS tiene una capacidad limitada en cuanto a efectuar la actualización "en dos capas" con múltiples manejadores de bases de datos.

La arquitectura de tres capas resuelve esta problemática. En este caso, el cliente se dedica a la lógica de presentación y a los servicios, contando con un API para invocar a la aplicación en la capa intermedia. A su vez, el servidor de base de datos está dedicado a los servicios de datos y de archivos. El nivel intermedio consiste en un servidor de aplicaciones en donde se ejecuta la lógica del negocio y desde el cual la lógica de datos invoca a los servicios de datos.

El modelo de tres capas presenta otros beneficios para el desarrollo de aplicaciones. Primero, el desarrollador de buenas interfaces requiere de habilidades y capacitación especial. No todos aquellos programadores que escriben una buena lógica del negocio pueden escribir un código de presentación. La separación de plataformas en las que se ejecutan estos componentes facilita la organización de la implementación, a efecto de reflejar las habilidades peculiares requeridas.

El motivo por el cual no se ha difundido la arquitectura de tres capas es que la computación en cliente/servidor apoyada por los proveedores de manejadores de bases de datos ha tratado de ser autosuficiente, en tanto que el modelo de tres capas y los productos que ayudan a su implementación provienen del universo de los sistemas abiertos. También, la mayoría de las implementaciones cliente/servidor han estado orientadas al trabajo en grupos y a las aplicaciones departamentales, donde el modelo de dos capas se desempeña adecuadamente.

Por otra parte, no se debe dejar de considerar los costos que implica la implementación de una arquitectura de tres capas, ya que se requiere al servidor de la capa intermedia, la cual se encarga de la lógica del negocio.

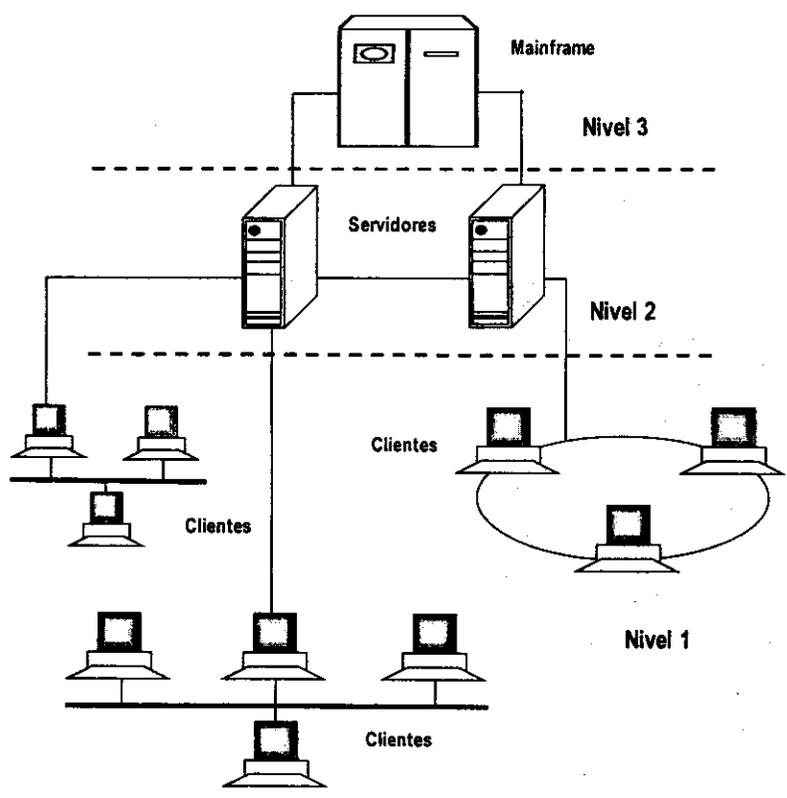


Figura 2.4.7.1 Niveles de la Arquitectura Cliente/Servidor

## 2.5. Metodologías de Ingeniería de Software Orientadas a Objetos (Ventajas y Desventajas)

Vivimos en un mundo de objetos. Estos objetos existen en la naturaleza, en entidades hechas por el hombre, en los negocios y en los productos que usamos. Ellos pueden ser clasificados, descritos, organizados, combinados, manipulados y creados. Por ello no debe sorprendernos que se proponga una visión orientada a objetos para la creación de software de computadora, una abstracción que modela el mundo de forma tal que nos ayuda a entenderlo y gobernarlo mejor.

Un enfoque orientado a objetos para el desarrollo de software se propuso por primera vez a finales de los 60's. Sin embargo, las tecnologías de objetos han necesitado casi veinte años para llegar a ser ampliamente usadas. Durante la primera mitad de la década de los 90's, la ingeniería del software orientada a objetos se convirtió en el paradigma de elección para muchos productores de software y un creciente número de sistemas de información y profesionales de la ingeniería.

Las tecnologías de objeto llevan un número de beneficios inherentes que proporcionan ventajas a los niveles de dirección y técnico.

Las tecnologías de objeto llevan a reutilizar y la reutilización (de componentes de software) lleva a un desarrollo de software más rápido y programas de mejor calidad. El software orientado a objetos es más fácil de mantener debido a que su estructura es inherentemente descompuesta.

Esto lleva a menores efectos colaterales cuando se deben hacer cambios y provoca menos frustración en el ingeniero del software y del cliente. Adicionalmente, los sistemas orientados a objetos son más fáciles de adaptar y escalar.

## El Paradigma Orientado a Objetos

Durante muchos años el término orientado a objetos (OO) se usó para significar un enfoque de desarrollo de software que usaba uno de los lenguajes orientados a objetos (Ada 95, C++, Eiffel, SmallTalk). Hoy en día el paradigma OO encierra una completa visión de la ingeniería del software, tal y como lo describiera Edward Berard:

*“Los beneficios de la tecnología orientada a objetos se fortalecen si se usa antes y durante el proceso de ingeniería del software. Esta tecnología orientada a objetos considerada debe hacer sentir su impacto en todo el proceso de ingeniería del software. Un simple uso de programación orientada a objetos (POO) no brindará los mejores resultados. Los ingenieros del software y sus directores deben considerar estos elementos como análisis de requisitos orientados a objetos (AROO), del diseño orientado a objetos (DOO), análisis del dominio orientado a objetos (ADOO), de sistemas de gestión de bases de datos orientadas a objeto (SGBDOO) y la ingeniería del software orientada a objetos asistida por computadora (ISOOAC).”*

El proceso OO se mueve a través de una espiral evolutiva que comienza con comunicación con el usuario. Es aquí donde se define el dominio del problema y se identifican las clases básicas del problema. La planificación y el análisis de riesgos establecen una base para el plan del proyecto OO. El trabajo técnico asociado con la ingeniería de software OO sigue el camino iterativo mostrado en la figura 2.5.1 en la caja gris. La ingeniería del software OO hace hincapié en la reutilización. Por lo tanto, las clases se buscan en una biblioteca (de clases OO existentes) antes de construirse. Cuando una clase no puede encontrarse en la biblioteca, el desarrollador de software aplica análisis orientado a objetos (AOO), diseño orientado a objetos (DOO), programación orientada a objetos (POO) y pruebas orientadas a objetos (PrOO) para crear la clase y los elementos de la clase. La nueva clase se pone en la biblioteca de tal manera que pueda ser reutilizada en el futuro.

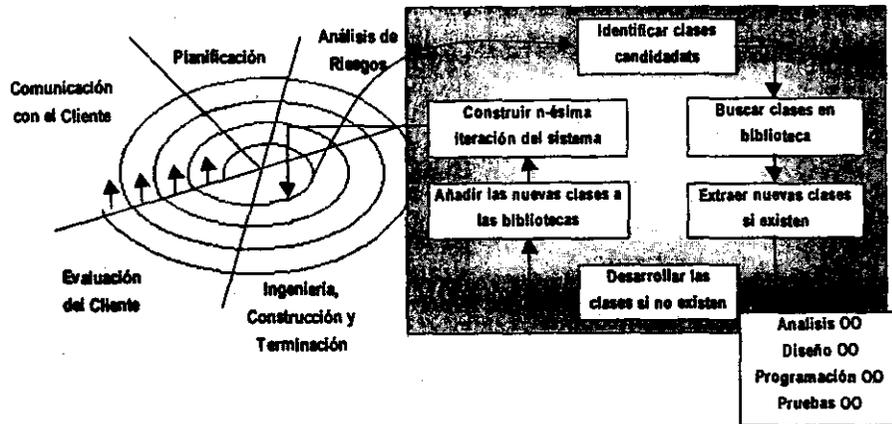


Figura 2.5.1 Modelo del Proceso Orientado a Objetos

La visión orientada a objetos demanda un enfoque evolutivo de la ingeniería del software. Es muy difícil definir las clases necesarias para un gran sistema o producto en una sola iteración. A medida que el análisis OO y los modelos de diseño evolucionan, se vuelve aparente la necesidad de clases adicionales. Es por esta razón por lo que el paradigma anteriormente descrito trabaja mejor para la OO.

### Conceptos de Orientación a Objetos

Cualquier discusión sobre ingeniería del software OO debe comenzar por el término "orientado a objetos". Durante años han existido opiniones diferentes sobre la respuesta correcta, a continuación se sintetizarán las opiniones más comunes.

Para entender la visión orientada a objetos, consideremos un objeto del mundo real, una silla. La **silla** es un miembro (o instancia) de una clase mucho más grande de objetos que llamaremos **mobiliario**. Un conjunto de atributos genérico puede asociarse con cada objeto en la clase **mobiliario**. Por ejemplo, todo mueble tiene un costo, dimensiones, peso, localización y color, entre otros muchos posibles atributos. Éstos son aplicables a cualquier elemento sobre el que se hable, una mesa o silla,

un sofá o un armario. Como **silla** es un miembro de la clase **mobiliario**, silla *hereda* todos los atributos definidos para la clase, como se ilustra en la figura 2.5.2.

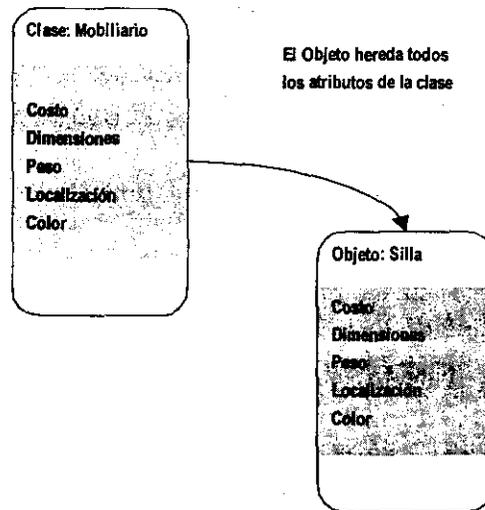


Figura 2.5.2 Herencia de Clase a Objeto

Una vez definida la clase, los atributos pueden reutilizarse al crear nuevas instancias de la clase. Por ejemplo, supongamos que debemos definir un nuevo objeto llamado **sillesa** (un cruce entre una silla y una mesa) que es un miembro de la clase **mobiliario**. La **sillesa** hereda todos los atributos de mobiliario.

Hemos intentado definir una clase describiendo sus atributos, pero falta algo. Todo objeto en la clase **mobiliario** puede manipularse de varias maneras. Puede comprarse y venderse, modificarse físicamente (por ejemplo, se puede eliminar una pata o pintar el objeto de otro color) o moverse de un lugar a otro. Cada una de estas *operaciones* (o *servicios* o *métodos*) modificará uno o más atributos del objeto. Por ejemplo, si el atributo *localización* es un dato compuesto definido, como:

localización = edificio + piso + habitación

entonces una operación denominada *mover* modificaría uno ó más de los elementos dato (*edificio, piso o habitación*) que conforman el atributo *localización*. Para hacer esto, *mover* debe tener "conocimiento" sobre estos elementos. La operación *mover* puede usarse para una silla o para una mesa, debido a que ambas son instancias de la clase mobiliario. Todas las operaciones válidas (por ejemplo, comprar, vender, pesar) de la clase mobiliario están conectadas a la definición del objeto como se muestra en la figura 2.5.3 y son heredadas por todas las instancias de esta clase.

El objeto *silla* (y todos los objetos en general) encapsula datos (los valores de los atributos que definen la silla), operaciones (las acciones que se aplican para cambiar los atributos de la silla), otros objetos (pueden definirse objetos compuestos), constantes (fijar valores) y otra información relacionada. El encapsulamiento significa que toda esta información se encuentra empaquetada bajo un nombre y puede reutilizarse como una especificación o componente de programa.

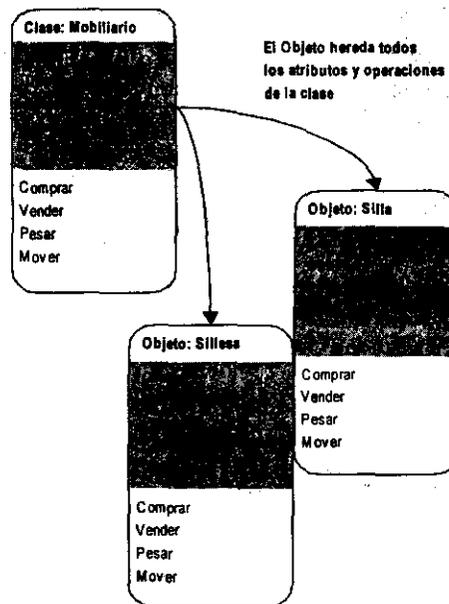


Figura 2.5.3 Herencia de operaciones de Clase a Objeto

A continuación se dará una definición más formal de la "orientación a objetos":

Orientación a objetos = objetos+clasificación+herencia+comunicación

### Clases y Objetos

Los conceptos fundamentales que llevan a un diseño de alta calidad son igualmente aplicables a sistemas desarrollados usando métodos convencionales y orientados a objetos. Por esta razón, un modelo OO de software de computadora debe exhibir abstracciones de datos y procedimientos que conducen a una modularidad eficaz. Una clase es un concepto OO que encapsula las abstracciones de datos y procedimientos que se requieren para describir el contenido y comportamiento de alguna entidad del mundo real. A continuación ilustramos la descripción de una clase y sus objetos derivados:

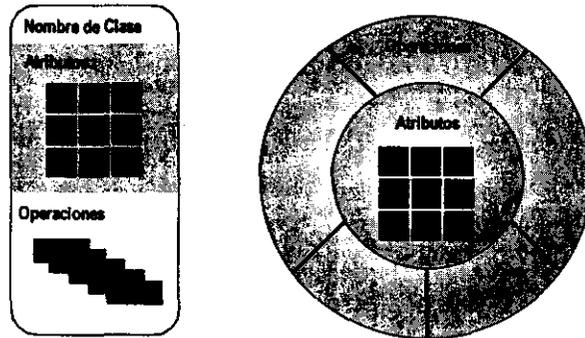


Figura 2.5.4 Representación alternativa de una clase Orientada a Objetos

Las abstracciones de datos (atributos) que describen la clase están encerradas por una "muralla" de abstracciones procedurales (llamadas operaciones, métodos o servicios) capaces de manipular los datos de alguna manera. La única forma de alcanzar los atributos y operar sobre ellos es ir a través de alguno de los métodos que forman la muralla. Por lo tanto, la clase encapsula datos (dentro de la muralla) y el proceso que manipula los datos (los métodos que componen la muralla). Esto

posibilita el ocultamiento de información y reduce el impacto de efectos colaterales asociados a cambios. Como estos métodos tienden a manipular un número limitado de atributos, esto es una alta cohesión, y como la comunicación ocurre sólo a través de los métodos que encierra la "muralla", la clase tiende a un acoplamiento con otros elementos del sistema. Todas estas características del diseño conducen a software de alta calidad.

Puesto de otra manera, una clase es una descripción generalizada (por ejemplo una plantilla, un patrón o una copia) que describe una colección de objetos similares. Por definición, todos los objetos que existen dentro de una clase heredan sus atributos y las operaciones disponibles para la manipulación de los atributos. Una *superclase* es una colección de clases y una *subclase* es una instancia de una clase.

Estas definiciones implican la existencia de una *jerarquía de clases* en la cual los atributos y operaciones de la superclase son heredados por subclases que pueden añadir, cada una de ellas, atributos "privados" y métodos. Una jerarquía de clases para la clase mobiliario se ilustra en la figura 2.5.5.

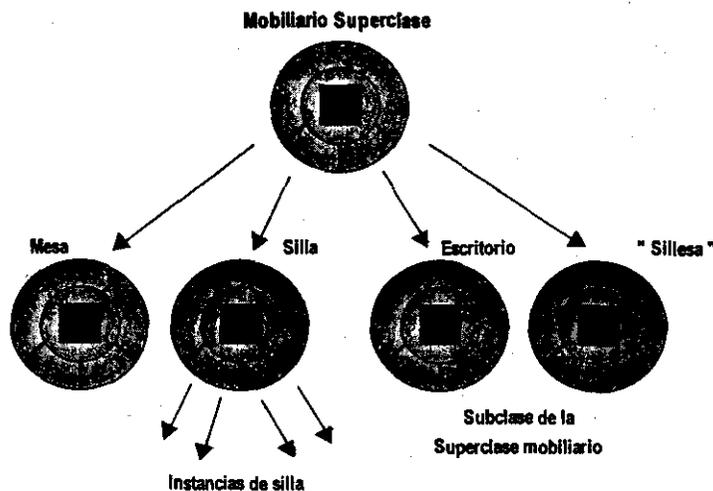


Figura 2.5.5 Jerarquía de Clases

## Atributos

Ya hemos visto que los atributos están asociados a clases y objetos, y que ellos describen la clase o el objeto de alguna manera. A continuación una definición de atributos:

*"Las entidades de la vida real están a menudo descritas con palabras que indican características estables. La mayoría de los objetos físicos tienen características tales como forma, peso, color y tipo de material. Las personas tienen características incluyendo fecha de nacimiento, padres, nombres y color de ojos. Una característica puede verse como una relación binaria entre una clase y cierto dominio"*

La relación binaria anteriormente señalada implica que un atributo puede tomar un valor definido por un dominio enumerado. En la mayoría de los casos, un dominio es simplemente un conjunto de valores específicos. Por ejemplo, suponga que una clase **coche** tiene un atributo *color*. El dominio de valores de *color* es {blanco, negro, plata, gris, azul, rojo, amarillo, verde}. En situaciones más complejas, el dominio puede ser un conjunto de clases. Continuando el ejemplo, la clase **coche** también tiene un atributo *motor* que abarca los siguientes valores de dominio: valor {15 válvulas opción económica, 16 válvulas opción deportiva, 24 válvulas opción superdeportiva, 32 válvulas opción de lujo}. Cada una de las opciones indicadas tiene un conjunto de atributos específicos de ella.

Las características (valores del dominio) pueden aumentarse asignando un valor por defecto (característica) a un atributo. Por ejemplo, el atributo *motor* destacado antes tiene el valor por defecto 16 válvulas opción deportiva. Esto puede ser también útil para asociar una probabilidad con una característica particular a través de pares <valor,probabilidad>. Consideremos el atributo *color* para **coche**. En algunas aplicaciones (por ejemplo, planificación de la producción) puede ser necesario asignar una probabilidad a cada uno de los colores (blanco y negro son más probables como colores de coches).

## Operaciones, métodos y servicios

Un objeto encapsula datos (representados como una colección de atributos) y los algoritmos que procesan estos datos. Estos algoritmos son llamados *operaciones*, *métodos* o *servicios* y pueden ser vistos como módulos en un sentido convencional.

Cada una de las operaciones encapsuladas por un objeto proporciona una representación de uno de los comportamientos del objeto. Por ejemplo, la operación *DeterminaColor* para el objeto **automóvil** extraerá el color almacenado en el atributo *color*. La consecuencia de la existencia de esta operación es que la clase **automóvil** ha sido diseñada para recibir un estímulo (llamaremos al estímulo un "mensaje") que requiere el color de una instancia particular de una clase. Cada vez que un objeto recibe un estímulo, éste inicia un cierto comportamiento. Éste puede ser tan simple como determinar el color del coche o tan complejo como la iniciación de una cadena de estímulos que se pasan entre una variedad de objetos diferentes. En este último caso, consideremos un ejemplo en el cual el estímulo inicial recibido por el objeto No. 1 resulta en una generación de otros dos estímulos que se envían al objeto No. 2 y No. 3. Las operaciones encapsuladas por el segundo y tercer objetos actúan sobre el estímulo devolviendo información necesaria para el primer objeto. El objeto No. 1 usa la información devuelta para satisfacer el comportamiento demandado por el estímulo inicial.

## Mensajes

Los mensajes son el medio a través del cual los objetos interactúan. Un mensaje estimula la ocurrencia de cierto comportamiento en el objeto receptor. El comportamiento se realiza cuando se ejecuta una operación.

La interacción entre objetos se ilustra esquemáticamente en la figura 2.5.6.

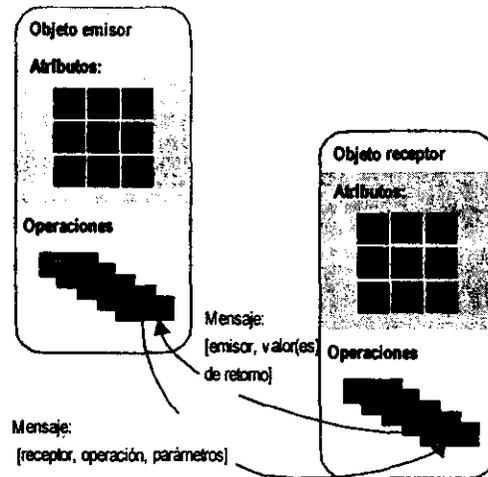


Figura 2.5.6 Paso de mensajes entre objetos.

Una operación dentro de un *objeto emisor* genera un mensaje de la forma:

**Mensaje:** [destino, operación, parámetros]

Donde *destino* define el objeto *receptor* el cual es estimulado por el mensaje, *operación* se refiere al método que recibe el mensaje y *parámetros* proporciona información requerida para el éxito de la operación.

A continuación una definición de intercambio entre objetos:

“Se solicita de un objeto que ejecute una de sus operaciones enviándole un mensaje que le informa acerca de lo que debe hacer. El objeto receptor responde al mensaje eligiendo primero la operación que implementa el nombre del mensaje, ejecutando dicha operación y después devuelve el control al objeto que origina la llamada”

El paso de mensajes mantiene comunicados un sistema orientado a objetos. Los mensajes proporcionan una visión interna del comportamiento de objetos individuales, y del sistema OO como un todo.

### **Encapsulamiento, Herencia y Polimorfismo**

Existen tres características de los sistemas orientados a objetos que los hacen únicos. Como ya hemos observado, las clases OO y los objetos derivados de ella encapsulan los datos y las operaciones que trabajan sobre estos en un paquete único. Esto proporciona un número importante de beneficios:

- Los detalles de implementación interna de datos y procedimientos están ocultos al mundo exterior (ocultamiento de la información). Esto reduce la propagación de efectos colaterales cuando ocurren cambios.
- Las estructuras de datos y las operaciones que las manipulan están mezcladas en una entidad sencilla: la clase. Esto facilita la reutilización de componentes.
- Las interfaces entre objetos encapsulados están simplificadas. Un objeto que envía un mensaje no se tiene que preocupar de los detalles de estructuras de datos internas en el objeto receptor. Por tanto se simplifica la interacción y el acoplamiento del sistema tiende a reducirse.

La herencia es una de las diferencias clave entre sistemas convencionales y sistemas OO. Una subclase Y hereda todos los atributos y operaciones asociadas con su superclase X. Esto significa que todas las estructuras de datos y algoritmos originalmente diseñados e implementados para X están inmediatamente disponibles para Y (no es necesario más trabajo extra).

La reutilización se realiza directamente. Cualquier cambio en los datos o en las operaciones contenidas dentro de una superclase se hereda inmediatamente por todas las subclases que se derivan de la superclase. Debido a esto, la jerarquía de

clases se convierte en un mecanismo a través del cual los caminos (a altos niveles) pueden propagarse inmediatamente a través de todo el sistema.

Es importante destacar que en cada nivel de la jerarquía de clases, pueden añadirse nuevos atributos y operaciones a aquellos que han sido heredados de niveles superiores de la jerarquía.

De hecho, cada vez que se debe crear una nueva clase, el ingeniero tiene varias opciones:

- La clase puede diseñarse y construirse de la nada. Esto es, no se usa la herencia.
- La jerarquía de clases puede ser rastreada para determinar si una clase ascendente contiene la mayoría de los atributos y operaciones requeridas.
- La nueva clase hereda de su clase ascendente, y pueden añadirse los elementos requeridos.
- La jerarquía de clases puede reestructurarse de tal manera que los atributos y operaciones requeridas puedan heredarse por la nueva clase.
- Las características de una clase existente pueden sobrescribirse y se pueden implementar versiones privadas de atributos u operaciones para la nueva clase.

Para ilustrar cómo la reestructuración de la jerarquía de clases puede conducir a la clase deseada, consideremos el ejemplo que se ilustra en las siguientes figuras:

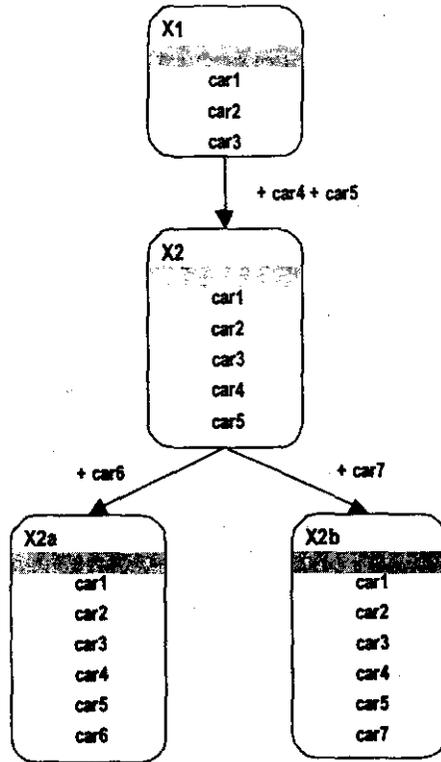


Figura 2.5.7a Jerarquía de clases original

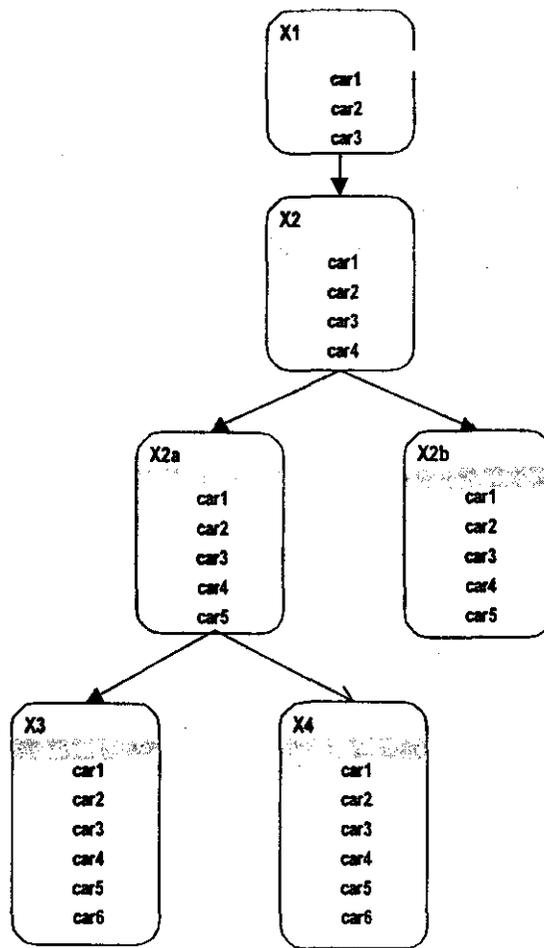


Figura 2.5.7b Jerarquía de clases original

La jerarquía de clases mostrada en la figura 2.5.7a nos permite derivar las clases X3 y X4 con las características (que en este caso pueden ser atributos u operaciones) 1,2,3,4,5 y 6, y 1,2,3,4,5 y 7, respectivamente. Ahora, supongamos que deseamos crear una nueva clase solamente con las características 1,2,3,4, y 8. Para derivar esta clase, llamada X2b en el ejemplo, la jerarquía debe reestructurarse como se muestra en la figura 2.5.7b, es importante notar que la reestructuración de la jerarquía puede ser difícil, y por esta razón, se usa a veces la *anulación*.

En esencia la anulación ocurre cuando los atributos y operaciones se heredan de manera normal, pero después son notificados según las necesidades específicas de la nueva clase; lo cuál nos lleva a una anotación: "la herencia no es transitiva".

En algunos casos, es tentador heredar algunos atributos y operaciones de una clase y otros de otra clase. Esta acción se llama *herencia múltiple* y es controvertida. En general, la herencia múltiple complica la jerarquía de clases y crea problemas potenciales en el control de la configuración. Como las secuencias de herencia múltiple son más difíciles de seguir, los cambios en la definición de una clase que reside en la parte superior de la jerarquía pueden tener impactos no deseados originalmente en las clases definidas en zonas inferiores de la arquitectura.

El *polimorfismo* es una característica que reduce en gran medida el esfuerzo necesario para extender un sistema OO. Para entender el polimorfismo, consideremos una aplicación convencional que debe dibujar cuatro tipos diferentes de gráficos: gráficos de líneas, gráficos de pie, histogramas y diagramas. Idealmente, una vez que se han recogido los datos necesarios para un tipo particular de gráfico, éste debe dibujarse por si solo. Para realizar esto en una aplicación convencional (y mantener la cohesión entre módulos), es necesario desarrollar módulos de dibujo para cada tipo de gráfico. Entonces, dentro del diseño para cada tipo de gráfico, se debe añadir una forma de control semejante a la que sigue:

Case of tipo\_grafico:

If tipo\_grafico = grafico\_linea then DibujarLinea (datos) ;

If tipo\_grafico = grafico\_pie then DibujarPie (datos) ;

If tipo\_grafico = grafico\_histograma then DibujarHisto (datos) ;

If tipo\_grafico = grafico\_diagrama then DibujarDiagrama (datos) ;

End case;

Aunque este diseño es razonablemente evidente, añadir nuevos tipos de gráficos puede ser artesanal o complicado. Un nuevo módulo de dibujo deberá crearse para cada tipo de gráfico y entonces la lógica de control deberá ser actualizada para cada gráfico.

Para resolver este problema en un sistema OO, cada uno de los gráficos mencionados arriba se convierte en una subclase de una clase general llamada **gráfico**. Usando un concepto llamado *sobrecarga*, cada subclase define una operación llamada *dibujar*. Un objeto puede enviar un mensaje *dibujar* a cualquiera de los objetos de cualquiera de las subclases. El objeto que recibe el mensaje invocará su propia operación *dibujar* para crear el gráfico apropiado. Por esto, el diseño mostrado anteriormente se reduce a:

tipo\_grafico dibujar

Cuando se debe añadir un nuevo tipo de gráfico al sistema, se crea una subclase con su propia operación *dibujar*. Pero no se requieren cambios dentro de un objeto que quiere dibujar un gráfico pues mensaje *tipo\_grafico dibujar* permanece sin cambiar. En resumen, el polimorfismo permite que un número de operaciones diferentes tengan el mismo nombre. Esto reduce el acoplamiento entre objetos, haciendo cada uno más independiente.

## **Análisis Orientado a Objetos (AOO)**

La popularidad de las tecnologías orientadas a objetos ha generado docenas de métodos de AOO. Cada uno de ellos introduce un proceso para el análisis de un producto o sistema, un conjunto de modelos que evoluciona fuera del proceso, y una notación que posibilita al ingeniero del software crear cada modelo de una manera consistente. A continuación mencionamos algunos de los métodos más populares de AOO.

### **El Método de Booch**

Este método abarca un "micro proceso de desarrollo" y un "macro proceso de desarrollo". El nivel micro define un conjunto de tareas de análisis que se reapiplan en cada etapa en el macro proceso. Por esto se mantiene un enfoque evolutivo. El método Booch está soportado por una variedad de herramientas automatizadas. A continuación describimos brevemente el micro proceso de desarrollo del AOO de Booch:

➤ **Identificar clases y objetos**

- Proponer objetos candidatos
- Conducir el análisis de comportamiento
- Identificar escenarios relevantes
- Definir atributos y operaciones para cada clase

➤ **Identificar la semántica de clases y objetos**

- Seleccionar y analizar escenarios
- Asignar responsabilidades para alcanzar el comportamiento deseado
- Dividir las responsabilidades para equilibrar el comportamiento
- Seleccionar un objeto y enumerar sus papeles y responsabilidades
- Definir operaciones para satisfacer las responsabilidades
- Buscar colaboraciones entre objetos

- **Identificar relaciones entre clases y objetos**
  - Definir las dependencias que existen entre objetos**
  - Describir el papel de cada objeto participante**
  - Validar los escenarios para la revisión completa**
  
- **Realizar una serie de refinamientos**
  - Producir diagramas apropiados para el trabajo realizado en los puntos anteriores**
  - Definir jerarquías de clase apropiadas**
  - Crear agrupamientos basados en clases comunes**
  
- **Implementar clases y objetos (en el contexto del AOO esto implica complementar el modelo de análisis)**

### **El Método de Coad y Yourdon**

Este método se considera, con frecuencia, como uno de los métodos del AOO más sencillos de aprender. La notación del modelado es relativamente simple y las reglas para desarrollar el modelo de análisis son evidentes:

- **Identificar objetos usando el criterio de “qué buscar”**
- **Definir una estructura de generalización-especificación**
- **Definir una estructura de todo-parte**
- **Identificar temas (representaciones de componentes de subsistemas)**
- **Definir atributos**
- **Definir servicios**

### **El Método de Jacobson**

También llamado ISOO (Ingeniería del Software Orientada a Objetos), el método de Jacobson es una versión simplificada de Objectory, un método patentado, también desarrollado por Jacobson. Este método se diferencia de los otros por el gran énfasis en el *caso de uso*, una descripción o escenario que describe cómo el usuario interactúa con el producto o sistema:

- Identificar los usuarios del sistema y sus responsabilidades globales
- Construir un modelo de requisitos
  - Definir los actores y sus responsabilidades
  - Identificar los casos de uso para cada actor
  - Preparar una visión inicial de los objetos del sistema y sus relaciones
  - Revisar el modelo usando los casos de uso como escenarios para determinar su validez
- Construir un modelo de análisis
  - Identificar objetos de interfaz usando información del tipo actor-interacción
  - Crear vistas estructurales de los objetos de interfaz
  - Representar el comportamiento del objeto
  - Aislar subsistemas y modelos para cada uno
  - Revisar el modelo usando casos de uso como escenarios para determinar su validez

### **El Método de Rumbaugh**

Rumbaugh y sus colegas desarrollaron la Técnica del Modelado de Objetos (OMT) para el análisis, diseño del sistema y diseño del nivel de objetos. La actividad de análisis crea tres modelos: el modelo de objeto (una representación de objetos, clases, jerarquías y relaciones), el modelo dinámico (una representación del

comportamiento del sistema y los objetos) y el modelo funcional (una representación a alto nivel de flujo de información a través del sistema similar al DFD):

- **Desarrollar una declaración del ámbito del problema**
- **Desarrollar un modelo de objetos**
  - Identificar clases relevantes al problema
  - Definir atributos y asociaciones
  - Definir enlaces de objetos
  - Organizar las clases de objetos usando herencia
  
- **Desarrollar un modelo dinámico**
  - Preparar escenarios
  - Definir eventos y desarrollar una taza de eventos para cada escenario
  - Construir un diagrama de flujo de eventos
  - Desarrollar un diagrama de estados
  - Revisar el comportamiento para comprobar consistencia y complejión
  
- **Desarrollar un modelo funcional para el sistema**
  - Identificar entradas y salidas
  - Usar diagramas de flujo de datos para representar transformaciones del flujo
  - Desarrollar EP (Especificación del Proceso) para cada función
  - Especificar criterios de restricciones y optimización

#### **El Método de Wirfs-Brock**

El método de Wirfs-Brock no hace una distinción clara entre las tareas de análisis y diseño. En su lugar, propone un proceso continuo que comienza con la valoración de una especificación del cliente y termina con el diseño:

- **Evaluar la especificación del cliente**

- Usar un análisis gramatical para extraer clases candidatas de la especificación
- Agrupar las clases en un intento de determinar superclases
- Definir responsabilidades para cada clase
- Asignar responsabilidades para cada clase
- Identificar relaciones entre clases
- Definir colaboraciones entre clases basándose en sus responsabilidades
- Construir representaciones jerárquicas de clases para mostrar relaciones de herencia
- Construir un gráfico de colaboraciones para el sistema

Aunque la terminología y etapas del proceso para cada uno de estos métodos de AOO difieren, los procesos generales de AOO son en realidad muy similares. Para realizar un análisis orientado a objetos, el ingeniero de software debe ejecutar las siguientes etapas genéricas:

- Obtener los requisitos del cliente para el Sistema OO
- Identificar escenarios o casos de uso
- Construir un modelo de requisitos
- Seleccionar clases y objetos usando los requisitos básicos como guías
- Identificar atributos y operaciones para cada objeto del sistema
- Definir estructuras y jerarquías que organicen las clases
- Construir un modelo objeto-relación
- Construir un modelo objeto-comportamiento
- Revisar el modelo de análisis OO en relación a los casos de uso/escenarios

## 2.6 CONCEPTOS BASICOS DE POWERBUILDER.

En el pasado, con los lenguajes procedurales tradicionales como Basic, COBOL o C, el programador controlaba la operativa de ejecución de una aplicación. Un programa controlaba cuándo se abrían y cerraban las ventanas y el movimiento del cursor de un campo a otro. El programa en ejecución determinaba la secuencia de eventos. El programador determinaba dicha secuencia codificándola en el programa.

En las aplicaciones modernas con interfaces gráficas, es el usuario el que gestiona la secuencia de eventos que se producen en un programa en ejecución. El usuario determina cuándo comienza la aplicación, cuándo se abre o se cierra una ventana, cuándo se introduce el siguiente campo o cuándo finaliza la aplicación. En lugar del programa o el programador, es el usuario el que controla los eventos.

En un entorno en el que el usuario es el que provoca los eventos a través de una interfaz gráfica de usuario, es difícil realizar aplicaciones con un lenguaje procedural. Es complicado escribir un programa para un entorno orientado a eventos cuando el programador es el que debe determinar la secuencia de eventos. Sin embargo, es fácil escribir programas para este entorno con los nuevos lenguajes orientados a objetos tal como Power Builder, éste está diseñado para trabajar con objetos y eventos. Tiene gran cantidad de facilidades predefinidas para la gestión de ventanas, eventos y objetos.

La realización de aplicaciones con Power Builder es fácil. Tiene herramientas para la creación de las aplicaciones y sus objetos, para la conexión a las fuentes de datos y para poder responder a los eventos que van ocurriendo mientras la aplicación se está ejecutando. Todo esto permite construir aplicaciones industriales fiables, de manera rápida, fácil y eficiente.

Una interfaz gráfica de usuario es un entorno orientado a objetos, dirigido a eventos. Una imagen, un icono, un gráfico, las ventanas, así como los controles y otros objetos que aparecen en ellas, son todos objetos.

Los usuarios pueden provocar varios eventos. Un usuario puede hacer clic (oprimir el botón del mouse) sobre una ventana, Power Builder responde a estos eventos dando respuesta al evento y controlando la operativa de la aplicación.

Los objetos tales como ventanas y controles tienen cada uno una lista de eventos asociados. El objeto aplicación de Power Builder tiene cuatro eventos asociados:

- Close
- Open
- Idle
- System error

El objeto ventana tiene muchos eventos asociados, entre otros:

- Open
- Close
- Move
- Re-size

Algunos objetos de Power Builder, tales como las estructuras de datos y las funciones de usuario, no tienen eventos asociados. Para poder realizar programas con Power Builder, quizá lo más difícil sea conocer los eventos y cómo gestionarlos. Se deben conocer los diferentes eventos de cada objeto y cómo realizar procedimientos y funciones para dichos eventos.

Por ejemplo, todas las ventanas tienen el evento open. Al abrirse una ventana se provoca el evento open. Se puede escribir un programa PowerScript para dicho

evento. Cuando se abre la ventana, el procedimiento se ejecuta. Por ejemplo, se puede incluir en el procedimiento del evento open de una ventana la conexión a una fuente de datos.

Cualquier interfaz gráfica de usuario es un entorno con gran cantidad de eventos. Debido a esto, Power Builder tiene muchas funciones para realizar la gestión de dichos eventos.

Power Builder tiene todas las herramientas necesarias para construir aplicaciones cliente - servidor. Permite realizar aplicaciones que se ejecuten en una computadora cliente, comunicándose entre sí, utilizando protocolos de red estándar. Las aplicaciones pueden acceder a datos locales o remotos provenientes de una serie de fuentes. En los servidores se pueden ejecutar la mayor parte de los sistemas de bases de datos más importantes. Los servidores pueden ser computadoras personales con Novell y Microsoft SQL; estaciones de trabajo o mini-computadores con Oracle, Informix o Sybase; o Mainframes con DB2.

#### **Herramientas de Power Builder.**

Al arrancar Power Builder se abrirá una ventana como la que se muestra en la figura 2.6.1. Power Builder tiene una serie de painters (lugar de diseño). Cada painter es una herramienta con un propósito particular. Cada icono situado en la Power Bar (barra de herramientas de Power) permite el acceso a un painter diferente. Cada painter de Power Builder soporta una funcionalidad distinta, al abrir uno de ellos, se abre una nueva ventana. La barra de título de la ventana indica el tipo de painter abierto. Se puede abrir más de un painter al mismo tiempo. También se puede tener más de una copia del mismo painter al mismo tiempo.

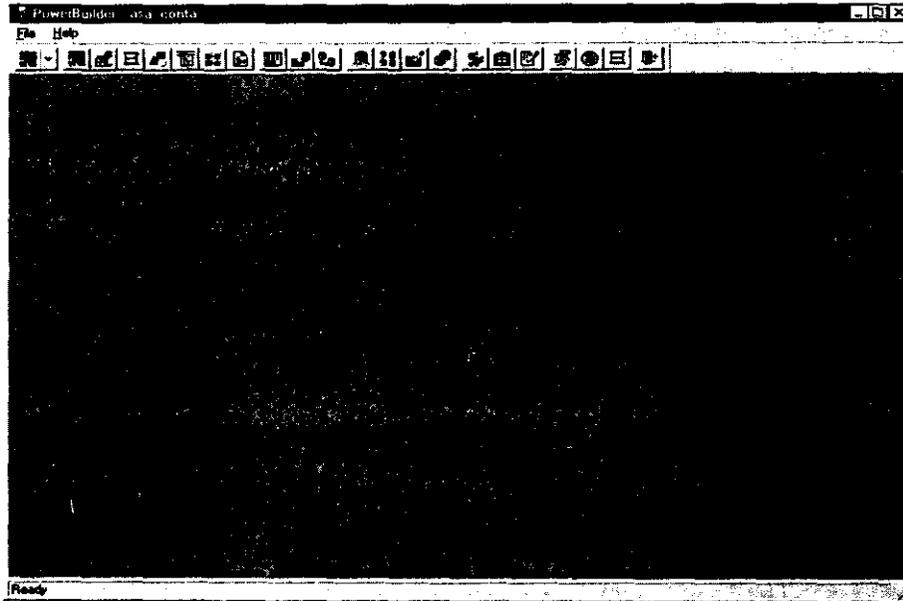


Figura 2.6.1 ventana de trabajo de Power Builder

#### **El application Painter.**

El marco general para cualquier cosa que se construya con Power Builder es la aplicación. La aplicación que se construya puede contener muchos objetos (ventanas, menús, gráficos o informes). Todos esos objetos se encuentran dentro de una aplicación.

Haciendo clic sobre el icono aplicación del Power Panel se presenta la pantalla mostrada en la figura 2.6.2. El application painter (diseño de aplicación) selecciona una aplicación Power Builder que ya existe o crea una nueva aplicación.

En la esquina superior izquierda de la ventana del application painter se presenta una serie de menús. Por medio de éstos, se puede acceder a las diferentes funciones del application painter.

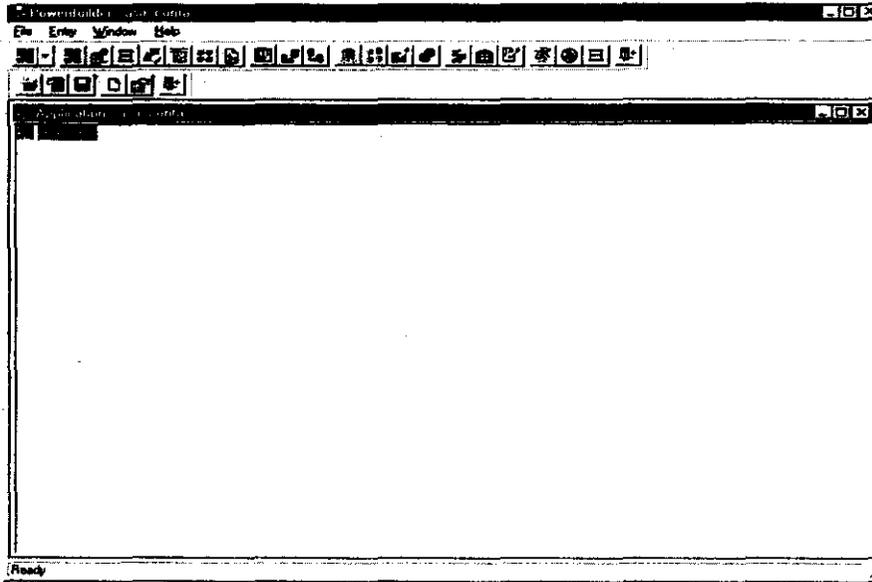


Figura 2.6.2 Application Painter

#### **El Database Painter.**

El database painter suministra facilidades para el mantenimiento de la base de datos tales como la definición y el acceso de datos. Con el database painter se puede crear una nueva base de datos o eliminar una ya existente, añadir o borrar tablas de una base de datos o cambiar su formato. También se pueden cambiar autorizaciones y añadir índices. Finalmente, se puede utilizar el painter para añadir o cambiar los datos de la base de datos.

Aunque el database painter es de gran utilidad, no soporta algunas operaciones. Las herramientas suministradas con ciertos gestores de bases de datos permiten cambiar el formato de una columna de una tabla, mientras que el database painter no lo permite. Si hay alguna operación que se desea realizar y que no es posible hacerla con el database painter, se debería ver si dicha facilidad es suministrada por las herramientas propias del gestor de la base de datos.

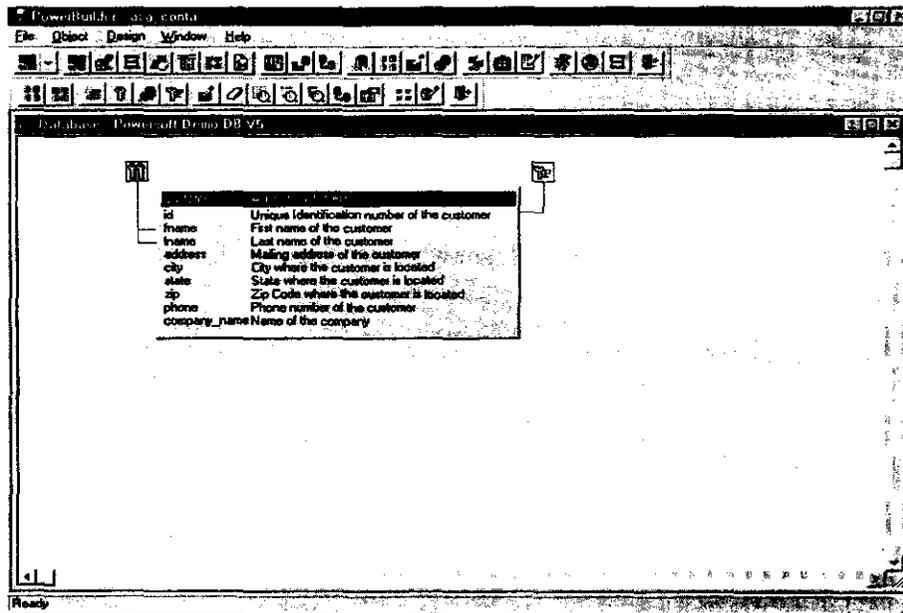


Figura 2.6.3 Database Painter

### El Library Painter.

Una aplicación Power Builder normalmente incluye muchos objetos. Los objetos tales como ventanas o controles residen en una librería Power Builder. Una librería Power Builder es un archivo que tiene un nombre con extensión .pbl (denominado pibble). Todos los objetos de una aplicación pueden residir en una única librería o en múltiples. El usuario es el que decide colocar un objeto Power Builder en una librería u otra.

El library painter suministra todas las facilidades necesarias para la gestión de las librerías y de los objetos que hay en ellas. El library painter permite crear una nueva librería o eliminar una vieja. También tiene órdenes para copiar o mover objetos entre librerías.

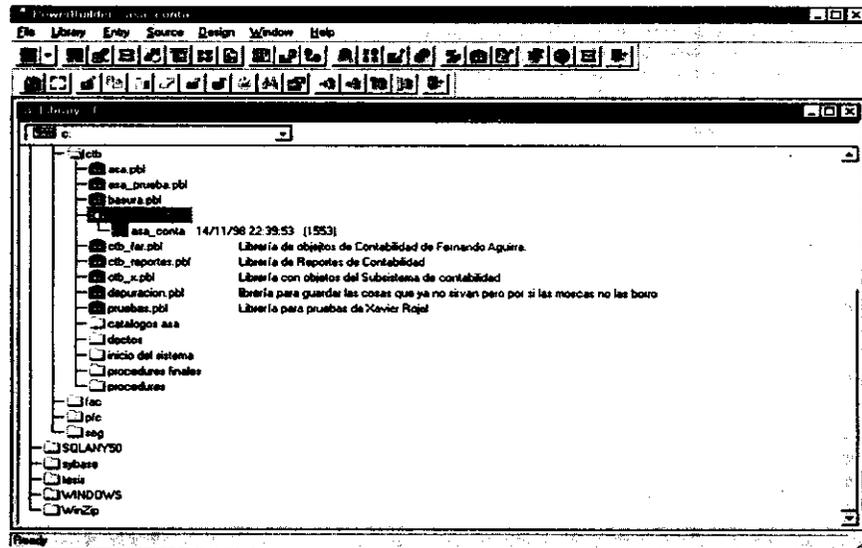


Figura 2.6.4 Library painter

El library painter también tiene dos clasificadores. Un clasificador es una herramienta utilizada para la búsqueda de objetos. En el menú de Utilities (utilidades) se puede seleccionar el clasificador de jerarquía de clase de objetos.

#### El Window Painter.

Con el window painter se puede construir cualquiera tipo de ventana. Las ventanas son bloques de construcción básicos de una aplicación. Otros objetos tales como controles, menús o gráficos aparecen sobre una ventana. La figura 2.6.5 muestra una pantalla en la cual se está diseñando una ventana.

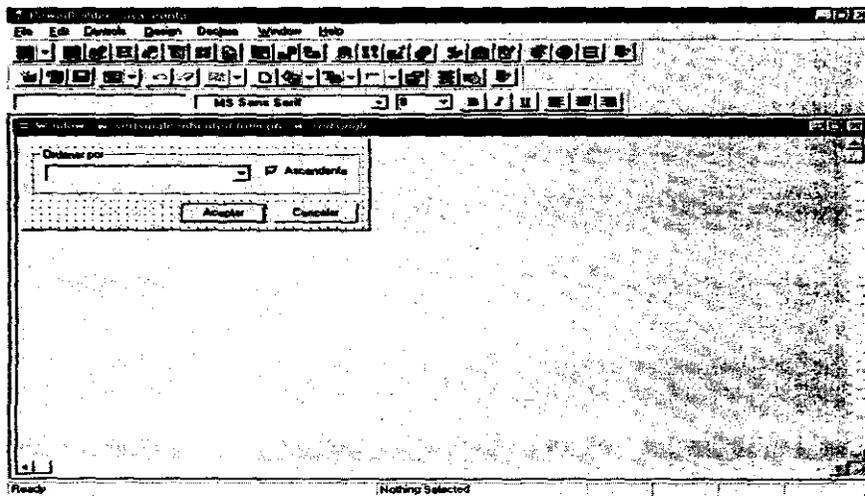


Figura 2.6.5 Window Painter

Las data windows son objetos "SQL smart" que sólo se encontrarán en Power Builder. Cuando se construye un objeto data window con el data window painter, se generan automáticamente las ordenes SQL para el control de la ventana. La data window también permite dar formato a los datos para presentarlos en informes.

Ninguna otra herramienta de desarrollo cliente – servidor tiene la capacidad del data window Power Builder.

### El Menu Painter.

Un menú es un objeto que se puede añadir a una ventana en una aplicación Power Builder. Los menús que se pueden construir con el menu painter se parecen a los menús familiares con cualquier aplicación windows (figura 2.6.6).

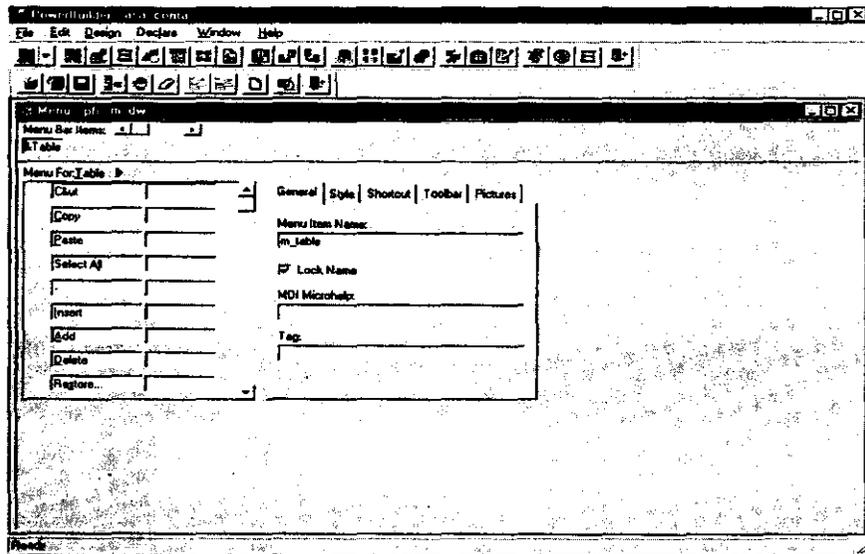


Figura 2.6.6 Menu Painter.



### El Query Painter.

SQL (structured Query Language) es un lenguaje estándar industrial que permite el acceso o modificación de los datos contenidos en una base de datos. Los operadores SQL seleccionan o manipulan datos de una base de datos situada en un servidor. En algunos casos, como cuando se construye una data window, Power Builder crea las secuencias SQL correspondientes. En otros casos, éstas deben ser escritas por quien está realizando la aplicación. Power Builder define diccionarios que asocian palabras inglesas con elementos de la base de datos, lo que facilita las consultas (figura 2.6.8).

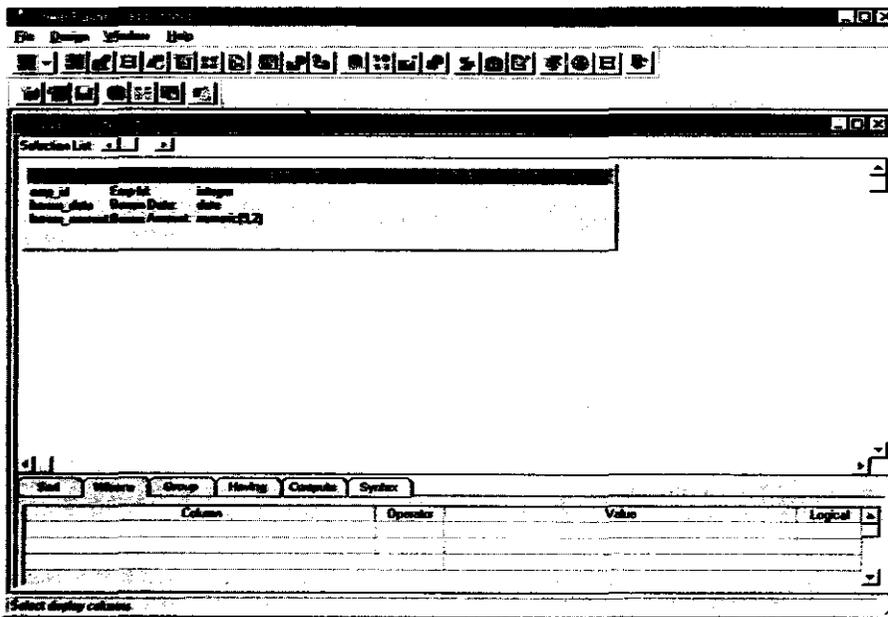


Figura 2.6.8 Query Painter

### El User Object Painter.

Power Builder contiene una serie de objetos predefinidos, tales como ventanas, menús o controles. Se pueden crear objetos a medida con el user object painter. Se puede crear un objeto que contenga una ventana, una data window, algunos controles y un menú (figura 2.6.9).

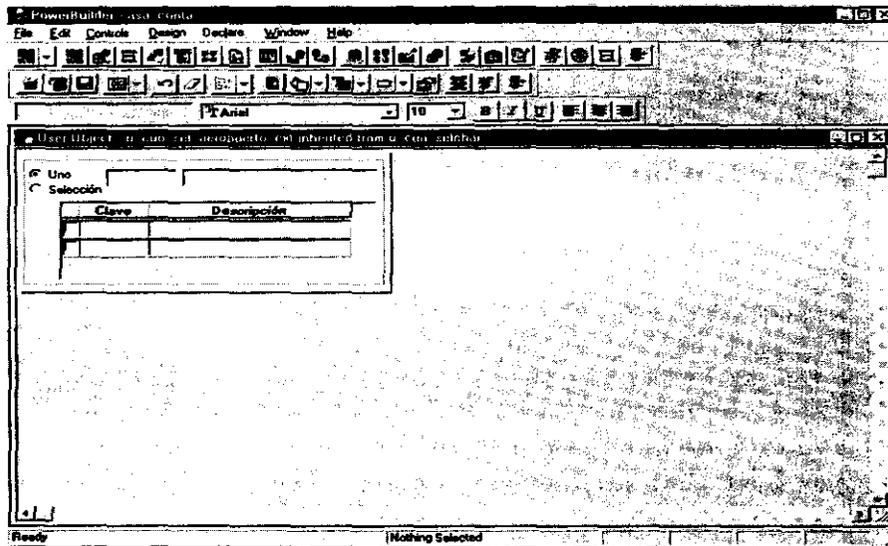


Figura 2.6.9 User Object Painter

### El Scrip Painter.

La mayoría de los objetos Power Builder tienen asociados eventos. Por ejemplo, una aplicación Power Builder tiene los eventos open, close, system error e idle asociados a ella. Un procedimiento Power Builder (el programa que se ejecuta cuando ocurre el evento) puede corresponder con cualquier evento. El script painter se utiliza para escribir estos procedimientos.

En la mayoría de las aplicaciones, se escriben pocos y cortos procedimientos debido a la potencia del modelo de Powersoft. Los procedimientos van asociados a los eventos, aunque normalmente no todos los eventos los llevan asociados. Los procedimientos tienden a ser cortos debido a la simplicidad del lenguaje y a la capacidad de la librería de funciones predefinidas (figura 2.6.10).

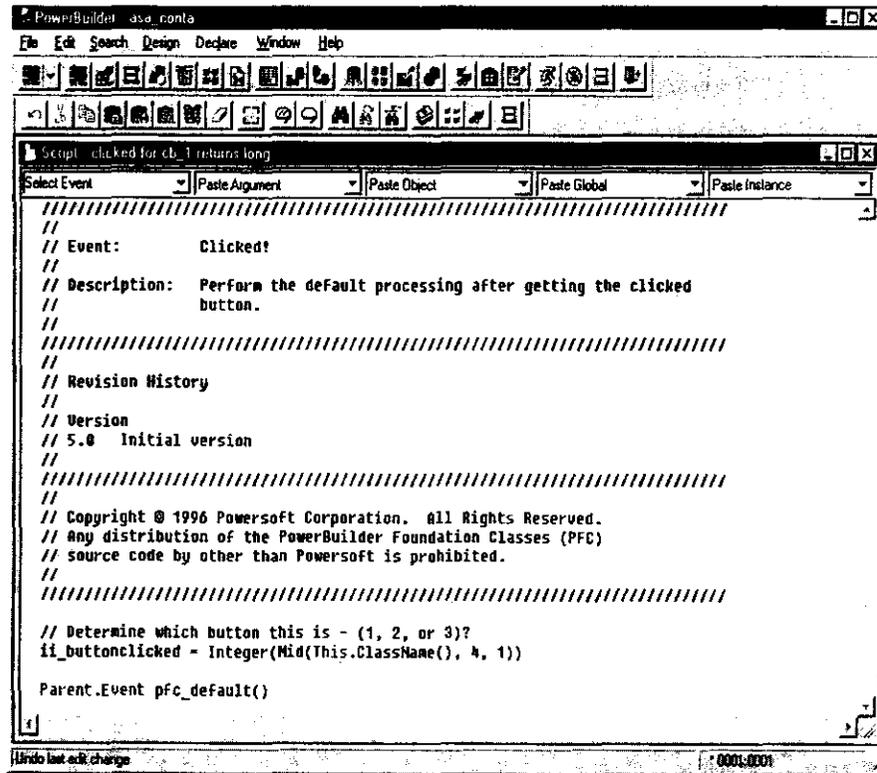


Figura 1.1.10 Scrip Painter

## El Depurador.

Power Builder incluye un depurador interactivo. Con él se pueden establecer puntos de ruptura en diferentes partes de la ejecución de una aplicación y así poder examinar cómo cambian los valores de las variables. Este depurador suministra todas las facilidades necesarias para encontrar los problemas que puede tener un programa al ejecutarse (figura 2.6.11).

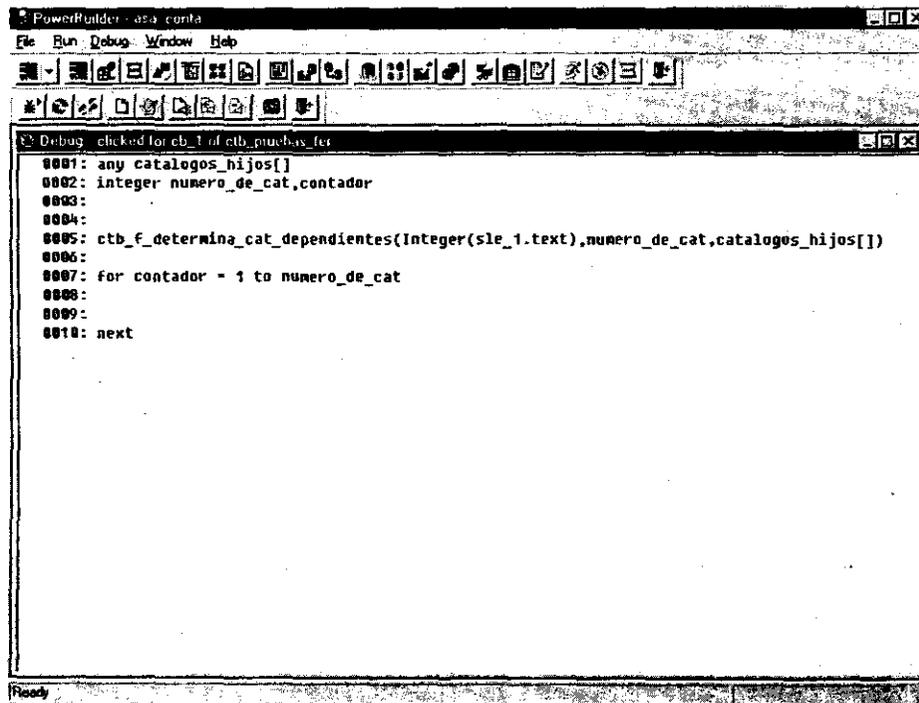


Figura 2.6.11 El Depurador.

## **Tipos de Datos y Declaraciones.**

### **Identificadores.**

Los identificadores como su nombre lo indica identifican objetos tales como variables, etiquetas, funciones, ventanas, controles y menús. Un identificador debe comenzar con una letra y utilizar como máximo cuarenta caracteres y:

- No debe contener espacios en blanco.
- Puede incluir algunos caracteres no alfabéticos, como:
  - Guión
  - \_ Subrayado
  - \$ Signo de Dólar
  - # Signo de número
  - % Signo de tantos por ciento

Los identificadores no diferencian las mayúsculas. El identificador Subtotal es equivalente al identificador subtotal y al SubTotal. Aunque se pueden utilizar caracteres no alfabéticos tales como guión o un subrayado, no es conveniente su uso.

### **Palabras Reservadas.**

Existen ciertas palabras que no pueden ser utilizadas como identificadores ya que Power Builder las utiliza con una determinada finalidad. A continuación se presenta una lista de palabras reservadas.

and	enumerated	library	select
call	event	loop	selectblob
case	execute	next	shared
choose	exit	not	step
close	external	of	subrutine
commit	false	on	super
connect	fetch	open	system
continue	first	or	then
create	for	parent	this
cursor	forward	prepare	to
declare	from	prior	tur
deleted	funtion	private	type
describe	global	procedure	until
descriptor	toto	protected	update
destroy	halt	prototypes	updateable
disconnect	if	public	using
do	immediate	readonly	variables
dynamic	insert	ref	while
else	into	return	with
elseif	is	rollback	within
end	last		

#### **Tipos de Datos Estándar.**

Ciertos tipos de datos son intrínsecos al lenguaje PowerScrip, pero la mayoría son tipos de datos que se encuentran en otros lenguajes de programación. La siguiente tabla muestra los tipos de datos estándar.

<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
Blob	Binary Large Object. Datos sin formato, por ejemplo una imagen o un bloque de texto.	PowerObject	Puede contener cualquier objeto powerbuilder, incluyendo estructuras.
Boolean	True o False	Real	Número en coma flotante con una precisión de hasta 6 dígitos, su rango va desde 1.18E-38 hasta 3.40 E+38
Character (Char)	Un único carácter	String	Contiene desde 0 a 60000 caracteres ascii
Date	Tipo fecha	Time	Tipo hora
DateTime	Combinación de una fecha y una hora	UnsignedInteger	Enteros sin signo de 16 bits.
Decimal	Números decimales con signo hasta una longitud de 18 dígitos. El punto decimal no se cuenta como un dígito	UnsignedInt	Rango desde 0 hasta 65535. Formato abreviado uint.
Double	Números en coma flotante tiene hasta 15 dígitos de precisión y su rango va desde 2.23E-308 hasta 1.79E+308	UnsignedLong	Entero sin signo de 32 bits
DragObject	Puede contener cualquier tipo de objeto dragged incluyendo todos los controles, pero no objetos dibujo.	Integer	Enteros con signo de 16 bits. El rango va desde -32768 hasta +32767
Object	Contiene un tipo enumerado de un objeto PowerBuilder. Puede incluir todas las ventanas y controles.	Long	Enteros con signo de 32 bits el rango va desde -2147483648 hasta +2147483647

### **Tipos de Datos Enumerados.**

Power Builder tiene muchos tipos de datos enumerados. Estos tipos de datos especifican los atributos de un objeto. Están predefinidos y no hay que declararlos.

Por ejemplo el objeto ventana tiene un atributo denominado pointer, el valor de este atributo determina la forma del puntero cuando éste se encuentra en algún lugar de la ventana. El puntero puede tener diversas formas.

A continuación se da una lista de tipos enumerados del atributo pointer, mostrando las diferentes formas que puede tomar un puntero. Estos tipos de datos siempre tienen una exclamación al final, la cual indica al compilador de procedimiento que es un dato enumerado:

Arrow!  
Beam!  
Cross!  
HourGlass!  
SizeNESW!  
SizeNS!  
SizeNWSE!  
SizeWE!  
UpArrow!

El compilador checa los tipos de datos enumerados cuando compila un procedimiento. Esto implica que un tipo de datos enumerado incorrecto será detectado al compilarlo.

### **Alcance de las Variables**

El alcance de una variable determina desde qué parte de una aplicación puede ser utilizada. La manera de declarar la variable determina su alcance. Este puede ser:

- Global
- Instance
- Shared
- Local

#### **Variables Globales.**

Una variable global es accesible desde cualquier parte de una aplicación Power Builder. Se puede acceder a una variable global desde cualquier procedimiento de una aplicación. Se puede declarar seleccionando la opción Global Variables desde los window painters, objetos de usuarios, menús o procedimientos. Es necesario resaltar que una aplicación correctamente escrita tendrá un mínimo de variables globales, si tiene alguna.

#### **Variables Instancia.**

Las variables instancia siempre se corresponden a una instancia de un objeto. Se podría querer asociar una variable con un objeto ventana. Por ejemplo, se podría crear una variable instancia de una ventana para acceder al número de pedido o al número de entrega. Las variables instancia pueden existir para los siguiente objetos:

- Ventana
- Aplicación
- Objeto de usuario
- Menú

Las variables instancia pueden tener uno de los tres siguientes niveles de acceso: public, private y protected.

**Public.** Cualquier procedimiento de una aplicación puede acceder a una variable de instancia pública.

**Private.** Las variables instancia se definen para un solo objeto. Sólo desde el procedimiento escrito para ese objeto se puede acceder a una variable de instancia privada.

**Protected.** Se puede acceder a una variable de instancia protegida desde cualquier procedimiento de un determinado objeto o desde cualquier procedimiento de un objeto descendiente.

#### **Variables Locales.**

Se pueden declarar variables en cualquier procedimiento. Sólo en el procedimiento se pueden referenciar las variables locales que se definan. Las variables locales guardan valores temporales.

#### **Variables no clasificadas.**

Se puede clasificar una variable utilizada en un procedimiento. Para un objeto denominado *x*, el siguiente procedimiento selecciona el atributo *a* para el objeto *x*.

Ejemplo: `x.a = x.a+100`

Si existe una variable no clasificada en un procedimiento, el compilador debe descubrir a qué variable se está refiriendo. Power Builder resuelve el problema buscando en los diferentes tipos de variables llevando el siguiente orden:

- Variables local
- Variables shared

- Variables global
- Variables instance

#### **Acceso a una base de Datos.**

Una aplicación Power Builder en ejecución puede establecer una conexión con una base de datos. Para establecer esta conexión se debe escribir un procedimiento. Este procedimiento se asocia normalmente al evento abrir de la aplicación. También se puede conectar con una base de datos desde dentro del entorno de desarrollo de Power Builder. La conexión ocurre cuando se crea un objeto Data Window, o cuando se utiliza el Database Painter.

Para conectarse a una base de datos desde el entorno de desarrollo, primero se deben establecer las preferencias para esa conexión. Hay tres maneras de modificar estas preferencias. En primer lugar, se puede editar el archivo pb.ini, el archivo perfil de Power Builder. Esto no es una buena idea, ya que requiere parar y arrancar Power Builder para que los cambios tengan efecto. En segundo lugar, se puede llamar a la ventana de perfiles de base de dato y en tercer lugar, se pueden cambiar las preferencias con el Preference Painter.

Los valores de las preferencias de la base de datos se utilizan al establecer una conexión a una base de datos. Estos valores se usan cada vez que se conecte a una base de datos con el Data Window Painter o el Database Painter.

Las preferencias de la base de datos identifican la conexión por defecto a la base de datos dentro del entorno de desarrollo. Los valores de cada una de las preferencias varían según los sistemas de bases de datos. Las preferencias para una conexión a una base de datos Oracle se diferencian enormemente de las de una conexión a una base de datos Sybase o DB2.

### Transacciones.

Un sistema de gestión de base de datos relacional es responsable de soportar la integridad de los datos. Un mecanismo proporcionado para soportar la integridad de los datos es la transacción o unidad lógica de trabajo.

Las transacciones se soportan con el lenguaje PowerScript. Los procedimientos pueden utilizar las transacciones para permitir la recuperación ante una transacción incompleta o incorrecta. Pueden usar las transacciones para soportar control de concurrencia. También pueden maximizar el rendimiento utilizando las transacciones para minimizar el número de conexiones y desconexiones.

Power Builder soporta muchos tipos de objetos incluyendo ventanas, ventanas de datos y controles. Uno de estos tipos de datos predefinidos es el objeto transacción. Las aplicaciones Power Builder usan el objeto transacción para gestionar la conexión a la base de datos (figura 2.6.12).

El objeto transacción especifica la información de la base de datos, incluyendo el nombre de la base de datos y el identificador de usuario. El objeto transacción también devuelve a la aplicación en ejecución información del estado del sistema de gestión de base de datos. Las siguientes oraciones pueden aceptar un nombre de objeto transacción:

- Connect
- Commit
- Delete
- Disconnect
- Insert
- Rollback
- Update

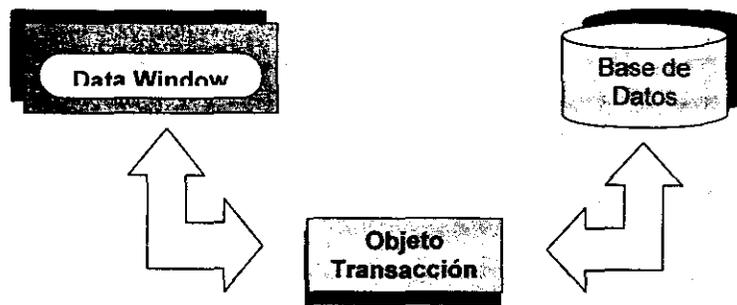


Figura 2.6.12 Objeto Transacción

Un objeto transacción es un objeto no-visual. Una aplicación puede crear y utilizar múltiples objetos transacción, aunque uno puede ser suficiente. Un solo objeto transacción realiza una sola conexión a una base de datos. Conexiones múltiples requieren múltiples objetos transacción.

Toda aplicación Power Builder puede acceder a un objeto transacción por defecto llamado SQLCA. La creación de un objeto aplicación crea automáticamente este objeto transacción.

#### **Ventajas de Power Builder.**

Con Power Builder se pueden realizar aplicaciones que pueden ser ejecutadas en entornos diferentes al de Microsoft Windows, como por ejemplo: HP/UX de Hewlett Packard, AIX de IBM, Solaris de Sun Microsystems, Macintosh System 7, Microsoft Windows NT y OS/2 de IBM. Se puede realizar una aplicación en un entorno y pasarla a cualquier otro, simplemente recompilandola en el nuevo entorno.

Power Builder incluye funcionalidades para los objetos, entre las cuales se encuentran las que permiten crear o ampliar las claves no visuales. También se encuentran las de acceso a objetos y clases externas, incluyendo herramientas DLL (Microsoft Dynamic Link Library).

Soporta la anidación de informes. Se puede incluir un objeto DataWindow dentro de otro DataWindow. Esto es muy útil en gran cantidad de aplicaciones, por ejemplo, para crear cabeceras de informes, secciones detalladas o para la realización de diversos formatos en un único informe.

Las tuberías de datos permiten conectar una fuente de datos con un destino simplemente definiendo la conexión. Pueden moverse decenas de miles de filas por minuto entre aplicaciones a través de estas tuberías de datos. También cuenta con el soporte para los procedimientos de almacenamiento de Oracle 7.

Power Builder soporta la arquitectura OLE (Object Embedding and Linking) de Microsoft. También incluye la nueva versión OMG (Object Management Group) y CORBA (Common Object Request Broker), que se utilizan para la especificación de objetos.

Cuenta con la tecnología de compilación Watcom. Incluye el generador de clases de objetos C++ Watcom. Con dicho generador se pueden construir librerías compatibles con C y C++.

Es posible gestionar diversas aplicaciones y desarrollos en paralelo.

Las versiones de las aplicaciones incluyen etiquetas de todos los objetos PowerBuilder. Una vez que los objetos se etiquetan es fácil realizar su seguimiento o devolverlo a cualquier versión anterior.



### 3.1 SITUACIÓN ACTUAL.

El Sistema Contable que se pretende realizar será utilizado en la Red Aeroportuaria Nacional, la cual está conformada por todos los aeropuertos del país. A continuación se explicará de que forma se lleva la contabilidad actualmente.

La Red Aeroportuaria Mexicana esta conformada con 61 aeropuertos, estos están agrupados por áreas geográficas, esas áreas las cuales son conocidas como regionales, son las siguientes:

➤ **SURESTE.** Conformada por los aeropuertos:

- a) *Cancún*
- b) *Cozumel*
- c) *Mérida*
- d) *Minatitlán*
- e) *Oaxaca*
- f) *Tapachula*
- g) *Veracruz*
- h) *Villahermosa*
- i) *Huatulco*

➤ **PACÍFICO.** Conformada por los aeropuertos:

- a) *Aguascalientes*
- b) *Guadalajara*
- c) *Hermosillo*
- d) *La Paz*
- e) *Los Mochis*
- f) *Manzanillo*
- g) *Mexicali*
- h) *Morelia*
- i) *Puerto Vallarta*

- j) San José del Cabo
- k) Tijuana
- l) El Bajío

➤ *SUR*. Conformada por los aeropuertos:

- a) Campeche
- b) CD. Del Carmen
- c) Chetumal
- d) Poza Rica
- e) Puerto Escondido
- f) Tehuacán
- g) Tuxtla Gutierrez
- h) Toluca
- i) Puebla
- j) Lázaro Cárdenas
- k) Pachuca
- l) Cuernavaca
- m) Palenque
- n) San Cristóbal

➤ *NORTE*. Conformada por los aeropuertos:

- a) CD. Obregón
- b) CD. Victoria
- c) Guaymas
- d) Loreto
- e) Matamoros
- f) Nogales
- g) Nuevo Laredo
- h) Tamuín
- i) Tepic
- j) Uruapan

- k) Colima
- l) Queretaro

➤ *A.I.C.M.* Conformada por el aeropuerto:

- a) *Internacional de La Ciudad de México*

Además se cuenta con oficinas en el D.F. las cuales son conocidas como las Oficinas Generales de la Red Aeroportuaria y físicamente se encuentran dentro del área geográfica del aeropuerto de la Ciudad de México.

Cada uno de los aeropuertos puede tener varias clasificaciones. La primer gran clasificación de los aeropuertos se hace atendiendo a la infraestructura en materia aeroportuaria con la que cuentan, ya que dicha infraestructura determina si se puede o no otorgar un servicio a aeronaves que hagan recorridos solo en el territorio nacional o que hagan recorridos alrededor del mundo, por esto podemos encontrar aeropuertos nacionales e internacionales. Los primeros sólo están capacitados para atender vuelos nacionales tanto comerciales como turísticos y los segundos para atender vuelos nacionales e internacionales tanto comerciales como turísticos.

Otra de las clasificaciones de los aeropuertos es dependiendo del área geográfica donde se encuentren y podemos encontrar aeropuertos fronterizos y costeros.

En la figura 3.1.1 se muestran los aeropuertos del país su ubicación física y la regional a la que pertenece.



Figura 3.1.1 Red Aeroportuaria Nacional

Dentro de cada uno de los aeropuertos se llevan a cabo infinidad de operaciones que generan recursos monetarios para la red aeroportuaria, y que se registran y controlan por medio de una contabilidad a nivel aeropuerto y una contabilidad a nivel nacional, la contabilidad a nivel a aeropuerto la realiza el personal del aeropuerto y la contabilidad a nivel nacional se hace en las Oficinas Generales en la Ciudad de México.

A continuación se explicarán brevemente algunas de las operaciones más importantes que se realizan en los aeropuertos.

Cada uno de los aeropuertos proporciona un gran número de servicios a las aeronaves que llegan a sus instalaciones y a los pasajeros que estas transportan.

Existen áreas dentro de los aeropuertos que proporcionan los servicios aeroportuarios de aterrizaje, estacionamiento, pernocta, uso de abordadores mecánicos, suministro y succión de combustibles y recepción de manifiestos de llegada y de salida.

En todos los aeropuertos existe un área llamada Centro de Control de Operaciones desde la cual se controla la asignación de posiciones de estacionamiento y pernocta, la planeación de asignación de posiciones se hace con base en los itinerarios de vuelos que presentan las líneas aéreas, la capacidad del aeropuerto en lo que se refiere a cantidad de posiciones existentes y a las reglas que indican las políticas de asignación, mismas que se rigen por normas internacionales y por los acuerdos tomados con las líneas aéreas en el Comité de Horarios del aeropuerto. Un aeropuerto puede ser visto como un estacionamiento, en el cual además de brindar el servicio de una estancia para una aeronave, le ofrece otros servicios.

El área Centro de Control de Operaciones también se encarga del control de llegadas y salidas de las aeronaves, vuelos no planeados, cambios de vuelo,

cambios de posición, cancelación de vuelos, etc. todo esto conforma el movimiento operacional.

El área de Abordadores se encarga de atender con algún servicio de abordador (aerocar, sala móvil o pasillo telescópico) a una aeronave que se encuentra en el aeropuerto. El servicio de Abordadores se utiliza para facilitar el acceso a los pasajeros a las aeronaves o para facilitar la carga y descarga de las mercancías que transporta la aeronave.

Una de las áreas de mayor importancia dentro de los aeropuertos es sin duda la de Combustibles en donde se controla el inventario del combustible a nivel general del aeropuerto y por equipo de almacenamiento (tanques o autotanques). Se controla el registro de las entradas y salidas de combustible del almacén y de los equipos (recepción de PEMEX, suministros, succiones, traspaso entre equipos, etc.), así como el registro del estado de los medidores al cierre del día. Cabe recalcar que una aeronave de cualquier tipo ya sea nacional o internacional sólo podrá abastecerse de combustible en cualquiera de las instalaciones de cualquiera de los aeropuertos de la red aeroportuaria ya que en ningún otro lugar se puede adquirir el combustible que utilizan los aviones y que se llama turbosina. PEMEX fabrica la turbosina, y se la vende a la Red Aeroportuaria, nadie más en el país puede comprar turbosina directamente con PEMEX. Este Combustible es llevado a cada uno de los aeropuertos principalmente por vía terrestre (se utilizan camiones especiales de PEMEX). El aeropuerto de la Ciudad de México se surte por medio de un ducto subterráneo. Una vez que la turbosina se encuentra en los aeropuertos se procede a suministrarla a las aeronaves que así lo requieran. Otro de los servicios que ofrecen los aeropuertos es el de succiones, esto se refiere a quitar el combustible de la aeronave, esto se da principalmente por exceso de combustible, sobrepeso de la nave, etc.

La Red Aeroportuaria cuenta con convenios con un gran número de líneas aéreas, tanto nacionales como internacionales, estas líneas aéreas al momento de

firmar un convenio con la Red, se comprometen a entregar a cada uno de los aeropuertos de los cuales vayan a hacer uso, la información de los manifiestos de llegada y salida de sus aeronaves. Con estos manifiestos el aeropuerto puede hacer un itinerario de llegadas y salidas y un uso óptimo de sus instalaciones.

Un cliente de la Red Aeroportuaria puede ser clasificado como cliente de contado o cliente de crédito, los clientes que se encuentren dentro de la primera clasificación deberán de pagar en el momento que se les brinde el servicio, y los que se encuentren en la segunda clasificación, sólo firmarán un contrato en el cual se comprometen a pagar el monto arrojado por el servicio en una fecha posterior, esto se da principalmente con las líneas aéreas, las cuales, firman convenios con la Red Aeroportuaria, a fin de que a sus aeronaves se les brinde el servicio que requieran y posteriormente se hará el pago correspondiente.

Hasta el momento se han explicado las operaciones más importantes que generan movimientos contables, existen otros servicios que brindan los aeropuertos y que también se registran.

Un aeropuerto como cualquier empresa también tiene además de ingresos monetarios, egresos por diferentes conceptos, tales como mantenimiento de edificios y de pistas, limpieza de instalaciones, pago de personal, etc.

Debido a que la situación de todos los aeropuertos no es homogénea en cuestión de personal con conocimientos contables, podemos encontrar que en algunos la contabilidad no se lleva a cabo, en otros se hace un procesamiento previo de la información contable, y en el mejor de los casos sí realizan totalmente la contabilidad y se generan todos los reportes contables básicos.

Es en la Gerencia de Contabilidad de las Oficinas Generales en donde se concentra toda la información contable de los aeropuertos, ya que es una de las funciones de esta, el generar todos los reportes contables básicos (libros diarios,

libros de mayor, libros auxiliares, estado de resultados, estado de situación financiera) de cada uno de los aeropuertos, de las regionales y de la Red Aeroportuaria. Para poder lograr este objetivo se han implementado algunos procedimientos operativos, a fin de poder contar con toda la información necesaria para poder emitir los reportes antes mencionados.

Como explicamos anteriormente la situación en materia contable de los aeropuertos no es homogénea, debido a esto la Gerencia de Contabilidad de las Oficinas Generales puso en marcha varios procedimientos de trabajo, a fin de poder contar con la información necesaria para emitir los reportes contables.

Algunos aeropuertos que realizan su contabilidad, tienen demasiadas fallas en el procesamiento de la información, en muchos casos la manipulación de la información es hecha por personal que no tiene el conocimiento contable adecuado, se da el caso de que la contabilidad la realizan empleados del aeropuerto que no son contadores.

La Gerencia de contabilidad pide a todos los aeropuertos toda la información de todas sus operaciones, esta información puede ser muy variada y entre los casos más frecuentes e importantes por su cantidad y los montos que se manejan podemos encontrar:

- Facturas
- Notas de recepción de combustible
- Notas de crédito
- Remisiones
- Contratos varios
- Contratos de arrendamientos
- Notas de servicios
- Pagares
- Inventario de almacenes

- Estados de cuentas bancarias
- Listas de precios de combustibles
- Copias fotostáticas de cheques emitidos recibidos
- Comprobantes de gastos de viáticos
- Nóminas de empleados
- Pagos a personal contratado eventualmente
- Retenciones de impuestos

Todos los aeropuertos tienen la obligación de mandar toda esta información aun en el caso de que ellos manejen total o parcialmente su contabilidad. Los medios por los cuales llega esta información son por medio de mensajería, o vía fax.

Los aeropuertos que procesan de alguna forma la información contable que ellos generan lo hacen de diversas maneras, y auxiliándose de diversas herramientas. En la Mayoría de los casos los aeropuertos cuentan con un sistema de contabilidad comercial llamado COI®, en el cual registran las operaciones contables y emiten sus reportes. Otros aeropuertos llevan la contabilidad de forma totalmente manual auxiliados por hojas de cálculo.

La información requerida en Oficinas Generales, es la concentración de cada mes, y se ha intentado de que se respeten formatos establecidos por la Gerencia de Contabilidad. La información debe de recibirse a mas tardar dentro de los tres primeros días siguientes a cada fin de mes. Por lo que es conveniente señalar que cada aeropuerto establece una fecha de corte mensual, a fin de poder cumplir con la fecha de entrega de la información, cabe destacar que en los cierres anuales, la información que se remita deberá contener las operaciones realizadas hasta el 31 de diciembre.

A continuación se muestran los formatos de los reportes que las Oficinas Generales piden a los aeropuertos.

➤ Balanza de Comprobación

El primer reporte que se pide es el de la balanza de comprobación el formato se muestra en la figura 3.1.2, los números coinciden con los del reporte.

1. Número de hoja.
2. Fecha de elaboración
3. Aeropuerto
4. Nombre de la persona que elaboró el documento
5. Nombre de la persona que revisó el documento
6. Nombre de la persona que autorizó el documento
7. Cuenta contable en donde se hace el movimiento, atendiendo al catálogo vigente
8. Subcuenta se anotará el número de la subcuenta en donde se hará el movimiento atendiendo al catálogo vigente
9. Saldo de la cuenta de naturaleza deudora con que finalizó en la última balanza reportada
10. Saldo de la cuenta de naturaleza acreedora con que finalizó en la última balanza reportada.
11. Periodo que ampara la balanza.
12. Movimientos al debe de la cuenta contable
13. Movimientos al haber de la cuenta contable
14. Resultado aritmético del saldo con naturaleza deudora
15. Resultado aritmético del saldo con naturaleza acreedora

Las sumas de las columnas de los números 12 y 13 así como de las columnas 14 y 15 siempre deben de ser iguales.



➤ *Póliza de Registro Contable*

Otro de los formatos que se han hecho es el de la póliza de registro contable, el cual se utiliza para contabilizar los ingresos, egresos y los movimientos de diario. La figura 3.1.3 nos muestra el formato de la póliza de registro, y los siguientes números coinciden con los de la figura.

1. Fecha de la elaboración de la póliza con formato mes, día, año.
2. Número de póliza consecutivo que le corresponda el cual iniciará cada mes.
3. Número de hoja
4. Número de aeropuerto que corresponda
5. Cuenta contable
6. Subcuenta contable
7. En caso de que se trate de un egreso se debe de anotar el número de cheque o referencia de la operación
8. Concepto de la póliza
9. Importe del cargo
10. Importe del abono
11. Nombre y firma de la persona que elaboró la póliza
12. Nombre y firma de la persona que registra la póliza en los libros contables
13. Nombre y firma de la persona que revisó la póliza
14. Suma de los movimientos deudores y acreedores en la última hoja de cada póliza.



➤ *Libro tabular de ingresos y egresos*

Para el registro de las operaciones relacionadas con el movimiento de fondos de bancos se debe de llevar un libro tabular de ingresos y egresos, en el cual como su nombre lo indica, únicamente se registran las pólizas de ingresos y egresos, haciéndose aplicaciones a nivel de cuenta.

Los datos que debe de contener son los siguientes:

- Fecha del registro de la operación
- Número progresivo de la póliza
- Nombre del beneficiario del cheque o concepto de ingreso
- Número de cheque en forma progresiva
- Las primeras dos columnas de debe y haber se utilizan para la cuenta 101 bancos. Las restantes se utilizan con los números y nombres de las cuentas más usuales y se asienta primero las de naturaleza deudora y posteriormente las de naturaleza acreedora. Las cuatro columnas finales se utilizan para varias cuentas anotando en las dos primeras el debe y el haber y en las siguientes, el número y el nombre de la cuenta.

En la figura 3.1.4 se pueden observar las partes que componen este formato.

ABRILMENTO X  
LIBRO DE INGRESOS Y EGRESOS DEL MES DE DICIEMBRE DE 1994

FECHA	REF.	CONCEPTO	BANCOS		FUNCIONARIOS Y EMP.		IMPUESTOS Y CUOTAS Y PAGOS REMESAS		GASTOS Y CUOTAS Y PAGOS REMESAS		BASTOS DE OPERACION		IMPUESTOS CUENTAS		CUBIERTA	CONCEPTO
			DEBE	HABER	DEBE	HABER	DEBE	HABER	DEBE	HABER	DEBE	HABER	DEBE	HABER		
10/12/94	2002	JUAN DIEZ. PAGO DE RENTA		200												
5/12/94	2003	EMER. AMERIZACION DE GTS. DE UNION		1400												
8/12/94	2004	FCO. ORTIZ AVILA DE LEONES		160												
19/12/94	2005	FCO. ORTIZ LUCENA DE MARLEJO		90												
16/12/94	2006	REMESA LA SEMANA	2000			1800										
16/12/94	2007	ING. ALBA ANTICIPO		600												16 ANTICIPO A CONTRIBUTIVAS
19/12/94	2008	ALVARO BARRERA SANCHEZ GASTOS A COMPROMISO		500												
20/12/94	2009	PAGO POR RETIRO		950												
20/12/94	2010	REMESA LA OMA DE 94		6600												
24/12/94	2011	INQUIRIDA LA QUINCEMA DIC.		2000												
30/12/94	2012	BOLETIN PAS. GUBERN. CURSO COMPUTACION		2000												
31/12/94	2013	FERNANDO GUEZOSABAN UNIFORME		2500												
		SUMAS	6600	9740	6600	9800	0	0	200	7200	1300	0	6000	300		

Figura 3.1.4 Formato del libro tabular de ingresos y egresos.

➤ *Libro diario tabular*

Los movimientos que no se relacionan con el movimiento de fondos en bancos se registran en el libro diario tabular, estos registros son las pólizas de diario. Los registros se dan a nivel de cuenta y al fin de mes este libro contiene la concentración de movimientos del libro de ingresos y egresos. El formato de este libro contiene los siguientes datos:

- Fecha del registro de la operación
- Número progresivo de la póliza de diario
- Origen del movimiento
- Las columnas se ordenan de la misma forma que el libro de ingresos y egresos.

En la figura 3.1.5 se muestra el formato del libro diario tabular.

➤ *Libro mayor*

El formato de libro mayor que se estableció contiene el resumen de las operaciones de cada mes reflejadas en un solo movimiento y su saldo siempre debe de coincidir con la suma de los auxiliares.

Los datos que el formato contiene son:

- Fecha de registro
- El concepto de anota el mes en que se hace el registro
- El folio del libro diario de donde procede el asiento
- El importe de la concentración del mes
- El importe del saldo acumulado

En la figura 3.1.6 se muestra el formato del libro mayor.



AEROPUERTO DE "x"  
LIBRO MAYOR

CUENTA 202			NOMBRE: IMPUESTOS Y CUOTAS POR PAGAR		
FECHA	CONCEPTO	DEBE	FECHA	CONCEPTO	HABER
Nov-94	DEL LIBRO DIARIO 11	225.00	Nov-94	DEL LIBRO DIARIO FOLIO 11	225.00
Dic-94	DEL LIBRO DIARIO 12	276.00	Dic-94	DEL LIBRO DIARIO FOLIO 12	276.00

Figura 3.1.6 Formato del libro mayor.

➤ Libros auxiliares

Como se comentó anteriormente el libro de mayor contiene los registros de las operaciones en forma global o sea a nivel cuenta, esto es un panorama general.

En un auxiliar el registro es analítico, es decir por subcuenta, esto nos proporciona información más detallada de las operaciones que se realizaron.

Oficinas Generales pide que se abran tantos registros auxiliares como subcuentas tenga cada cuenta y la suma de estas deberá coincidir con las sumas del libro de mayor.

Los datos que los registros auxiliares deben de tener son los siguientes:

- Nombre y número de la cuenta y de la subcuenta
- Fecha de la póliza
- Número progresivo de la póliza
- Origen del movimiento o nombre del beneficiario
- Columnas de debe, haber y saldo.

En la figura 3.1.7 se muestra el formato de un libro auxiliar

Existen otros formatos para otro tipo de información que las Oficinas Generales requiere para el análisis contable de cada uno de los aeropuertos, esos formatos por lo regular no son utilizados, razón por la cual no se mencionarán en este trabajo.

Como podemos observar los formatos que Oficinas Generales ha establecido como estándares contables son los reportes básicos de cualquier contabilidad.

AEROPUERTO DE: \_\_\_\_\_

AREA: \_\_\_\_\_

AUXILIAR DE LA CUENTA: IMPUESTOS Y CUOTAS POR PAGAR	NÚMERO: 202
SUB-CUENTA 0016	NOMBRE DE LA CUENTA: 10% I.S.R. HONORARIOS

FECHA	NUM. PÓLIZA	CONCEPTO	DEBE	HABER	SALDO
		SALDO ANTERIOR			0.00
30/12/94	120011	CURSO DE COMPUTACIÓN		192.00	
31/12/94	120015	TRASPASO DE IMPUESTOS RETENIDOS	192.00		
		SUMAS	192.00	192.00	0.00

Figura 3.1.7 Formato del libro auxiliar

Para los aeropuertos que cuentan con el personal capacitado para el manejo de una contabilidad, se han establecido funciones que debe de llevar a cabo el personal contable del aeropuerto esas funciones son:

- Revisar que la documentación con la cual se contraigan compromisos o pasivos, cumpla con los requisitos fiscales establecidos tanto por la Red Aeroportuaria como por las Leyes Hacendarias.
- Supervisar la elaboración de las pólizas de ingresos egresos y diario.
- Registrar en los libros diario y mayor las operaciones efectuadas por el aeropuerto.
- Cuidar que se remita a las Oficinas Generales la información pertinente.
- Realizar la conciliación y depuración de sus cuentas bancarias, tanto de egresos como de inversión.
- Organizar, integrar y controlar el archivo de la documentación contable y financiera del aeropuerto.
- Verificar que se apliquen las modificaciones a las normas y disposiciones contables y fiscales emitidas tanto por la Red Aeroportuaria como de las dependencias gubernamentales.
- Recuperar con oportunidad los adeudos que tengan los empleados con el Organismo, evitando así conservar importes con antigüedades mayores a 30 días.
- Revisar y aprobar las operaciones que por concepto de intereses por las inversiones en valores sean correctos, de acuerdo con las tasas de interés vigentes así como el envío mensual de éstos a la Tesorería de las Oficinas Generales.
- Realizar las conciliaciones mensuales de la cuenta del Almacén de Materiales entre los saldos de contabilidad y Almacén evitando así diferencias entre ambos.
- Calcular y deducir de la facturación que presenten los proveedores los descuentos por pronto pago que negocie la administración del Aeropuerto.

- Elaborar el calendario de pagos a proveedores y la relación de pasivos pendientes de pago.
- Organizar y supervisar el levantamiento del inventario físico de materiales y refacciones al cierre de cada ejercicio o cuando sea necesario.

Un punto importante que actualmente se toma en cuenta para el manejo de la contabilidad son los canales de comunicación entre las Oficinas Generales y los aeropuertos a fin de que se pueda llevar a cabo una correcta aplicación de la normatividad en el uso de los sistemas establecidos para el manejo de la información contable, controlando así la adecuada aplicación de los registros y detectando los errores que se presenten en los mismos, efectuando con oportunidad las correcciones y aclaraciones que procedan a fin de optimizar los procedimientos vigentes. Para lograr lo anterior se han establecido sistemas de comunicación entre la Gerencia de Contabilidad de las Oficinas Generales y cada uno de los aeropuertos, mediante el envío de circulares, oficios e instructivos sobre el manejo de cuentas y catálogos.

Es de suma importancia dentro de toda contabilidad el manejo del catálogo contable, este manejo se hace de forma centralizada por niveles, esto es que hasta un nivel contable de subsubcuenta, un aeropuerto puede manipular el catálogo contable dando de baja, alta o modificando cuentas, pero en los niveles de cuentas y subcuentas, no debe hacer modificación alguna

En el apéndice A se muestra el catálogo contable que actualmente se utiliza en la red Aeroportuaria.

Una vez recibida la información en Oficinas Generales se verifican de nueva cuenta, todas las pólizas que han llegado y en general toda la información, esto se hace utilizando las facturas, pagarés, contratos, etc. de las operaciones que realizaron los aeropuertos y que respalda la información contable.

En Oficinas Generales se cuenta con un departamento de sistemas dedicado a la función contable el cual se encarga de respaldar a la Gerencia de Contabilidad en todos sus procesos contables, a través de un sistema que se desarrollo utilizando como sistema operativo UNIX, como base de datos INFORMIX ON-LINE y el programa AXCEL como herramienta de desarrollo. De AXCEL sólo de cuenta con dos licencias, que están instaladas en una computadora HP-G30 y en una computadora HP-G25. Este sistema esté dejando de ser utilizado debido a que el mantenimiento del mismo se está volviendo cada día más costoso, ya que en el mercado laboral no se encuentran personas especializadas en AXCEL, y los desarrolladores del sistema actualmente ya no laboran en el organismo, por esto sólo se consulta en algunos casos información histórica, y eventualmente se hace captura de información en el sistema.

El paquete contable COI también se encuentra en Oficinas Generales, se cuenta con una licencia de la versión que maneja noventa y nueve empresas, este paquete se instaló en una PC con sistema operativo Windows 95.

Es importante recalcar que la información se carga en ambos sistemas, a pesar de que entre ellos no hay compatibilidad alguna.

A pesar de que como ya explicamos se tiene formatos establecidos para la información contable, hay que recalcar que son muy pocos los aeropuertos que envían la información tal y como se les pide, debido a esto la labor de captura de información en Oficinas Generales se ve retrasada ya que antes se debe de hacer un análisis de la información. Esto orilla a que muchas veces se tenga que volver a hacer el trabajo que debió hacer el aeropuerto.

Una vez que toda la información fue capturada en los sistemas existentes se proceden a obtener los reportes contables, esta labor es crítica, debido a que el sistema COI que genera los reportes se encuentra instalado en una sola máquina, esto da por resultado que la emisión de reportes sea muy lenta, para las

necesidades de información, tomando en cuenta que existen aproximadamente cincuenta contadores que se encuentran en Oficinas Generales los cuales requieren reportes.

Otra de las características operativas de los sistemas que se encuentran actualmente es que la gente capacitada para su manejo es extremadamente escasa, dos personas por sistema.

Una vez que los reportes contables han sido entregados a los contadores, estos se dan a la tarea de analizarlos, para poder descubrir desviaciones, y en su caso corregirlas, lo que genera más carga de trabajo para los sistemas y la gente que los opera, esto repercute en periodos más largos para la emisión de reportes.

Después de que se analiza la información y pasa todas las validaciones que los contadores realizan, se generan reportes contables consolidados por regionales, y Red Aeroportuaria.

Por este método de trabajo la emisión de estados financieros se realiza en el mejor de los casos un mes después del cierre mensual.

En la figura 3.1.8 podemos observar el flujo que sigue la información contable que envían los aeropuertos a Oficinas Generales y en la figura 3.1.9 el procesamiento que se hace en las Oficinas Generales.

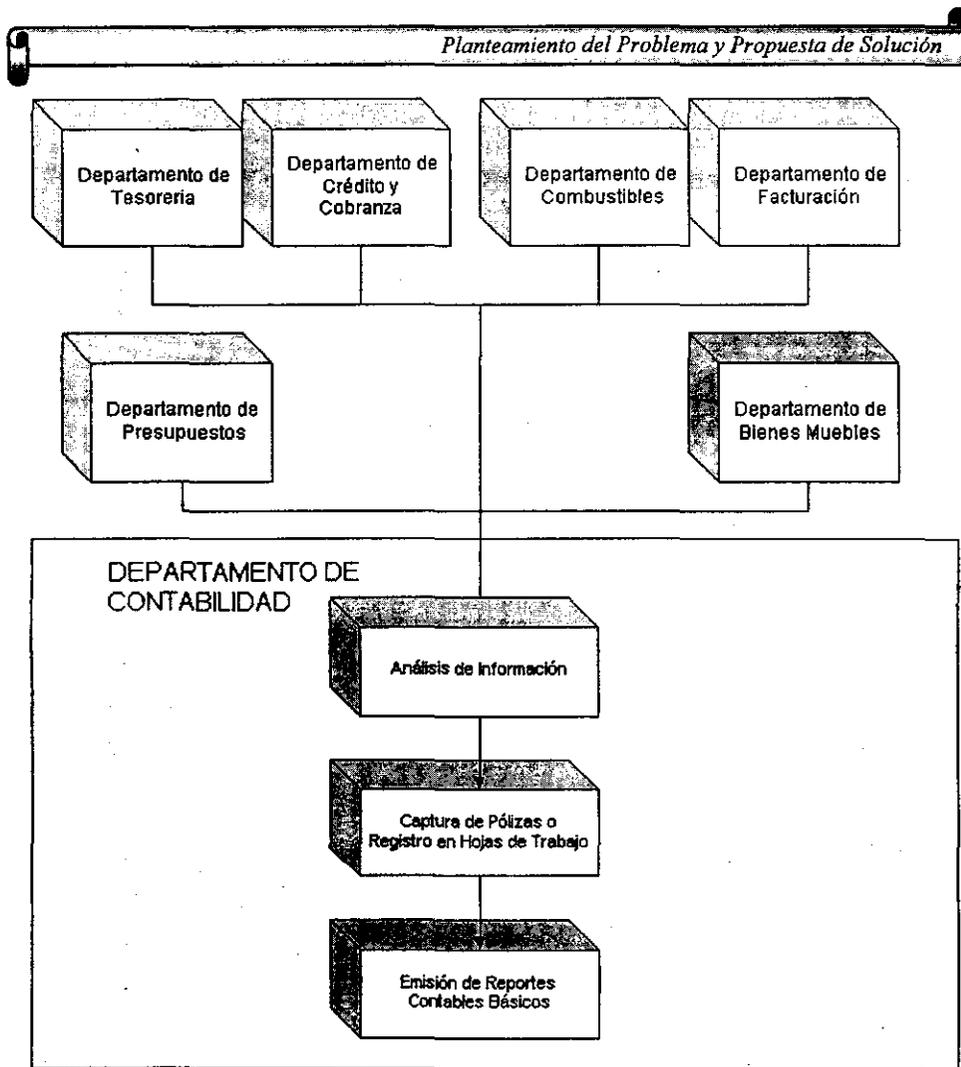


Figura 3.1.8 Flujo de información contable de los aeropuertos a Oficinas Generales

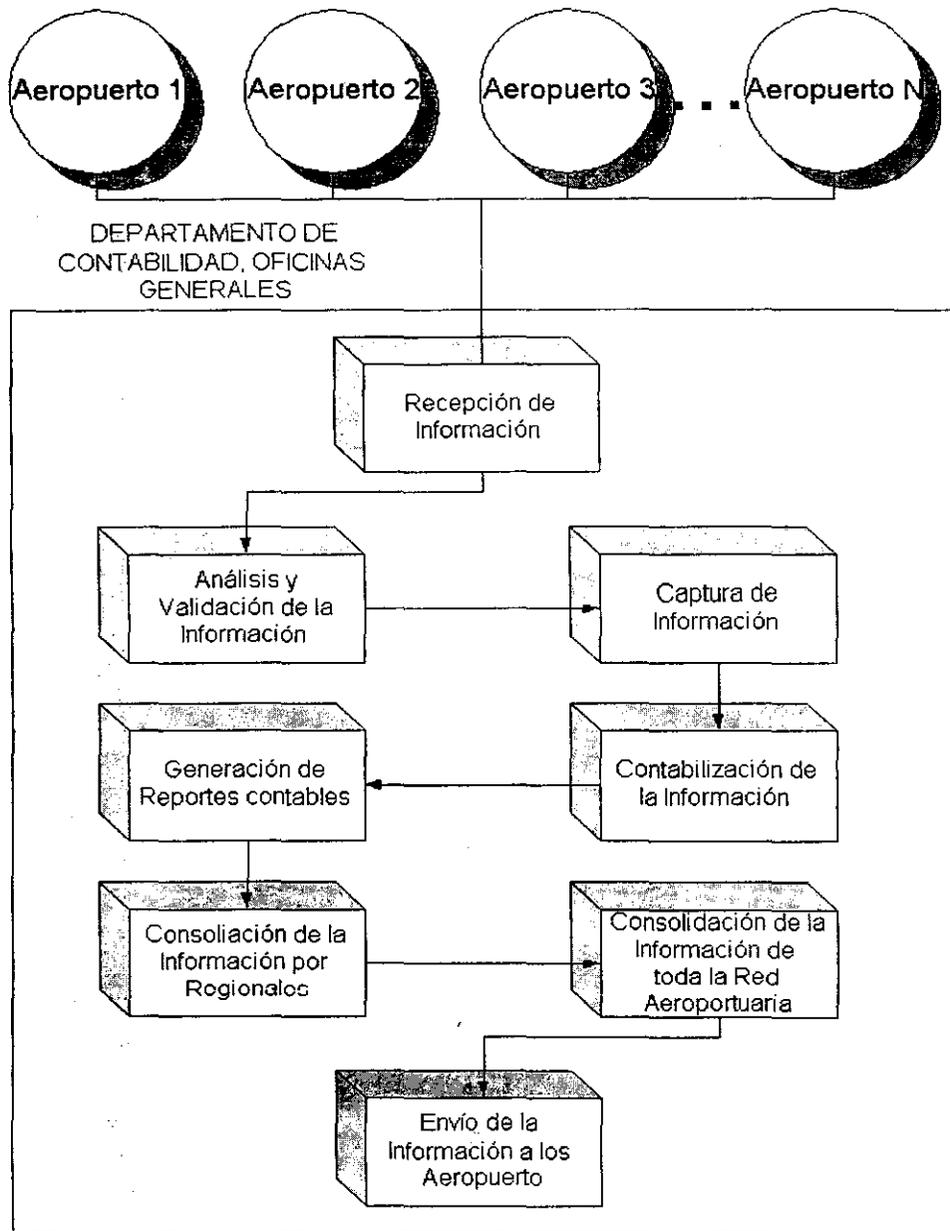


Figura 3.1.9 Procesamiento de la información contable de los aeropuertos en Oficinas Generales

### 3.2 EL USUARIO Y SUS REQUERIMIENTOS

Al hacer un análisis de requerimientos se deben de tomar en cuenta todos los elementos que conforman un sistema (hardware y software) y se deben de tener claros los siguientes puntos para lograr la completa satisfacción del usuario:

1. Identificar plenamente las necesidades del cliente. Es necesario tener una visión clara de lo que el cliente tiene en mente acerca de lo que el sistema le puede ofrecer y de lo que el necesita. Este proceso se lleva a cabo por medio de entrevistas con el cliente, en donde se marcan los objetivos del sistema, por ejemplo la información que el cliente introducirá al sistema, y la información que el sistema le puede ofrecer, las funciones que los usuarios tendrán, el rendimiento que se espera del sistema etc. Este punto no podrá ser cubierto plenamente sin una comunicación efectiva entre el cliente y el desarrollador.



Figura 3.2.1 Las entrevistas con el cliente son de vital importancia para el desarrollo del sistema

2. Estudiar la viabilidad del proyecto, esto es que el sistema puede ser llevado a cabo, muchos proyectos se han dejado inconclusos porque no eran viables. La viabilidad del proyecto puede dividirse en cuatro partes:
  - Económica: Se refiere a la evaluación del costo al momento del desarrollo comparado con el beneficio que el sistema le brindará al cliente.
  - Técnica: Aquí se analizan las restricciones que puedan afectar la posibilidad de realizar un sistema que cumpla con las expectativas del cliente, y estas restricciones son de funcionalidad y rendimiento.
  - Legal: Todo elemento que forme parte de una sociedad, se ve obligado a respetar las leyes establecidas, y en el campo de la computación no se da la excepción.
  - Alternativas: En la mayoría de los casos, siempre se pueden encontrar varias alternativas para enfrentar una determinada situación, y el estudiar estas nos brinda la oportunidad de elegir la mejor.
3. Realización de un estudio técnico y económico. Hoy en día el mundo de la computación cambia aceleradamente con lo cual un buen estudio técnico del sistema nos podrá respaldar ante esos cambios. Con respecto a lo económico hay que enfatizar que para la mayoría de las empresas en las cuales su objetivo es aumentar sus ganancias, este puede ser un punto crítico para el éxito o fracaso del proyecto.
4. Identificar las funciones que cada una de las partes que van interactuar en el sistema realizarán. Hay que determinar que funciones específicas van a realizar el software, el hardware, los usuarios, la base de datos y en si todos los elementos del sistema.

5. Establecimiento de costos y tiempos.
6. Definición del sistema para el trabajo de ingeniería que se comenzará a realizar.

Para el sistema de contabilidad de la Red Aeroportuaria Mexicana se identificaron los siguientes requerimientos de los usuarios.

Una de las preocupaciones de los usuarios es la interface, ya que han tenido problemas con otros sistemas cuyas interfaces no son adecuadas y complican su trabajo en lugar de reducirlo. La interface hacia el usuario del nuevo sistema debería de tener las siguientes características:



Figura 3.2.2 Una interface bien planeada hace que el cliente interactue más fácilmente con el sistema

- Debe ser funcional, agradable a la vista y no debe prestarse a confusiones. Para poder lograr una interface que cubra esta expectativa se deben de entender a fondo los requerimientos de la empresa y se debe de tener un buen criterio con respecto a lo estético.
- Debe de ser fácil de leer y de usar, debe estar organizada, debe de tener un comportamiento coherente, y tiene que mostrar estrecha relación con los conceptos contables.
- Opciones fáciles de acceder desde cualquier punto de la aplicación.
- La captura de datos debe de ser en extremo ágil a fin de poder capturar en el menor tiempo posible toda la información que reciben.
- Debe el sistema manejar catálogos de datos a fin de que en el momento de capturar información si esta ya se encuentra en algún catálogo el sistema muestre los datos y el usuario sólo seleccione alguno.

Otros requerimientos generales del sistema son:

- Facilidad en el aprendizaje del manejo del sistema.
- El sistema debe de instalarse en un número ilimitado de computadoras y desde cualquier computadora cualquier usuario debe de poder ver su información con la seguridad de que nadie más podrá acceder a esta.
- Debe de ser fácilmente instalable, a fin de que ellos mismos pueden realizar esta tarea.

- El sistema no debe de adueñarse en forma total de todos los recursos de la computadora, ya que al mismo tiempo quieren interactuar principalmente con hojas de cálculo.
  
- Poder tener un catálogo contable único para todos los aeropuertos para las cuentas contables acumulativas.
  
- Debe de poder generar todos los reportes contables básicos como son:
  - Pólizas Contables
  - Auxiliares de Cuentas
  - Libro diario
  - Libro mayor
  - Libros auxiliares
  - Balanza de comprobación
  - Estado de resultados
  - Estado de situación financiera
  
- Se deben de poder introducir nuevos módulos al sistema.
  
- Se deben de poder anexar posteriormente nuevos reportes.
  
- El sistema debe de contar con la seguridad necesaria a fin de que no permita que un usuario trabaje con información que no le pertenece.
  
- Todo el manejo del sistema debe de hacerse por niveles esto es que se puedan obtener los reportes contables, por aeropuerto, por regionales y por toda la Red Aeroportuaria.

- El sistema debe de poder detectar desviaciones contables, tales como sobregiros de cuentas, en el momento de que una póliza se contabilice.
- El sistema debe de manejar los errores de forma clara a fin de que el usuario pueda tomar una decisión sin tener que esperar que alguien del área de sistemas le diga que hacer.
- Debe de tener manuales sobre los temas y conceptos que se involucran en el sistema.
- La estructura de los manuales debe de permitirle encontrar la información que requiera de una manera sencilla y rápida.
- Debe de contar con un manual del usuario.
- Con respecto al hardware se debe de poder realizar el sistema con los recursos con los que cuenta la institución.

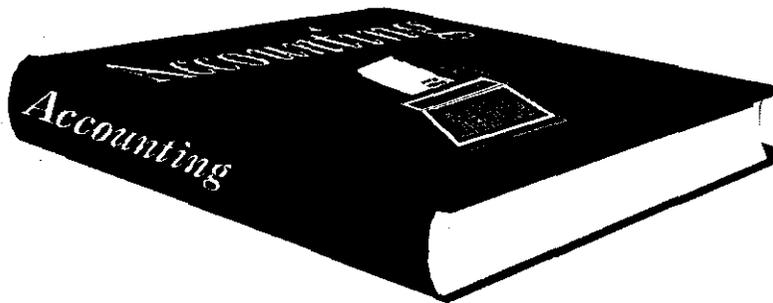


Figura 3.2.3 Un manual del usuario bien estructurado facilita las tareas del usuario.

### 3.3 RECOPIACIÓN Y ANÁLISIS DE INFORMACIÓN

Para poder construir el sistema que pueda manejar la información contable de la Red Aeroportuaria Nacional, es necesario contar con la información suficiente del trabajo que se desarrolla en las áreas contables de los aeropuertos, las sedes de las regionales y las Oficinas Generales de la Red.

Aunque ya hablamos en el punto 3.1 de las operaciones más comunes de los aeropuertos, es necesario identificar plenamente las áreas que dentro de ellos controlan esas operaciones y se encargan de generar los documentos que serán enviados a las áreas contables a fin de que se afecten las cuentas correspondientes.

Podemos clasificar a las áreas que existen en los aeropuertos como áreas operativas y administrativas, las áreas operativas son aquellas en donde se da el servicio y el área administrativa es aquella en donde se registra en un documento.

Las áreas administrativas que se encuentran en la mayoría de los aeropuertos son las siguientes:

- Facturación: se encarga de controlar la emisión de facturas por parte de la Red Aeroportuaria.
- Combustibles: se encarga del control del combustible que se les vende a los aviones.
- Personal: controla todo lo relacionado con los empleados de cada aeropuerto.
- Crédito y cobranza: se encarga de establecer contratos de créditos con los clientes de la red aeroportuaria y de tramitar el pago de los adeudos que estos contraigan en cualquier aeropuerto de la red.

- Bienes inmuebles: este departamento se encarga de controlar los inventarios de los inmuebles con los que cuenta cada uno de los aeropuertos.
  
- Tesorería: se encarga de controlar la emisión de pagos.

Estas áreas operativas se encuentran en los aeropuertos y cada una de ellas interactúa de distinta forma con las demás y con el área de contabilidad a la cual envían información y también reciben reportes de ella.

El área de contabilidad se considera un área administrativa. Esta se encarga de recibir y clasificar la información que recibe de las otras áreas administrativas, a fin de poder generar una póliza de registro contable. Ya explicamos en el punto 3.1 cuál es el formato de la póliza.

Una vez que la póliza está lista se hace el correspondiente registro contable.

Los reportes contables son generados por el área contable del aeropuerto, para tener un archivo conformado por ellos y para poder enviar una copia a las oficinas generales a fin de que se verifique la información.

En la figura 3.3.1 podemos observar gráficamente la forma en que fluye la información dentro de un aeropuerto, una sede de una regional y las Oficinas Generales.

Existen otras razones a parte de no contar con el personal contable adecuado en muchos de los aeropuertos, por las cuales los documentos que avalan las operaciones que se realizan, con particulares, viajen a la Ciudad de México, y una de las cuales es que los cobros de los servicios por lo regular se realizan en el D.F., por ejemplo la mayoría de las ventas de combustibles se dan a aeronaves que pertenecen a una determinada aerolínea y casi todas las aerolíneas importantes

tanto del país como extranjeras tienen convenios de crédito con la Red Aeroportuaria, y sus oficinas de las Aerolíneas se encuentran en el D.F.

Una vez que la información ya se ha registrado en los aeropuertos, se manda a la sede de la regional y las Oficinas Generales a fin de que se puedan obtener reportes consolidados a nivel regional y red aeroportuaria.

En cuanto a cantidades de información estamos hablando de que se manejan grandes volúmenes debido a la operación misma del aeropuerto, y la cantidad puede variar, existen algunos aeropuertos cuya afluencia de aviones es muy escasa y en cambio los que se encuentran en regiones donde hay complejos turísticos importantes tienen grandes cantidades de operaciones al día.

Cada día se generan en toda la Red aeroportuaria un promedio de 3000 facturas las cuales representan en forma individual una póliza de registro contable, de igual forma se establecen contratos de arrendamientos, de crédito, de servicios, etc. operaciones que deben de ser contabilizadas.

Uno de los problemas que se tiene con el sistema contable COI es que al ser una versión para PC la capacidad de almacenamiento es muy limitada. Y las velocidades de procesamiento de información y de generación de reportes no son óptimas.

En la figura 3.3.1. podemos observar el diagrama del flujo de información dentro de los aeropuertos y en la figura 3.3.2. se muestra como fluye la información a través de toda la red aeroportuaria.

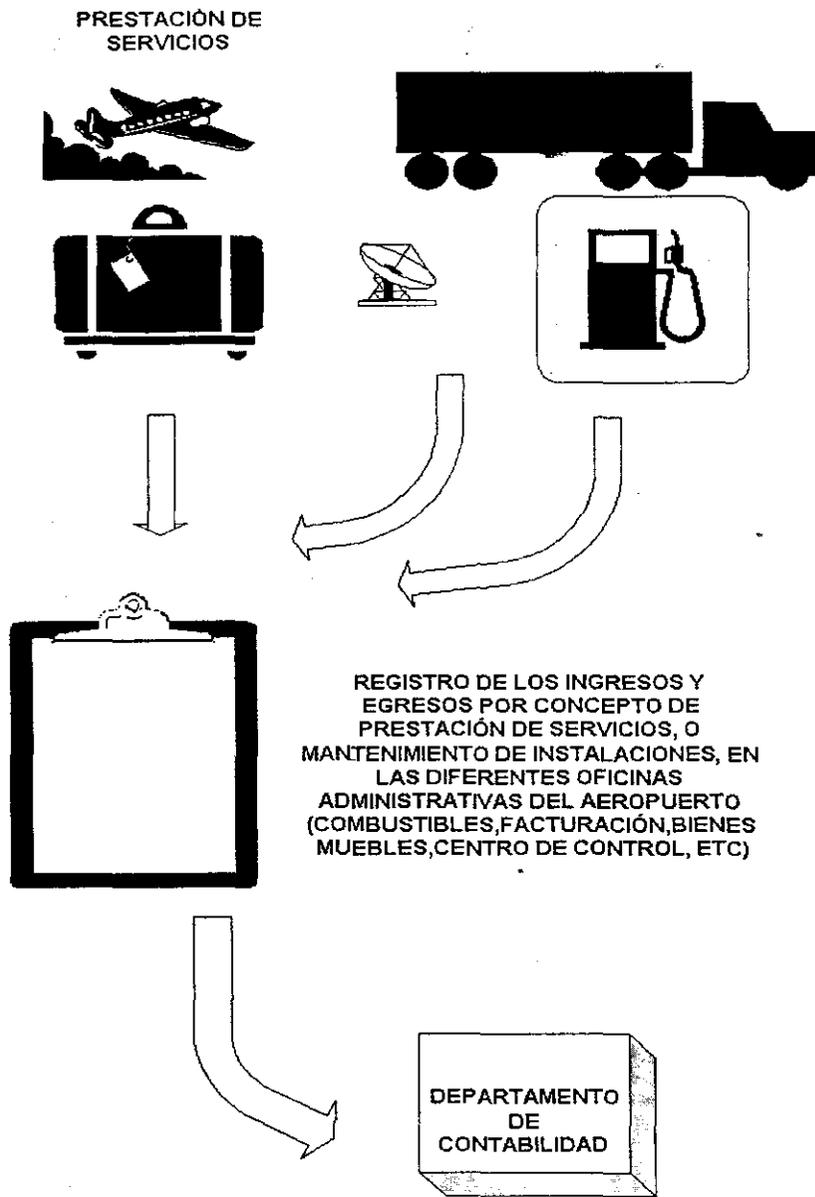


Figura 3.3.1. Flujo de información dentro de un aeropuerto hacia el área contable.

**ESQUEMA GENERAL DE LA CONTABILIDAD DE LA RED AEROPORTUARIA**

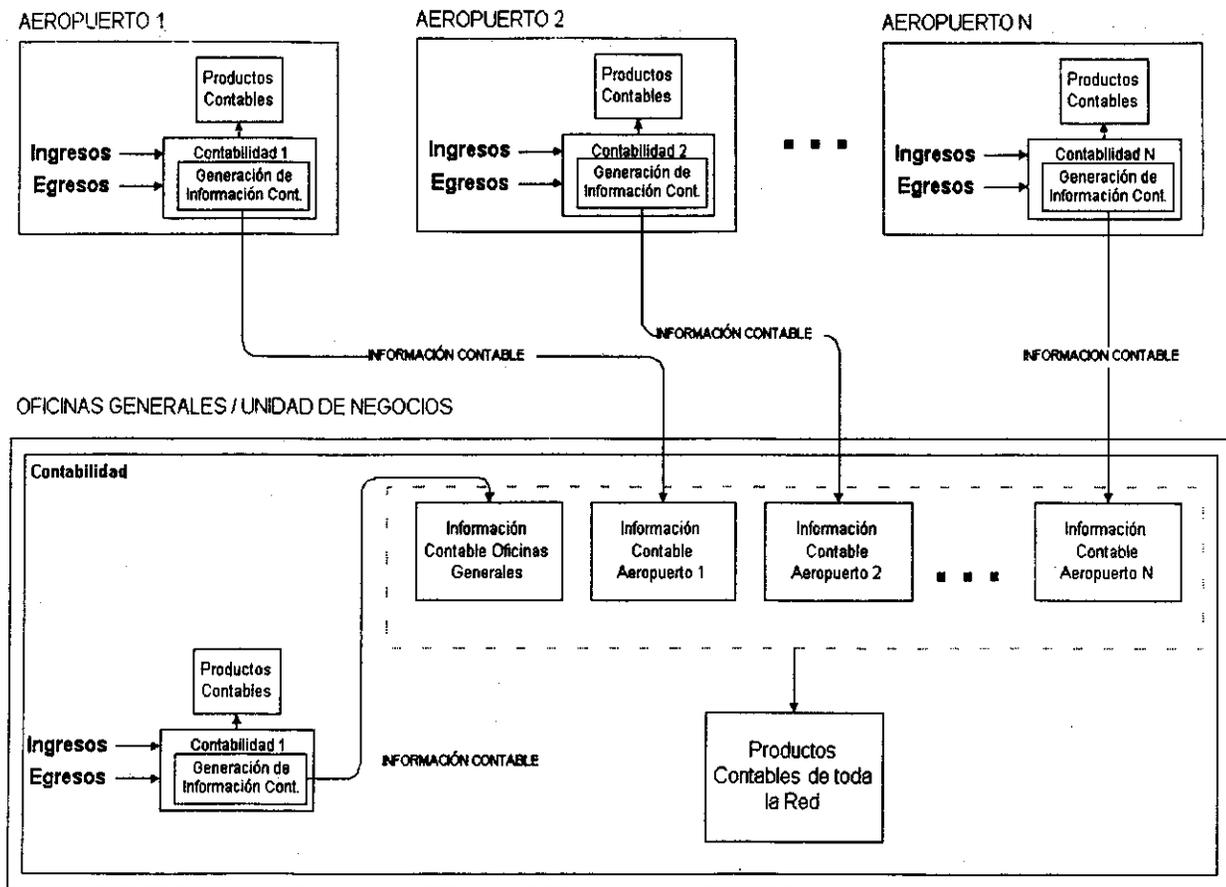


Figura 3.3.2. Flujo de Información a través de toda la red aeroportuaria.

La importancia de las oficinas de contabilidad de los aeropuertos y de las Oficinas Centrales, se basa en que es una de las principales fuentes de información de los administradores de los aeropuertos y de los funcionarios que trabajan en Oficinas Generales; con los datos que reciben toman decisiones.

Es el área de contabilidad la encargada de proporcionar información a casi todas las áreas administrativas a fin de que estas puedan realizar su trabajo.

Son muchas y muy variadas las actividades que se realizan en el área contable ya sea de un aeropuerto o de las oficinas generales, esto es debido a la gran diversidad de operaciones que se realizan en los aeropuertos y en las Oficinas Generales, además de la relación que existe con otros organismos del sector público.

Son varios los reportes que el área de contabilidad realiza y que envía a otras áreas o que utiliza ella misma para sus labores diarias.

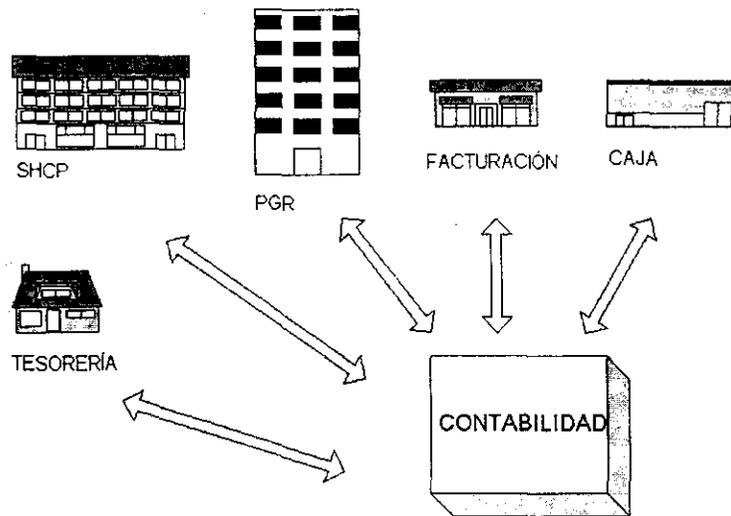


Figura 3.3.3. El área de contabilidad tiene relación con otras áreas dentro de las Oficinas Generales y con otros organismos del sector público

Es importante recalcar que los reportes elaborados por los contadores se basan en la información contable.

A continuación se hablará de las principales actividades que se realizan dentro del área contable de las Oficinas Generales:

- Autorizar los estados contables (balanza de comprobación, libros, etc.) del aeropuerto, vigilando que se remitan a las Oficinas Generales, de acuerdo a las fechas establecidas por las Oficinas Generales.
- Autorizar de conformidad los avisos de cargo o crédito por movimientos realizados por el Aeropuerto o recibidos de Oficinas Generales.
- Difundir las normas y disposiciones fiscales, contables y administrativas emitidas por el organismo.
- Autorizar de conformidad la documentación que ampare compromisos o pasivos para pago, verificando que en el segundo caso exista la creación del mismo y que cumpla ésta, con los requisitos fiscales o internos establecidos.
- Reportar los contratos por trabajos autorizados localmente así como las estimaciones correspondientes ante las Oficinas Centrales de ASA, dando aviso del inicio de la obra, ejecución y conclusión de la misma, de conformidad con las disposiciones que marca la Ley de Adquisiciones y Obras públicas.
- Coordinar, consolidar y aprobar de conformidad los anteproyectos de presupuestos de ingresos y egresos, así como los ajustes por reducción o ampliación de los mismos.
- Participar en la revisión de los presupuestos que se efectúan en Oficinas Generales, para su autorización o modificación.

- Controlar el ejercicio del presupuesto, evitando al máximo las desviaciones de los mismos, con base en los estados consolidados comparativos e informando en su caso a la Administración del aeropuerto las causas que las originaron.
- Someter a la consideración de los administradores de los Aeropuertos, los compromisos que no cuenten con partida presupuestal, tramitando en su caso, el envío de fondos extraordinarios para su pago.
- Vigilar el envío oportuno de las remesas a los aeropuertos por parte de las Oficinas Generales, de acuerdo a los presupuestos aprobados, cuidando que los recursos se apliquen en las partidas pre-establecidas.

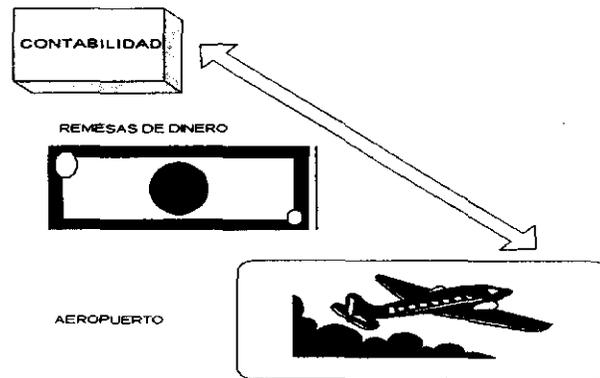


Figura 3.3.4 Contabilidad se encarga de enviar las remesas de dinero a los aeropuertos.

- Determinar la institución bancaria que otorgue mayores servicios y que de acuerdo a las indicaciones de las Oficinas Generales, convenga para el manejo

de los recursos financieros del Aeropuerto y tramitar la apertura de la cuenta correspondiente.

- Autorizar de conformidad las compras menores a través de los fondos fijos establecidos en el Aeropuerto.
- Analizar diariamente el comportamiento de los recursos financieros del aeropuerto contenido en algún reporte de disponibilidades para que de acuerdo con la programación de pagos establecida por la Administración del Aeropuerto, se distribuyan en forma adecuada y eficiente.

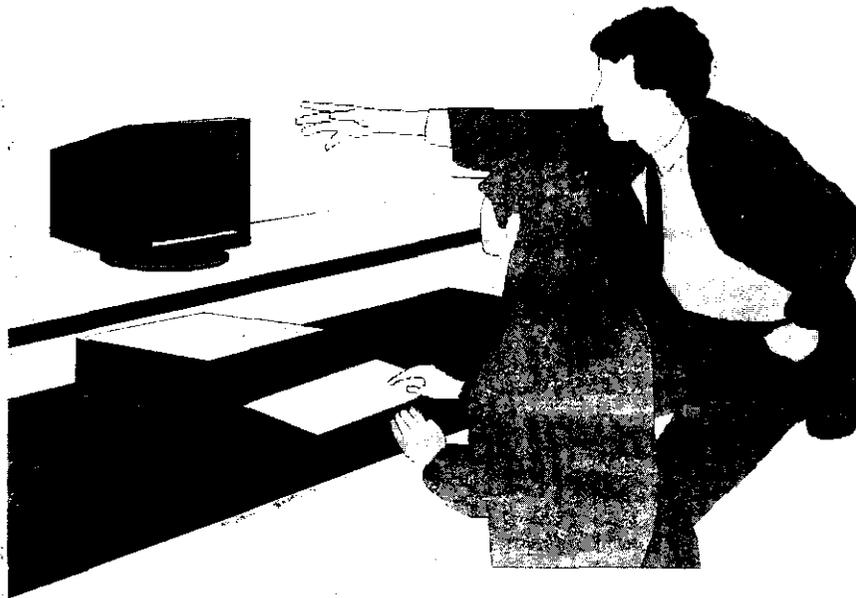


Figura 3.3.5. Contabilidad debe de analizar los recursos financieros de los aeropuertos

- Tener conocimiento de las relaciones entre las instituciones bancarias y financieras locales y los aeropuertos.
- Atender las solicitudes y observaciones de la auditoría cuando se presenten, estableciendo de inmediato las medidas correctivas que procedan en informar de éstas a la Contraloría Interna del Organismo.
- Coordinar, consolidar y formular el anteproyecto de programa anual de adquisiciones.
- Generar y formar de conformidad las requisiciones por equipos y materiales requeridos por las diversas áreas de su competencia, vigilando en todos los casos que éstas cuenten con partida presupuestal autorizada.

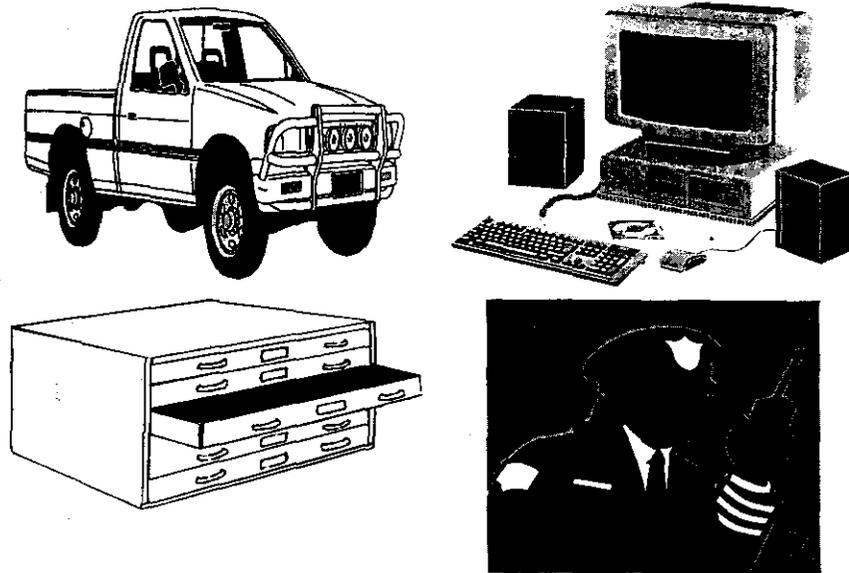


Figura 3.3.6. Las compras de equipo son controladas por el área de contabilidad.

- Vigilar que los materiales y equipos requeridos por el Aeropuerto, y que serán adquiridos a través de Oficinas Generales o localmente, se suministren en los tiempos programados.
- Vigilar que las convocatorias y concursos de adquisiciones y obra pública, se realicen en los tiempos programados y de acuerdo a las normas establecidas.
- Intervenir en las reuniones que lleve a cabo el comité de compras para autorizar la celebración de adquisiciones directas o mediante licitaciones.
- Aprobar de revisado los pedidos que se finquen localmente para la adquisición de productos, así como para las modificaciones o cancelaciones a los mismos.
- Intervenir en las reuniones semanales que se efectúen con las áreas técnicas y administrativas, para informar sobre el estado en que se encuentren sus requisiciones locales o de las Oficinas Generales.
- Supervisar que los materiales y equipos en almacenes se encuentren en condiciones de protección y seguridad.
- Supervisar que los inventarios rotativos se realicen en los tiempos y condiciones establecidas por las Oficinas Generales.
- Proponer ante la Administración del aeropuerto la baja o difusión a nivel nacional de los equipos y materiales sin uso, obsoletos o en mal estado.
- Supervisar que los trabajadores de nuevo ingreso sean afiliados al ISSSTE, al momento de realizarse la contratación.

- Autorizar de conformidad el tiempo extraordinario que se requiera en el aeropuerto, vigilando que éste haya sido previamente autorizado por la Administración.
  
- Aprobar de conformidad las nóminas, listas de raya u honorarios de los trabajadores del aeropuerto.
  
- Supervisar que el trámite para la obtención de placas, pagos de tenencia, altas y bajas, revistas y demás documentación que requieran los vehículos del aeropuerto, se efectúen en las fechas establecidas por las autoridades competentes, con el objeto de evitar multas y recargos.



Figura 3.3.6. El control de la nómina es una de las tareas más importantes del área contable.

- Promover ante la administración del aeropuerto la ocupación de locales y espacios comerciales que por algún motivo de encuentren desocupados.
- Supervisar que los arrendatarios cumplan con las condiciones establecidas en los contratos en cuanto a giros, espacios, pago de rentas, participaciones, etc., y que en su caso, se cobren los intereses moratorios.
- Tramitar y gestionar la recuperación oportuna de los fondos por daños incurridos a los equipos e instalaciones ante la compañía aseguradora contratada por el Organismo.
- Realizar la investigación local sobre la experiencia, solvencia económica, comparativo de precios, antecedentes comerciales de los solicitantes y antecedentes en actividades aeronáuticas.
- Formular los estudios especiales y los informes que requiera la administración.
- Llevar la contabilidad general de los Aeropuerto en base a los procedimientos y normas establecidas por la Red Aeroportuaria.
- Recibir y revisar que la documentación con la cual se contraigan compromisos o pasivos, cumpla con los requisitos fiscales o internos establecidos, así como tramitar el pago.
- Formular, codificar y aprobar las pólizas de ingresos, egresos y diario.
- Registrar en el libro diario y auxiliares y libro mayor la operaciones contables.
- Elaborar mensualmente la balanza de comprobación de los aeropuertos así como los informes periódicos o extraordinarios que se le requieran.

- Realizar la conciliación y depuración mensual de las cuentas de balance.
- Organizar, integrar y controlar el archivo de la documentación contable y financiera de los aeropuertos.
- Revisar con base en los documentos de pago que la caja haya efectuado el egreso correcto a los trabajadores por nómina y lista de raya,
- Coordinar con las áreas de personal y caja, el pago oportuno de las declaraciones de impuestos y cuotas al FOVISSSTE, ISSSTE, SAR, FONACOT, etc. , a efecto de evitar multas, recargos y sanciones.
- Promover ante la administración, la recuperación de los diversos adeudos que tengan los empleados o terceros con el organismo.
- Realizar las conciliaciones de las cuentas de enlace en los tiempos pertinentes.
- Realizar la solicitud para gastos de comisión oficial para su revisión y en su caso aprobación, así como la cuenta de gastos presentados por los mismos.
- Calcular y elaborar las bonificaciones o penalizaciones monetarias a favor del organismo por retrasos en la entrega de materiales y servicios por parte de terceros.
- Revisar y contabilizar los ingresos por concepto del cobro de tarifas de uso aeroportuario, así como el depósito bancario correspondiente.
- Registrar en auxiliares individuales los anticipos que se otorguen a proveedores y contratistas, controlando la antigüedad de los mismos y efectuando los ajustes en la facturación que se presenten.

- Realizar las conciliaciones mensuales de las cuentas de activo fijo, materiales en almacenes y materiales en tránsito, entre los saldos de contabilidad y almacén.
- Calcular y deducir de la facturación y/o estimaciones que presenten los proveedores y/o contratistas, los descuentos por concepto de ley.
- Reportar semanalmente a la caja la relación de los pasivos o compromisos que se contraigan, para la formulación del programa de tesorería.

Los puntos anteriores son un resumen de las actividades más importantes que se realizan dentro de las áreas contables de la Red Aeroportuaria, ya sea en los aeropuertos, en las cedes de las unidades de negocios o en las Oficinas Generales. Para realizar estas actividades, los contadores se auxilian de los reportes de los sistemas que actualmente se encuentran trabajando, pero hay que enfatizar que los reportes que estos sistemas generan son incompletos. Los reportes que generan para su trabajo son principalmente el de la balanza de comprobación y los libros auxiliares de cuentas.

Para poder llevar a cabo todas las tareas que se mencionaron anteriormente, de los reportes contables que obtienen, generan otros, ayudados de hojas de cálculo principalmente.

Todas las tareas que realiza el área contable, necesitan de información que proviene de la contabilidad de los aeropuertos y de la red en general. Al no contar con la información contable adecuada, muchos de los reportes que los contadores requieren no son tomados de la contabilidad que se tiene registrada, sino de los documentos fuente de las operaciones, esto es un factor sumamente importante que hace que el trabajo de las áreas administrativas se retrase, debido a que el manejar los documentos fuente cuando hablamos de cientos e inclusive miles de ellos es totalmente inadecuado.

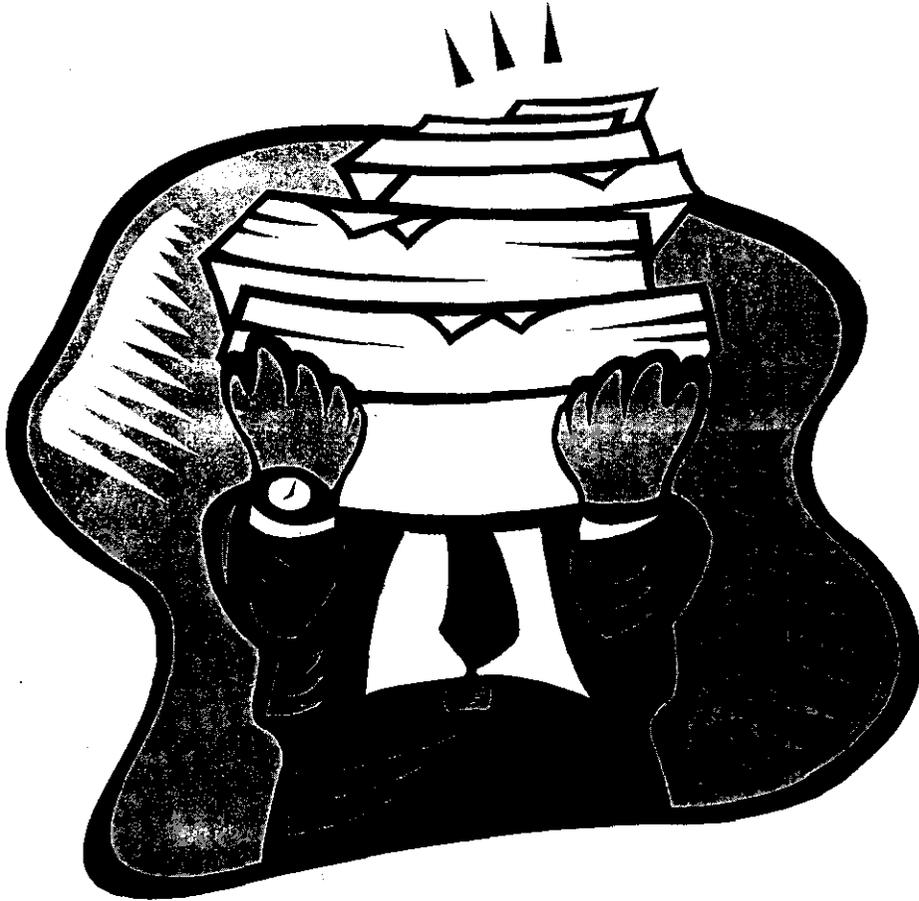


Figura 3.3.7. El manejo de la información fuente de las operaciones es algo inadecuado cuando hablamos de cientos o miles de operaciones.

### 3.4 PLANTEAMIENTO DEL PROBLEMA

La red aeroportuaria nacional cuenta con 61 aeropuertos, los cuales están agrupados por regionales, el manejo de la contabilidad se hace de forma centralizada en la Ciudad de México en las Oficinas Generales de la Red Aeroportuaria.

La cantidad de información que se concentra en las Oficinas Generales es demasiada, y al no tener un sistema de contabilidad adecuado, el registro de las operaciones contables se vuelve una labor sumamente difícil. No se cuenta con información de manera oportuna, y los registros muchas veces son erróneos, el uso de varios sistemas, dificulta la interpretación de los datos y complica la búsqueda de información.

El no tener un registro confiable de la información contable hace que se tengan que consultar los archivos en papel con los cuales cuenta la institución, labor que retrasa de manera considerable la entrega de información.

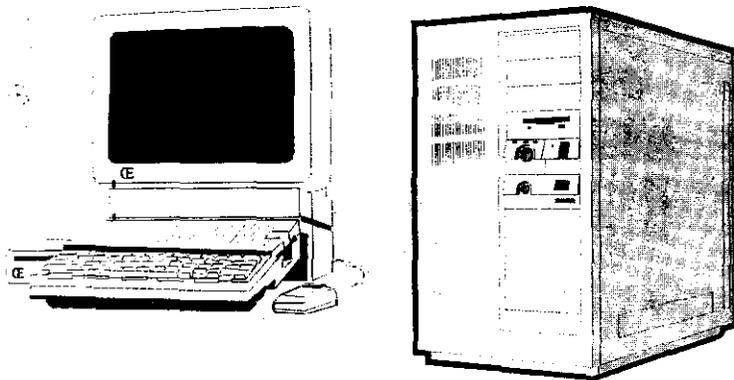


Figura 3.4.1. El uso de varios sistemas no compatibles entre si y sin ninguna relación dificulta la tarea contable.

Los cierres contables de cada mes se hacen por lo regular un mes después de la fecha de cierre, por lo cual, la información del estado financiero de cada aeropuerto es conocida en toda la red aproximadamente mes y medio después.

Con los sistemas que se encuentran trabajando actualmente sólo se logra obtener los estados financieros de cada aeropuerto, y la elaboración de los estados financieros de una regional, es una tarea que debe de realizarse de forma manual auxiliados solamente por hojas de cálculo.

Los sistemas con los que se labora actualmente son difíciles de aprender y no son lo suficiente amigables para poder ser utilizados por cualquier usuario.

No se cuenta con seguridad en la información cualquier usuario puede acceder los sistemas actuales, además de que en caso de una pérdida de información es casi imposible volverla a recuperar.



Figura 3.4.2. Actualmente los estados financieros de las regionales se elaboran de forma manual.



Figura 3.4.3. Los sistemas actuales con que trabaja la institución no cuentan con seguridad y no son amigables con los usuarios.

En las páginas anteriores analizamos de manera general cuáles son los problemas que se han detectado hasta el momento dentro de las Oficinas Generales, que hacen que el procesamiento de la contabilidad de la Red Aeroportuaria se vea retrasado y se haga de manera incorrecta en algunos casos.

Ya mencionamos que en la institución se cuenta con dos sistemas que auxilian a los contadores a elaborar los registros contables.

Hay que explicar la razón por la cual la institución cuenta con dos sistemas trabajando al mismo tiempo en lugar de adoptar uno sólo y trabajar con ese. El principal motivo es la información que se encuentra cargada en cada uno de los sistemas. En el sistema que se desarrolló en la institución hace algunos años y que se utilizó como manejador de base de datos Informix, se hizo una carga inicial de datos, y se empezó a trabajar con el, agregando cada día más información, al paso del tiempo, el sistema contiene muchos datos pero se hace cada vez más obsoleto, por lo cual se adopta como nuevo sistema para manejar la contabilidad el sistema contable COI el cual ofrece muchas ventajas con respecto al primero, pero no se puede cargar la información con la que cuenta el primer sistema, y actualmente las necesidades de los contadores de la institución hacen que se tenga que acceder a la información que se tiene en los dos sistemas, aunque en el primero principalmente se consulta información histórica.

Ahora vamos a analizar la problemática de cada uno de los sistemas.

El primer sistema que analizaremos es el que se elaboró hace algunos años en la institución; sus principales problemas los mencionaremos a continuación:

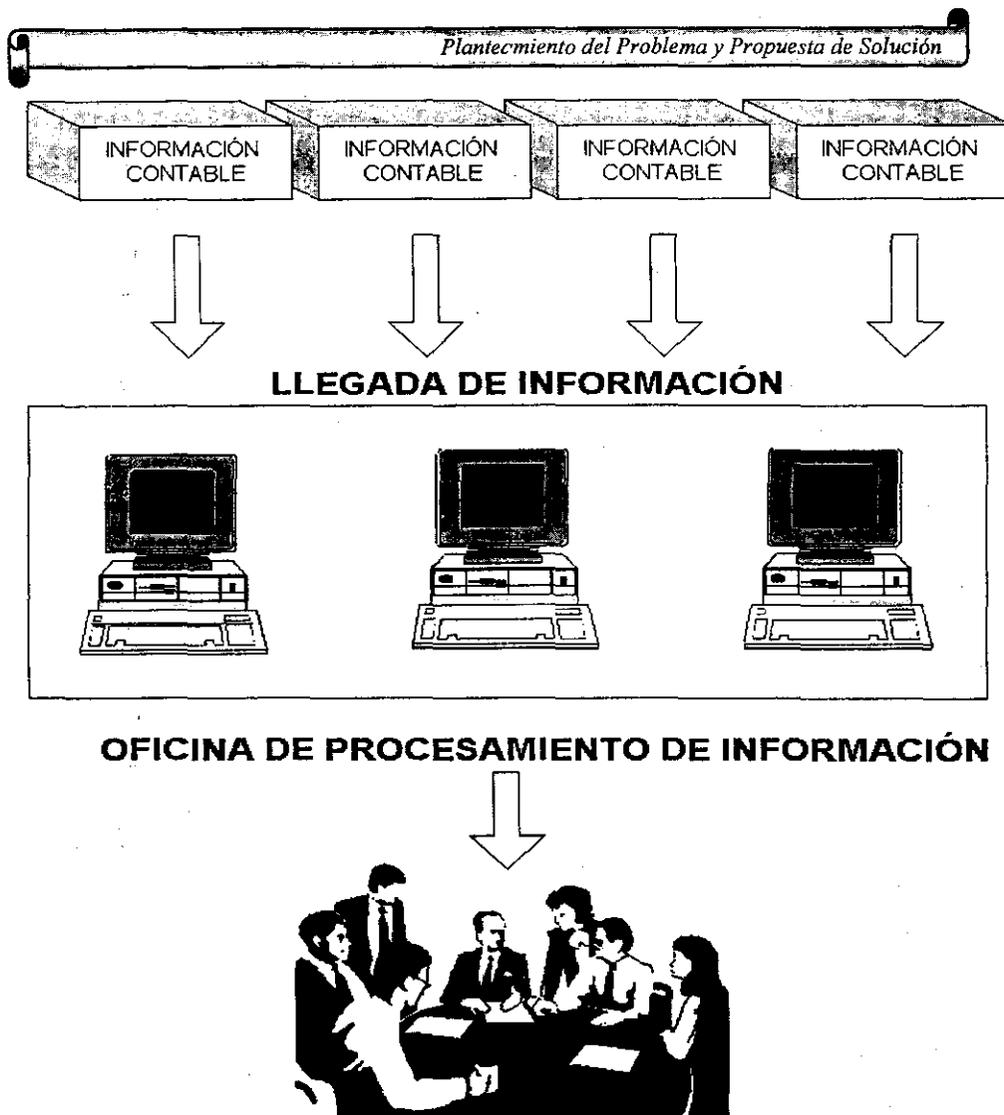
- Es un sistema poco amigable para el usuario, sus pantallas de captura son muy rudimentarias y no presentan ningún tipo de ayuda; el proceso de aprendizaje del uso del sistema es difícil y tardado.

- Se han detectado errores en la información, debido a cálculos erróneos, hechos por el sistema, el solucionar estos problemas ha retrasado días e incluso semanas los cierres contables.
- No se cuenta con personal suficiente que pueda dar mantenimiento al sistema, además de que dicho mantenimiento es difícil, ya que no se cuenta con documentación alguna.
- Los reportes que emite el sistema son en extremo incompletos, y difíciles de interpretar.
- Una vez que se ha capturado toda la información se corren procesos que tardan varias horas, y en caso de errores a la mitad de un proceso, se tiene que comenzar de nuevo.
- La seguridad de los datos es pobre, se han dado pérdidas de información, que a su vez producen inconsistencias de datos al cargar respaldos de información no actualizados completamente.
- El sistema no cuenta con claves de acceso para los usuarios, sólo cuenta con una clave general que ya es del conocimiento de todo el personal que labora en el área contable, por esta razón todos pueden acceder el sistema, y ver todos sus módulos sin cuidar que datos puede ver o procesar un usuario determinado.
- El crear nuevos módulos para el sistema es una tarea difícil debido a la falta de documentación.
- Existe una oficina en donde se generan las impresiones de los reportes que necesitan los contadores, y en caso de que los operadores no se encuentren, no se pueden tener los reportes, ya que ninguno de los usuarios contables del

sistema puede generar los reportes, a pesar de que puede manipular información y revisarla en pantalla.

Otro de los sistemas con los que cuenta la institución es el sistema contable COI, y a pesar de que es uno de los más populares en México, y en muchas empresas se encuentra funcionando se han encontrado varios puntos críticos que hacen que la generación de los estados financieros de la institución se haga con demoras; a continuación mencionaremos los problemas que se han detectado:

- El sistema cuenta con dispositivos que el fabricante entrega en el momento de la compra y que se colocan en el puerto paralelo de la computadora en donde se encuentre instalado el programa, sólo con este dispositivo el programa puede funcionar, estos dispositivos son llamados candados, si se quiere habilitar otra terminal se tiene que comprar el respectivo candado. Esto genera un cuello de botella debido a que no se cuenta con los candados suficientes para poder trabajar en varias máquinas, sólo se encuentran funcionando actualmente tres máquinas, lo que hace que la generación de reportes para los contadores sea lenta, tomando en cuenta que existen aproximadamente cincuenta contadores los cuales necesitan reportes de diferentes tipos.
- Los reportes que entrega el sistema son pobres en cuanto a contenido y presentación, no entrega toda la información que el usuario requiere y de la forma en que la requiere.
- La información se almacena en un formato especial, el cual sólo es reconocido por el paquete, se cuenta con opciones de exportación de información. Las cuales son limitadas y no contemplan toda la información en su conjunto.



**APROXIMADAMENTE 50 CONTADORES EN ESPERA DE INFORMACIÓN**

Figura 3.4.4. Existe una oficina que se encarga de obtener los reportes del sistema COI, esta oficina es un cuello de botella, cuenta con tres máquinas y tres personas encargadas.

- No se cuenta con esquemas adecuados de recuperación de información, en caso de pérdida de datos. Tiene opciones de respaldo, las cuales no son óptimas para el caso de los volúmenes de información de los que hemos venido hablando.
  
- No se pueden hacer adecuaciones al sistema, y la introducción de nuevos módulos, sólo se pueden hacer por medio de la compra de los otros productos que ofrece el mismo fabricante que elaboró el sistema COI.

Hasta aquí hemos analizado los problemas particulares de cada uno de los sistemas, pero además es importante mencionar que el uso de dos sistemas dificulta las tareas que se realizan en el área contable y esto se debe por los motivos que se mencionan a continuación.

- El tener dos sistemas hace que un usuario reciba reportes con formatos diferentes y conteniendo sólo parte de la información cada uno de ellos, por esto en muchos de los casos la información se tiene que volver a capturar en alguna hoja de cálculo, a fin de poder realizar sus cálculos correspondientes y emitir un reporte que tenga la información de los dos sistemas.
  
- Debido a que el área contable guarda relación con casi todas las áreas de la institución es necesario que se desarrollen nuevos módulos del sistema a fin de poder facilitar el trabajo de los contadores, por ejemplo uno de los módulos que sería de gran utilidad es el de la generación automática de los reportes de las conciliaciones bancarias, pero estos módulos no pueden llevarse a cabo, ya que la información que necesita, se encuentra en los dos sistemas, lo cual haría que el módulo fuera complejo, no por los procesos que realice sino por la extracción de información de otros sistemas y la garantía de consistencia de la misma.

- Mucha de la información debe de capturarse dos veces para poder alimentar a los dos sistemas, como los sistemas se encuentran en plataformas distintas la exportación y/o importación de archivos no se da, lo que genera errores humanos al capturar, por hacerse este trabajo dos veces.

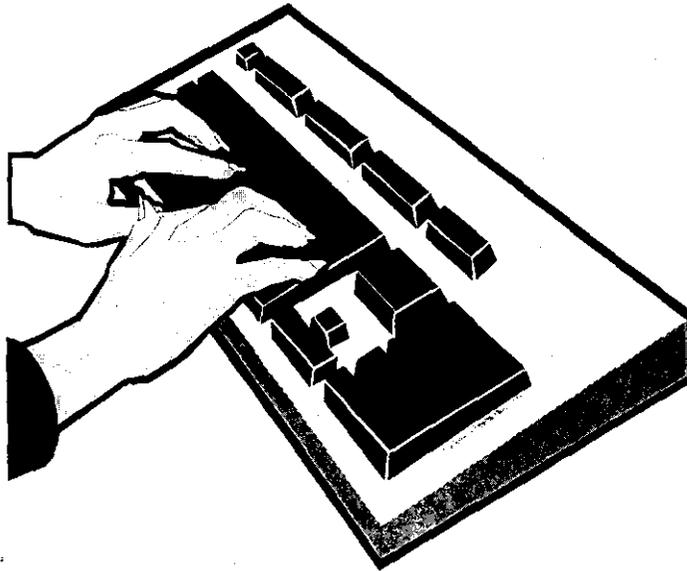


Figura 3.4.5. El capturar la misma información en los dos sistemas con que cuenta la institución genera errores humanos al momento de capturar.

- Ninguno de los sistemas tiene un manejo adecuado del catálogo contable, ya que como el manejo de la contabilidad de cada aeropuerto lo hacen en forma aislada, lo que trae como resultado que los catálogos de los aeropuertos sean en algunos casos diferentes entre sí, generando un gran número de problemas al momento de hacer la consolidación contable para generar balances de las regionales y de la red aeroportuaria.

### **3.5. DESCOMPOSICIÓN FUNCIONAL**

Procedimientos sistemáticos con notaciones únicas son combinados en un análisis de requerimientos a fin de analizar los dominios de información funcional de un problema.

El dominio de la información se caracteriza por tres atributos:

- Flujo de datos
- Contenido de los datos
- Estructura de datos

Todas las técnicas de análisis de requerimientos tienen características comunes las cuales vamos a mencionar a continuación:

- Son mecanismos para el análisis del dominio de la información.
- Nos auxilian en la definición de las interfaces del sistema.
- Con ellas podemos subdividir el problema que estudiamos.
- Son una forma de abstracción.

- Auxiliados con ellas podemos hacer una representación funcional del problema.
- Nos ayudan a realizar las representaciones de las visiones físicas y lógicas.

Siempre vamos a encontrar diferencias entre las diferentes metodologías de análisis de requerimientos, pero hay que saber que todas ellas tienen formas de conducirse análogas. Y como guías comunes entre todas ellas podemos encontrar las siguientes:

- Siempre van enfocadas al flujo de datos y a la estructura de los mismos.
- Se generan funciones que son vistas como procesos que transforman información.
- Se puede evaluar la construcción física de la solución del problema antes de que esta se lleve a cabo.
- Se utilizan mecanismos de diccionarios para poder representar en forma explícita el contenido de los datos.
- El flujo de información va encaminado siempre al análisis de las transformaciones que se aplican para cambiar una entrada de datos en una salida.

En las técnicas orientadas al flujo de datos es necesario que las personas que se encargan de la construcción de un sistema tomen en cuenta lo siguiente al momento de hacer el análisis de requerimientos:

- Los procesos que van a integrar al sistema.
- Los datos que se usan en cada proceso.

- Los datos que deben de almacenarse.
- Los datos de entrada y los de salida.

Esta técnica orientada al flujo de datos reconoce la importancia de los datos en una organización y los procesos que se encargan de transformar a estos, los objetivos de la empresa se ven desde el punto de vista de los datos que se desean obtener.

Los diagramas de flujo de datos son las herramientas utilizadas por las técnicas orientadas al flujo de datos, estos muestran la relación entre los procesos y los datos. Un sistema realiza principalmente un trabajo básico de transformación de datos de entrada en datos de salida como se muestra en la siguiente figura.

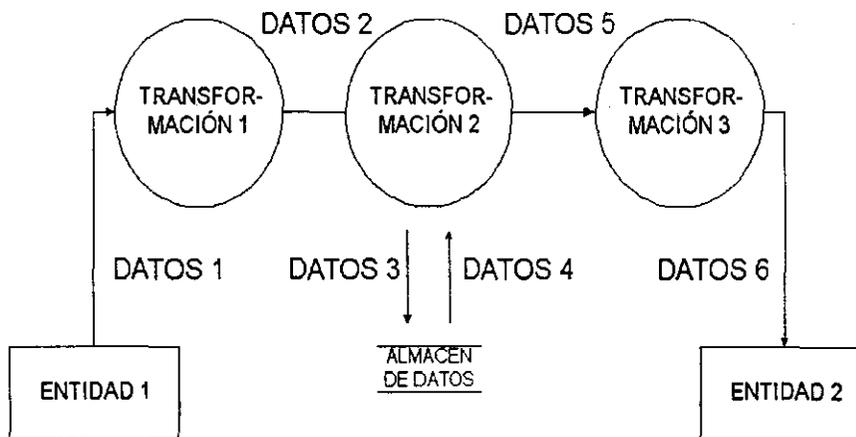


Figura. 3.5.1. Representación del Diagrama de Flujo de Datos.

Un sistema con las técnicas orientas al flujo de datos es mostrado como una red de procesos conectados entre si por medio de trayectorias, con esto se puede describir la forma en que el sistema está relacionado con el mundo real, y cómo las partes del sistema de relacionan unas con otras, además de cómo fluye la información y cuáles son los datos de entrada a un proceso y cuáles son los datos de salida que arroja el mismo.



Figura 3.5.2. Forma básica de un diagrama de flujo de datos.

Entre los principales usos de los diagramas de flujo de datos podemos encontrar:

- Se puede representar un sistema a diferentes niveles de abstracción.
- Dividir un sistema.
- Hacer un análisis de las interfaces que serán parte del sistema.
- Es una forma de comunicación entre los usuarios y diseñadores.
- Desarrollo de especificaciones.

- Dar consistencia entre procesos.

El funcionamiento de los diagramas de flujos radica en no introducir en el "nivel inferior" nuevas entradas o salidas en el siguiente proceso, que no se hayan identificado en los diagramas de mayor nivel, sin embargo, si es posible identificar nuevos flujos de datos y almacenamiento de los mismos.

Algunas de las ventajas del uso de los diagramas son:

- Podemos representar las actividades que se realizan en forma paralela dentro de la empresa.
- Con la concentración de los datos y de los procesos, los desarrolladores pueden tener una visión clara y total de todo el sistema.
- El desarrollador puede estudiar de manera sencilla lo que pasa con los datos al entrar a algún proceso y los datos que son obtenidos después del procesamiento.
- Se pueden aislar diferentes áreas de interés para poder ser estudiadas.
- Como la notación es sencilla personas que no estén relacionadas con el área computacional pueden fácilmente entender los diagramas y conocer rápidamente el diseño del sistema.
- Pueden detectarse errores en los procesos.
- Pueden desarrollarse en forma progresiva, esto es un proceso puede descomponerse en diferentes niveles dependiendo del nivel de abstracción que el desarrollador desee.

- Señalan que ocurre en la aplicación y cuando.
- Se pueden desarrollar y analizar especificaciones.
- Descompone todo un sistema a un nivel en el cual todos los procesos son funciones simples.

El diccionario de datos es una parte importante de las técnicas orientadas al flujo de datos y nos describe el contenido de los elemento de información de los diagramas de flujo de datos.

El diccionario de datos nos proporciona:

- Información adicional sobre el sistema.
- Ayuda al desarrollador al análisis de los requerimientos del sistema.
- Describe la actividad o entidad que se está estudiando
- Nos muestra los detalles adicionales de los diagramas de flujos de datos.
- Una visión clara sin ambigüedades de los elementos de información.
- Ilustra tanto los datos, como flujos a través de un sistema y las transformaciones que se dan en los datos del mismo.

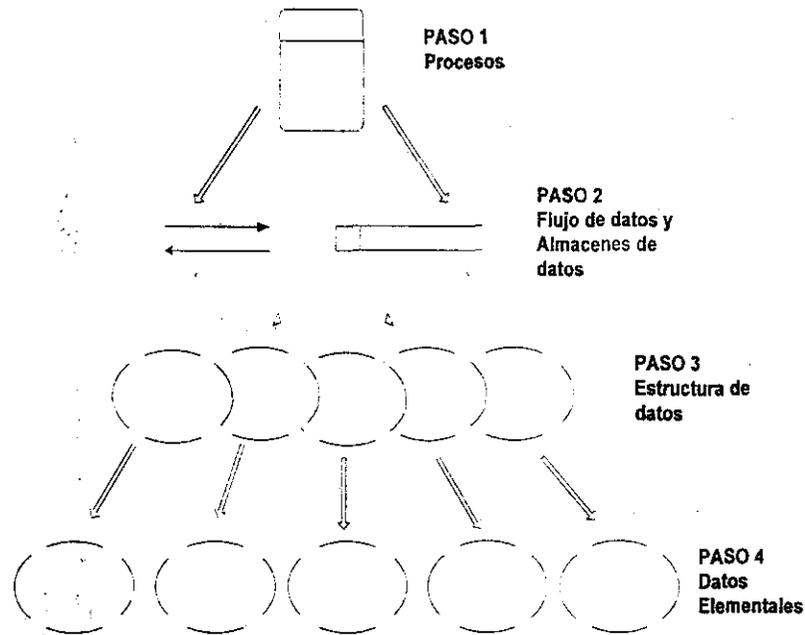


Fig. 3.5.3. Pasos para la construcción de un diccionario de datos.

Las ventajas que podemos encontrar en el uso de un diccionario de datos es que el usuario el tener un conjunto conciso de definiciones para todas las entidades de los procesos que conforman al sistema de información que se está desarrollando; es una fuente común de definiciones que es utilizada por los desarrolladores; nos permite depurar nuestros diagramas de flujos de datos, permitiendo detectar faltantes, duplicados.

El diccionario de datos nos proporciona una información completa de todos los datos utilizados en el sistema, se pueden tener listados que incluyan de los datos, sus nombres, descripciones, longitudes de los campo y nombres alternos.



DESCRIPCIÓN BREVE: \_\_\_\_\_

SINÓNIMOS (CONTEXTOS): \_\_\_\_\_

ALFABÉTICO: _____ ALFANUMÉRICO: _____ NUMÉRICO: _____	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center; padding: 5px;">SI ES DISCRETO</th> </tr> <tr> <th style="width: 50%; padding: 5px;">VALOR</th> <th style="width: 50%; padding: 5px;">SIGNIFICADO</th> </tr> </thead> <tbody> <tr> <td style="height: 100px;"></td> <td style="height: 100px;"></td> </tr> </tbody> </table>	SI ES DISCRETO		VALOR	SIGNIFICADO		
SI ES DISCRETO							
VALOR	SIGNIFICADO						

SI ES CONTÍNUO

LONGITUD: \_\_\_\_\_

RANGO: \_\_\_\_\_

Si son más de cinco valores, utilice el reverso de la tarjeta o tarjetas adicionales

Figura 3.5.4. Ejemplo de una forma utilizada para integrar un diccionario de datos.

DIAGRAMA DE DESCOMPOSICIÓN FUNCIONAL

Basándonos en el análisis de requerimientos podemos establecer los módulos principales del sistema de contabilidad que nos ocupa en este trabajo.

- A) Seguridad
  
- B) Manejo del catálogo contable
  
- C) Captura de pólizas

D) Administración de pólizas

E) Aplicaciones Contables

F) Reportes

G) Cierres contables



Figura 3.5.5. Diagrama de descomposición funcional (Módulo principal)

### SEGURIDAD

Para solucionar el problema de acceso a la información por parte de los usuarios, es importante que el sistema cuente con un módulo que se encargue del control de los accesos de los usuarios al sistema. Este módulo contendrá las opciones para el control de usuarios (altas bajas, modificaciones etc. ), cambio de contraseñas de entrada al sistema, control de los procesos que un usuario puede acceder, así como

las opciones del menú. Con esto la confidencialidad de la información será garantizada y evitará malos manejos de los datos. Se sabrá que usuarios modificaron la información, y sobre todo cada usuario observará y procesará la información que le corresponda.

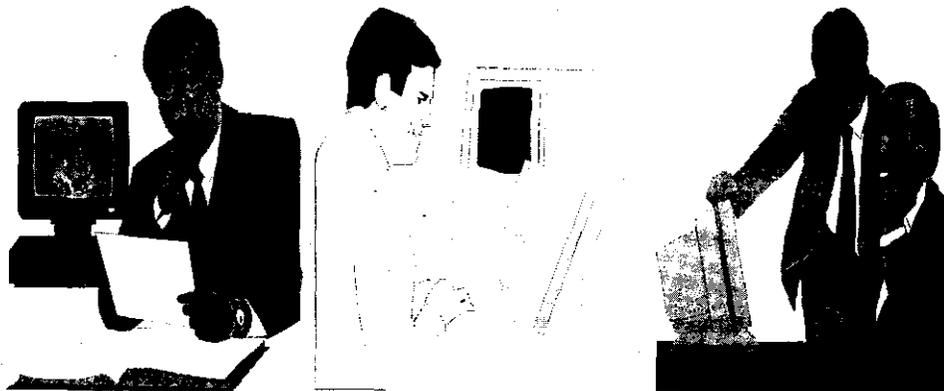


Figura 3.5.6. El módulo de seguridad del sistema permitirá que cada usuario vea y procese la información que le corresponda.

#### MANEJO DEL CATÁLOGO CONTABLE

Debido a que una de las partes más importantes de cualquier contabilidad es el catálogo de cuentas que maneja, se considerará un módulo que se encargue del manejo de las cuentas en donde se darán de alta, baja o modificarán , y se controlarán sus características contables de cada una de ellas, así como el manejo de sobregiros y horarios. Es importante recordar que uno de los principales problemas que se han detectado es la incompatibilidad de los catálogos contables

de los diferentes aeropuertos, ya que los contadores encargados de llevar la contabilidad de un aeropuerto en las oficinas generales, aunque se han normado las características de las cuentas contables, han dado de alta, baja o modificado cuentas de manera incorrecta, esto como mencionamos en el planteamiento del problema, trae consigo grandes problemas al momento de tratar de obtener la contabilidad consolidada de toda la red aeroportuaria.

El módulo del manejo del catálogo contable manejará un catálogo normativo que será la base para los catálogos de todos los aeropuertos, este catálogo normativo sólo podrá ser modificado por las personas que tengan ese privilegio, y las modificaciones que hagan afectarán directamente a todos los catálogos de los aeropuertos a fin de que todos los aeropuertos tengan la misma estructura de cuentas en los niveles acumulativos, entendiendo como niveles acumulativos, los niveles en los que se encuentran las cuentas de mayor.

Además el sistema en el módulo de catálogos tendrá un manejo por niveles, esto es que los saldos de los catálogos de los aeropuertos, se concentrarán en los catálogos regionales y los catálogos regionales se concentrarán en el catálogo normativo, con esto, se podrán obtener todos los reportes contables, por aeropuerto, regionales y de toda la red.

**ESQUEMA GENERAL DE LA CONTABILIDAD DE LA RED AEROPORTUARIA**

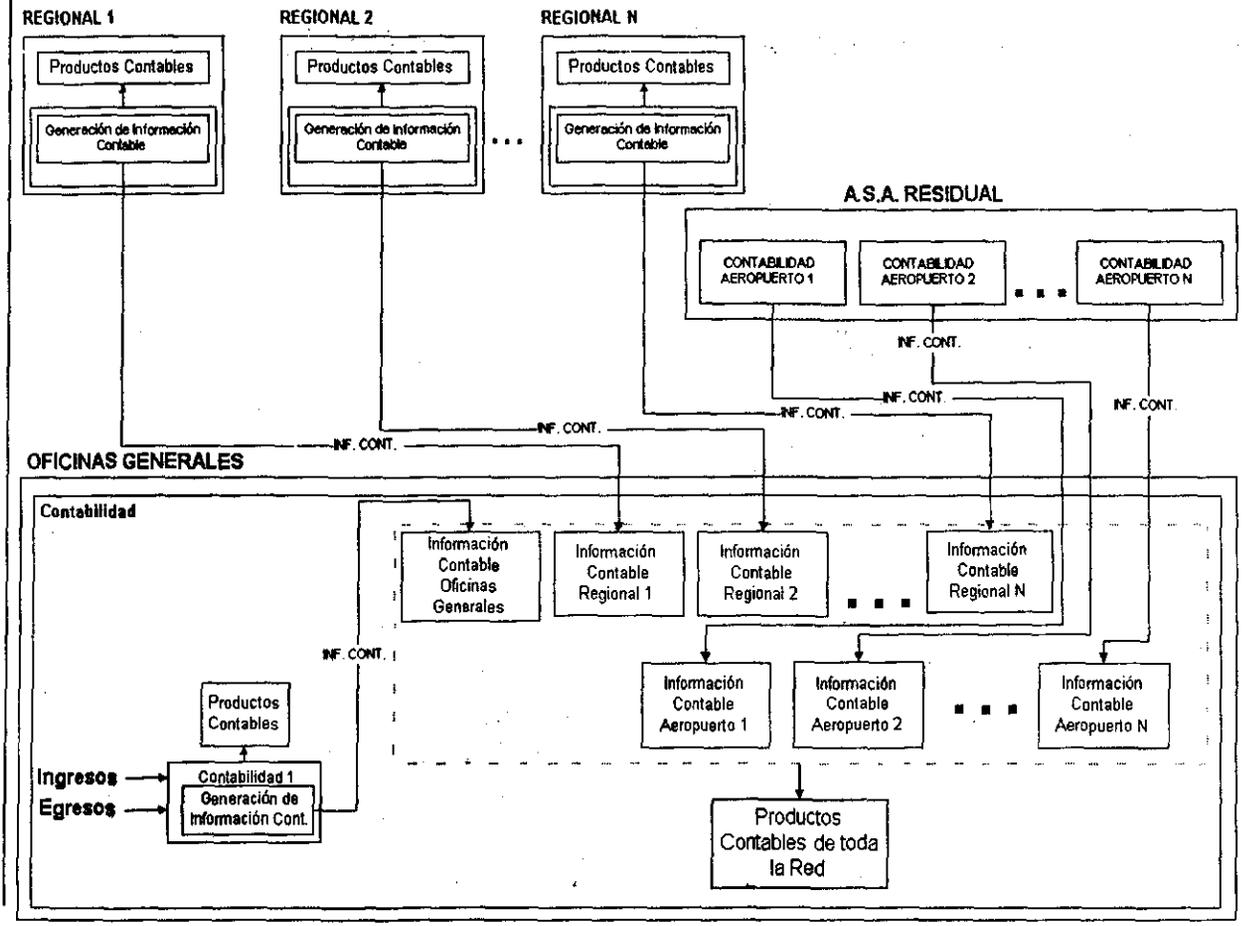


Figura 3.5.7. Esquema de manejo de las contabilidades por niveles (Aeropuertos, Regionales y Red Aeroportuaria).

Planteamiento del Problema y Propuesta de Solución

### CAPTURA DE PÓLIZAS

El primer elemento de información dentro de una contabilidad es una póliza, el módulo de captura de pólizas es el medio por el cual los movimientos contables contenidos en una póliza ingresan al sistema, en este módulo se tendrá cuidado especialmente en lo que a una captura rápida se refiere, el número de pólizas por mes puede oscilar entre las 1000 ó 1500, y tomando en cuenta que cada póliza está conformada de aproximadamente 20 movimientos, estamos hablando de un número considerable de información que debe de ser capturada, por lo cual debe de contar este módulo con ayudas en la captura a fin de poder realizar esta labor de manera ágil y sencilla, además de que su manejo debe de ser totalmente intuitivo, esto es que cualquier persona sin una capacitación a fondo del sistema pueda utilizar el módulo de captura de pólizas, esto es debido a que se contempla la posibilidad de contratar personal de captura para que realice esta labor.



Figura 3.5.8. El módulo de captura de pólizas es el medio por el cual ingresa la información al sistema.

### ADMINISTRACIÓN DE PÓLIZAS

Debido a los grandes volúmenes de información que manejará el sistema es necesario que haya un módulo que se encargue de la administración de las pólizas, que es la unidad contable básica de la contabilidad. Dentro de la administración de las pólizas englobamos la revisión y la autorización de una póliza, esto es que una persona captura una póliza y otra la revisa y por último otra la autoriza para su aplicación.

El módulo de administración es importante desde el punto de vista de que son las pólizas los elementos que nos generan todo el movimiento contable en el sistema, por esto, el sistema debe de contar con un manejo adecuado de estas, a fin de poder localizar, modificar, y borrar pólizas de manera rápida y confiable. No se debe de permitir el borrado de las pólizas que han sido ya contabilizadas, esto para no incurrir en faltas ante la Secretaría de Hacienda y Crédito Público que es la encargada de auditar la contabilidad de las empresas tanto públicas como privadas:

### APLICACIONES CONTABLES

Una vez que las pólizas contables se encuentran en el sistema es necesario realizar la afectación contable pertinente sobre las cuentas del catálogo. Es aquí en donde se prepara toda la información para que se puedan elaborar los reportes en el sistema. La aplicación contable debe de afectar todos los catálogos de todos los niveles hasta llegar al nivel más alto que ocupa el catálogo normativo. Esto nos permitirá tener la información contable actualizada en cualquier momento con lo cual cuando se pretenda generar un reporte no se tienen que hacer cálculos o actualizaciones que pueden tardar horas como en el caso de los sistemas que actualmente se encuentran trabajando, ya que para poder generar los reportes contables, se tienen que correr procesos que actualicen los saldos de las cuentas.

**ESQUEMA DE CONTABILIZACIÓN A NIVEL RED  
AEROPORTUARIA**

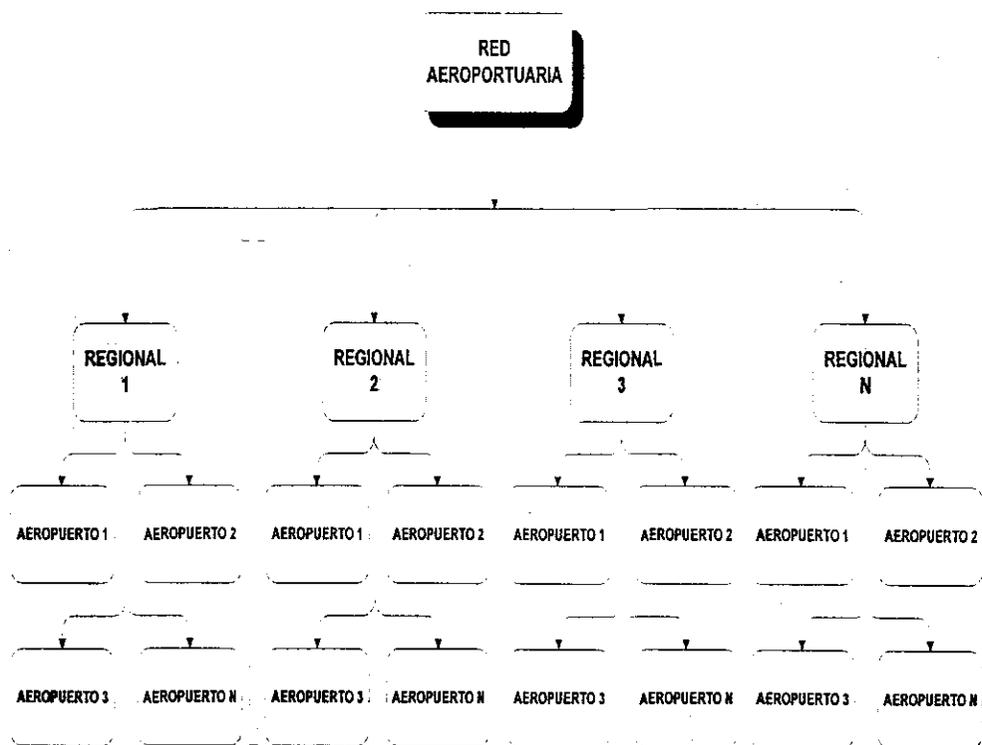


Figura 3.5.9. Al momento de aplicar una póliza la contabilidad a todos los niveles se actualizará.

## REPORTES

El módulo de reportes contendrá todos los reportes contables básicos como son el libro diario, el libro de mayor, los libros auxiliares, la balanza de comprobación y el estado financiero y de resultados. Este módulo quedará abierto a nuevas modificaciones debido a que estos reportes sólo son parte de una primera fase, después se implementarán otros reportes.

A pesar de que los reportes contables que se mencionan en el párrafo anterior tienen formatos bien establecidos a nivel mundial, es importante recalcar que tendrán la información presentada de la manera establecida por las Oficinas Generales, a fin de que la presentación sea la más satisfactoria para ellos, respetando claro está las convenciones tanto nacionales como internacionales en cuanto a teoría contable se refiere.

Algo muy importante que toman en cuenta los contadores es la posibilidad de que con el nuevo sistema deben de poder ellos mismo generar sus reportes, sin depender de una oficina en donde se concentre la impresión, esto con el fin de acelerar sus trabajos al tener la información en el momento en que la necesiten.

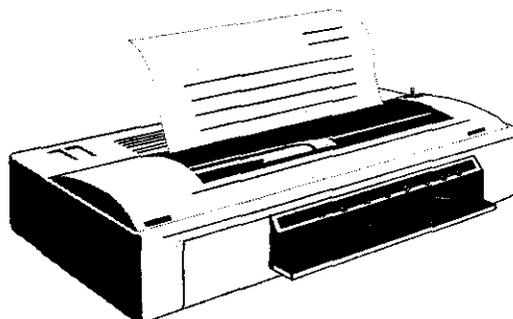


Figura 3.5.10. Con el nuevo sistema cada contador podrá generar sus reportes sin depender de una oficina de impresión.

### CIERRES CONTABLES

Cada mes todas las contabilidades tienen que cerrarse para poder evaluar la situación financiera de la empresa en ese momento; el cerrar una contabilidad es equivalente a no permitir aplicar más pólizas en un determinado mes, este módulo tendrá el control de los cierres que se hagan en las contabilidades de cada uno de los aeropuertos.

Uno de los problemas que tienen actualmente es que el sistema desarrollado en INFORMIX no contempla los cierres contables, y el otro que es el COI, al momento de cerrar la contabilidad, se deben de actualizar los saldos de las cuentas para el mes siguiente, y permite la contabilización de pólizas retroactivas sin llevar un control de la fecha de aplicación, esto es peligroso en el sentido de que no hay un control estricto de cuando se da un cierre contable, y se pueden emitir estados financieros que en realidad no nos representen el estado de la empresa, debido a que en cualquier momento pueden ser susceptibles de una modificación.

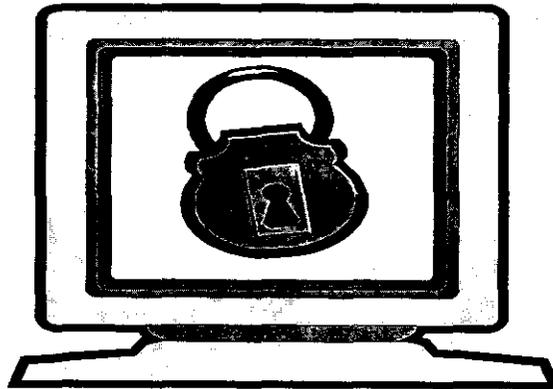


Figura 3.5.11. EL módulo de cierres contables controlará la aplicación de pólizas en meses en que la contabilidad ya se haya cerrado

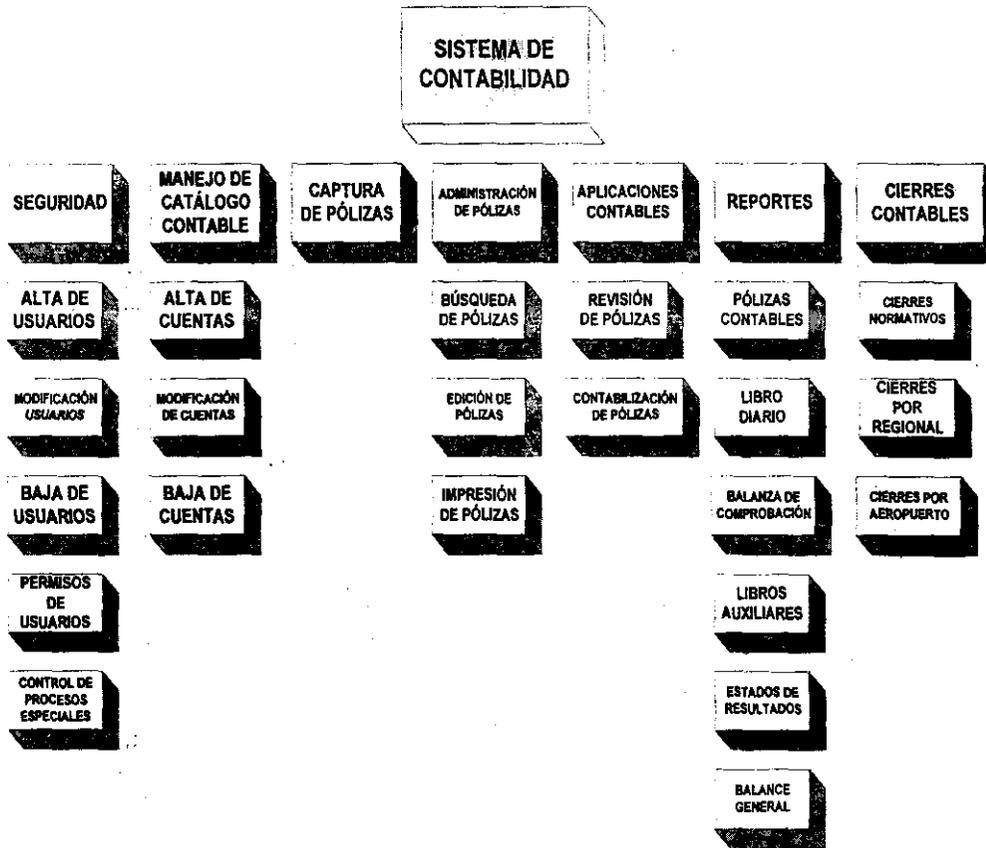


Figura 3.5.12. Diagrama de descomposición funcional del sistema de contabilidad

### **3.6. OPCIONES DE SOLUCIÓN**

El problema fundamental que se quiere resolver con la construcción de un sistema de contabilidad para la Red Aeroportuaria Nacional, es el facilitar el trabajo de los contadores que se encuentran en las Oficinas Generales y que realizan la tarea de llevar la contabilidad de todos los aeropuertos del país, para esto hay que determinar una vez que ya tenemos identificado el problema con cuantas posibles opciones de solución podemos contar para poder seleccionar la óptima.

Podemos enumerar de entrada tres soluciones que podrían resolver el problema:

1.- Continuar trabajando con el sistema que está realizado en informix y que actualmente se encuentra en proceso de desaparición ya que el mantenimiento se hace cada vez más costoso debido a que no se cuenta con la documentación necesaria acerca de los programas que conforman la aplicación además de otros factores.

Podemos justificar esta opción de solución diciendo que en pláticas iniciales con los usuarios, nos comentaban que cuando el sistema comenzó a funcionar, cubría en buena parte sus necesidades y les permitió acelerar su trabajo, aunque con el tiempo, las ventajas que les brindaba el sistema, se vieron sobrepasadas por las desventajas.

En términos de los programas actuales, esta adaptación consiste en optimizar el código de los programas existentes para disminuir los altos tiempos de repuesta que se presentan; incluyendo además las modificaciones necesarias para tener un buen funcionamiento del sistema en la red local, dado que constantemente se queda fuera de servicio.

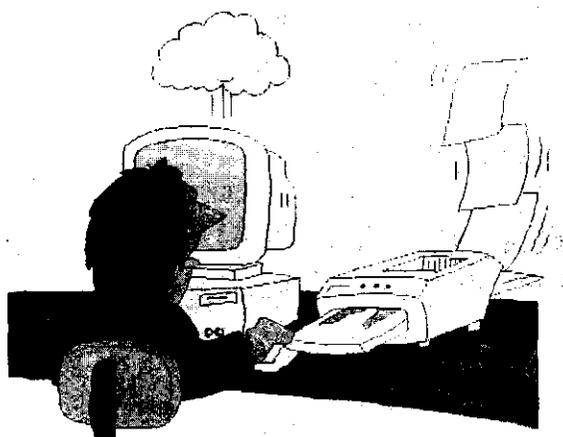
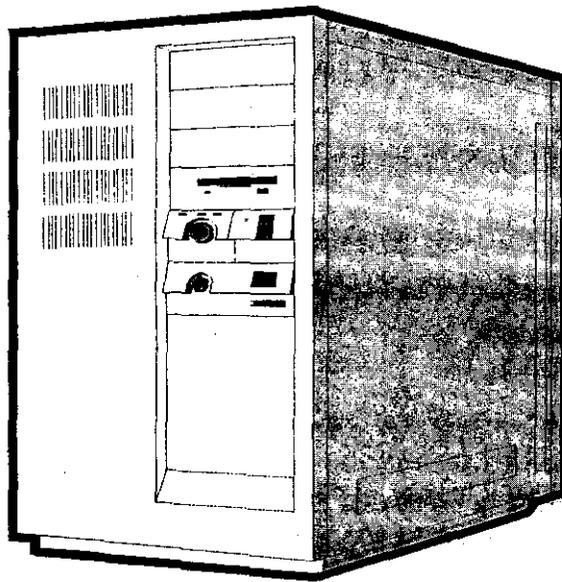


Fig. 3.6.1. Continuar con el sistema con el que actualmente se trabaja, traería consigo una labor ardua de mantenimiento

2.- Estudiar la forma de hacer que el sistema comercial COI pueda soportar una mayor cantidad de información y hacer que funcione en un ambiente multiusuario a fin de que los aproximadamente cincuenta contadores que trabajan en las Oficinas Generales puedan acceder a la información contable.

Habr  de contactar a los proveedores de este paquete y platicar detenidamente a fin de que nos muestren las ventajas que tendr  el seguir utilizando este paquete contable en la instituci3n.

De entrada podemos identificar algunas ventajas que podr  traer el seguir utilizando el sistema COI, como el no tener que desarrollar la parte de la aplicaci3n contable y la captura de p3lizas, es un sistema que puede manejar varias contabilidades, existe una versi3n para red, su interface al usuario es medianamente agradable y f cil de utilizar.

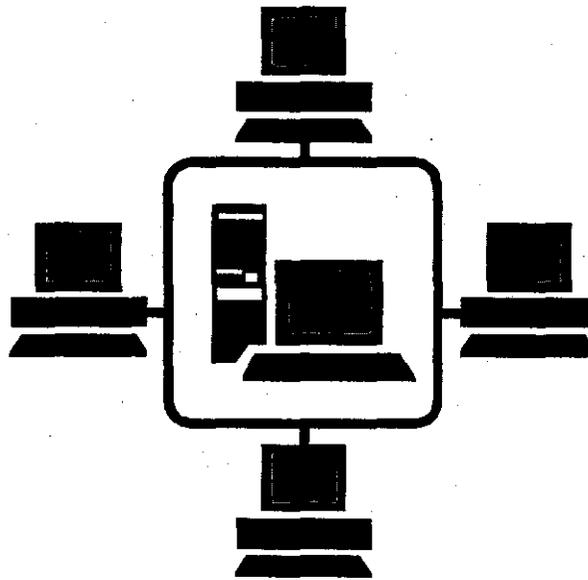


Figura. 3.6.2. Hacer que el sistema COI funcione en un ambiente multiusuario puede ser una de las opciones de soluci3n para el problema planteado.

3.-Desarrollar un nuevo sistema en un ambiente Cliente/Servidor que soporte grandes cantidades de información y un gran número de usuarios, además de poder soportar adecuaciones futuras.

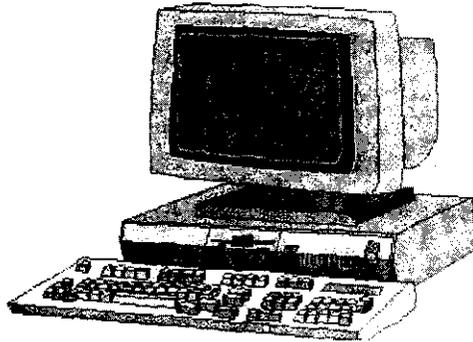
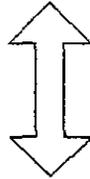
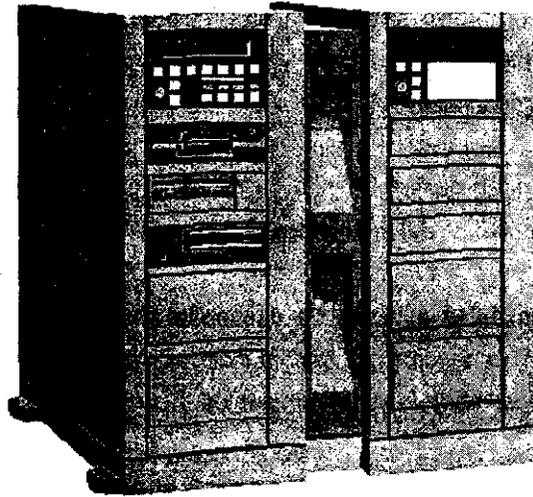
Con esto se podrían desarrollar nuevos módulos para el sistema a fin de poder resolver más problemas del área contable.

El utilizar herramientas actuales de desarrollo nos brindaría la oportunidad de manejar de manera sencilla grandes cantidades de información, además de que las interfaces con el usuario serían amigables.

Adecuaciones futuras serían posibles gracias a que existiría una documentación y en el mercado laboral se pueden encontrar personas que puedan desarrollar aplicaciones con herramientas actuales.

Se aprovecharían todas las ventajas que ofrece el desarrollar un sistema en ambiente Cliente/Servidor además de la utilización de un manejador de base de datos relacional.

La institución podría desarrollar otros módulos para otras oficinas que alimentaran directamente el sistema de contabilidad al momento de generarse una operación.



**Figura 3.6.3.** La creación de un sistema con arquitectura Cliente/Servidor es una opción de solución muy viable.

Las tres alternativas planteadas anteriormente son las opciones de solución más viables pero para poder determinar cuál es la mejor hay que hacer un estudio detallado de cada una a fin de estar seguros que la opción que se adopte es la mejor.

En el siguiente punto se analizarán las ventajas y desventajas de cada una de las opciones que se citaron a fin de poder determinar cuál es la opción más viable.

### 3.7. ELECCIÓN DE LA SOLUCIÓN ÓPTIMA

A continuación analizaremos las ventajas y desventajas de cada una de las opciones de solución que en el punto anterior se trataron.

#### 1.- Dar mantenimiento al sistema contable actual.

➤ Ventajas:

- Ninguna.

➤ Desventajas:

- No se cuenta con la documentación técnica de la aplicación.
- No se sabe la ubicación exacta de todos los programas fuentes.
- El personal que desarrolló el sistema no labora en la institución actualmente.
- Algunas partes del sistema ya no funcionan.
- El lenguaje en el cual se desarrolló el sistema es un lenguaje obsoleto y en el mercado no se encuentran desarrolladores que conozcan de forma fluida su manejo.

2.- Manejo del sistema contable comercial COI con el que cuenta la empresa actualmente.

➤ **Ventajas:**

- El sistema se encuentra totalmente construido y puede manejar la contabilidad de una empresa.
- Se puede comprar la versión para 99 empresas en ambiente de red.

➤ **Desventajas:**

- La versión para 99 empresas puede solucionar en parte el problema del manejo de la contabilidad de todos los aeropuertos de la Red, pero esta versión maneja las contabilidades de forma separada, de tal forma que la obtención de informes contables consolidados de toda la red sería una tarea que debe de hacerse de forma manual.
- Los formatos de los reportes son en extremo pobres en cuanto a presentación se refieren.
- No existe forma de controlar una pérdida de información en caso de que el equipo en donde resida la información contable falle. La única forma es realizando respaldos periódicos de los datos.
- Los formatos de los reportes no cumplen plenamente con las expectativas de información de los usuarios.
- Una adecuación del sistema por parte de un equipo de desarrolladores de la institución es imposible, debido a que no se cuenta con los programas fuentes y los datos tienen un formato propio. Las únicas adecuaciones o complementaciones al sistema se hacen comprando los módulos que el mismo fabricante ofrece.

### 3.- Creación de una nueva aplicación en un ambiente Cliente/Servidor

➤ **Ventajas:**

- El desarrollo del sistema se haría en base a un análisis de requerimientos con el cliente.
- La aplicación cubriría totalmente las expectativas del usuario.
- Utilizando alguno de los manejadores de bases de datos que actualmente se encuentran en el mercado y que ofrecen múltiples ventajas con respecto al manejo de información, se podrían manejar de forma muy eficiente toda la información contable de la institución y se tendrían esquemas de seguridad de los datos.
- Desarrollando la aplicación en un lenguaje de programación actual se puede contar con el personal adecuado que pueda construir y mantener el sistema.
- Adecuaciones posteriores o integración de nuevos módulos pueden realizarse.
- Se contaría con los manuales técnicos del sistema.
- Las interfaces con el usuario serían estéticas y fáciles de manejar.

➤ **Desventajas:**

- En caso de que la institución no cuente con las herramientas de desarrollo se tendría que realizar la compra de dichas herramientas.
- Se tendría que capacitar en el manejo de las herramientas de desarrollo al grupo desarrollador de software de la institución.

Hasta este momento se han enumerado las principales ventajas y desventajas de las opciones de solución, pero para poder tomar una decisión hay que ahondar con respecto a la última, ya que la construcción de una arquitectura cliente servidor trae consigo la determinación de las herramientas adecuadas para el desarrollo de la aplicación. Las herramientas que se deben de elegir al momento de desarrollar un sistema basado en esta arquitectura son las herramientas llamadas servidores de

bases de datos y las de desarrollo de clientes o front-ends, a continuación se hace un estudio de las herramientas más comunes.

Un sistema de procesamiento de datos consta, básicamente, de cuatro componentes: la máquina donde se ejecuta, los programas, los datos y la gente que utiliza el sistema. Los términos utilizados para las máquinas y los programas son hardware y software, respectivamente.

El gran advenimiento de la invención y desarrollo de las computadoras a partir de la década de los 40's, dio como resultado que los grandes consorcios y empresas en general, respondiesen a este advenimiento computacional; de esta forma, adquirieron e instalaron diversidad de equipos, se dieron a la tarea de capacitar programadores y analistas, y además, automatizaron diversos procesos de oficina. Aunado a lo anterior, se desarrollaron sofisticados algoritmos para hacer más vastos los alcances de las nuevas máquinas; se generaron nuevos lenguajes para optimizar el uso de las anteriores. Dado lo anterior, la industria del cómputo y las empresas usuarias han hecho diversos esfuerzos para desarrollar mejores técnicas de programación. Los lenguajes de alto nivel han sido desarrollados a un grado tal que sin importar la máquina donde se encuentren, éstos se ejecutan.

El sistema a desarrollar, será capaz de manejar una gran cantidad de información, la cual está dividida en diversos conceptos. Para facilitar su manejo, es necesario la conjunción de ésta, por medio de una base de datos.

La base de datos contendrá toda la información requerida para llevar la contabilidad de todos los aeropuertos de la red aeroportuaria nacional.

Todas aquellas personas que hacen uso de un sistema son llamadas usuarias. Los usuarios son clasificados en programadores y no programadores. Una vez que un DBMS (Sistema Manejador de Bases de Datos) es hecho, la habilidad para que

crezca y se desarrolle con vastedad, depende de la habilidad del programador para extender y modificar los componentes del sistema.

A continuación, se da una clasificación de los usuarios.

Usuarios no programadores:	Usuario Casual
	Usuario Paramétrico
	Usuario General
Usuarios programadores:	Programador de la Base de Datos
	Programador de Aplicaciones Convencionales

El usuario casual interactúa con el DBMS irregularmente, por lo tanto el sistema debe proporcionar un diálogo para ayudar al usuario y minimizar los errores de uso. el usuario paramétrico interactúa con el sistema en rutas preestablecidas y restringidas, llamando solicitudes ya almacenadas y dando los parámetros necesarios para llevar a cabo esa solicitud. El usuario general interactúa en rutas libres, por lo cual necesita un lenguaje de alto nivel que sea flexible.

En este sistema, se deberá contar con rutas preestablecidas, con la finalidad de que el usuario que haga uso del mismo, no tenga que ser todo un experto en llevar la contabilidad de los aeropuertos nacionales. El sistema conducirá al usuario de una forma amigable, y en caso de que éste requiera ayuda, se brindará de una forma sencilla de consultar.

Con las facilidades que proporcione el DBMS seleccionado, se tendrán que suministrar las ayudas necesarias para que el usuario no tenga problemas en el uso del sistema. Dado lo anterior, estamos en las condiciones de decir que la mejor manera de organizar la información es a través de una Base de Datos Relacional, para contar con archivos independientes donde sea contenida la información, sin mezclar campos o repetir información en cada archivo y teniendo relaciones entre

éstos, a través de campos claves, para poder accederlos y llevar a cabo la contabilidad de la manera más sencilla, contando con el acceso a todos los archivos que se estén manejando, optimizando el uso de la computadora.

Considerando los requerimientos del usuario se llevó a cabo el análisis de algunas herramientas de desarrollo.

➤ HERRAMIENTAS DE DESARROLLO

➤ Servidores de bases de datos. A continuación se analizan algunas de las características más importantes de los manejadores de bases de datos más difundidos actualmente en nuestro país.

➤ ORACLE:

Esta base de datos corre sobre más hardware, software y redes que cualquier otra base de datos relacional. Esta particularidad lo identifica como un producto muy portable. Oracle es soportado por más del 70% de los back-ends y front-ends que hay en el mercado.

Soporta el manejo de Triggers y Store procedures. Cuenta con opciones de seguridad para el manejo de la base de datos. Maneja los esquemas de discos espejos. Cuenta con un log (registro) de transacciones, en donde se almacenan las modificaciones que se hagan a la base de datos.

EL manejador de bases de datos relacionales Oracle fue uno de los primeros productos en su tipo y actualmente se encuentra instalado en un gran número de aplicaciones.

Este producto esta diseñado para proveer todas las facilidades de almacenamiento y recuperación de información en diferentes formatos.

Los datos almacenados son almacenados en una serie de archivos en los discos de la computadora en donde se encuentre corriendo Oracle.

Puede trabajar en un ambiente multiusuario, soportando un gran número de usuarios.

Para aumentar la velocidad de procesamiento y la consistencia de los datos, implementa el uso de grandes áreas de memoria para almacenar datos, además de que utiliza transacciones.

Tiene desarrollados un gran número de procesos que elevan el desempeño de las rutinas de almacenamiento y recuperación de datos.

Sus rutinas de almacenamiento y recuperación de datos aprovechan las ventajas de las computadoras que cuentan con varios procesadores.

Garantiza la integridad de los datos, llevando un control de los registros que han sido modificados dentro de una transacción a fin de poder deshacer todos los cambios en caso de que por algún motivo el proceso no se termine completamente.

Soporta los esquemas de discos espejo para poder continuar con los procesos en caso de que un disco se dañe.

Este manejador de base de datos es soportado por un gran número de lenguajes de cuarta generación, con lo cual se pueden desarrollar de forma rápida aplicaciones que impliquen un gran volumen de información.

Existen en el mercado diversas herramientas para el manejo de Oracle, que facilitan las tareas de desarrollo de aplicaciones.

Se puede implementar de manera sencilla integridad referencial.

Con respecto a los procedimientos almacenados que mencionamos al principio podemos decir que cuenta con una serie de procedimientos de ayudas que facilitan el trabajo de desarrollo.

Existen un gran número de herramientas de modelado de base de datos que soportan este producto y que generan completamente el código para la construcción de la base de datos.

#### ➤ INFORMIX

Informix Online opera bajo sistemas operativos UNIX, proporciona alto performance y alta disponibilidad de información. Incluye características como discos espejo y mecanismos ágiles de recuperación de información. Es por ello que con frecuencia se utiliza en aplicaciones de misión crítica.

Puede trabajar bajo la plataforma cliente/servidor mediante el producto de conectividad INFORMIX NET. Este soporta los protocolos TCP/IP y StartGroup, y una variedad de computadoras y hardware de comunicación de datos como Ethernet, Token Ring y Starlan.

Cuenta con un conjunto de tablas para registrar el diccionario de datos. En el caso particular del manejador de bases de datos OnLine se tiene la posibilidad de guardar en forma histórica las modificaciones hechas al diccionario de datos.

Soporta bases de datos distribuidas mediante el producto de conectividad INFORMIX STAR y el manejador de base de datos OnLine. Mediante estos productos se puede consultar y modificar varias bases de datos INFORMIX, en diferentes máquinas, como si fuera una sola base de datos centralizada.

Cuenta con recuperación de bases de datos de forma automática en casos de interrupciones por fallas de hardware o de energía. También se puede contar con una copia de la base de datos en un disco diferente, a fin de que si el disco primario queda fuera de línea, el sistema se cambia al disco espejo en forma transparente para los usuarios.

Soporta el uso de Triggers y Store Procedures.

#### ➤ SYBASE

El SQL Server de Sybase es una arquitectura multihilos, con su propio kernel que tiene integrado un monitor de control de Sql.

Parte de Sybase, Transact SQL es un conjunto de extensiones para ANSI-estándar SQL que es usado para escribir stored procedures.

Es posible construir triggers que se utilizan, entre otras cosas, para garantizar la importante propiedad de las bases de datos, denominada integridad referencial.

El producto integra funciones de seguridad que permiten a los administradores de sistemas, controlar el personal que tendrá acceso a la base de datos y el nivel de autoridad con que lo hará.

Se utiliza un log de transacciones, en el cual se escribirá cualquier modificación de la que sea objeto la base de datos.

Como parte de su seguridad, Sybase proporciona la opción de generar discos espejos.

EL API de SQL Server permiten a los desarrolladores de software construir aplicaciones cliente, que trabajen con SQL Server y con una interfaz transparente. DB-Library es un conjunto de funciones creadas en C, que permiten realizar comandos de SQL para recuperar y actualizar datos. Las rutinas de DB-Library pueden ser usadas con C o Cobol. DB-Library es una biblioteca con ligas estáticas bajo DOS y ligas dinámicas bajo Windows y OS/2.

Sybase SQL Server tiene más herramientas de terceros que cualquier otra base de datos, por lo cual, goza de popularidad y es considerada como una de las líderes,

Cuenta con un optimizador de consultas, el cual analiza las sentencias SQL que el usuario introduce y si la consulta que el usuario está realizando no es la óptima, genera una nueva consulta que de manera más eficiente obtenga los datos requeridos.

El servidor se encarga de garantizar la integridad referencial de los datos.

Cuenta con un servidor de respaldos, el cual corre junto con el servidor SQL, a fin de brindar un alto desempeño en respaldos de bases de datos, cargas, y recuperaciones.

Cuenta con herramientas tales como editores de archivos de configuración, probadores de comunicación para testificar la comunicación entre el servidor y los clientes, editores para la creación de procedimientos con el lenguaje Transac-SQL.

Cuenta con herramientas para monitoreo de procesos en el servidor.

Cuenta con un gran número de procedimientos almacenados de ayuda para trabajar en la base de datos.

Es uno de los productos con más alto desempeño al correr en aplicaciones de tiempo real.

Es un manejador de bases de datos que puede acceder información en otras bases de datos con otros formatos.

Cuenta con herramientas muy avanzadas de replicación de datos. Los esquemas de replicación son muy seguros y eficientes.

➤ **Herramientas de desarrollo de clientes.**

Dentro de la industria del software existen dos tipos de lenguajes de programación, los cuales son:

**Constructores de Soluciones**

**Constructores de Componentes**

**CONSTRUCTORES DE SOLUCIONES:**

Los constructores de soluciones son programas que se caracterizan por tener un contacto más directo con los usuarios que utilizarán el software, por lo cual conocen más las necesidades. Éstos, usualmente, construyen soluciones de negocios, se caracterizan por la habilidad con que trabajan con los usuarios, hasta refinar sus requerimientos.

## CONSTRUCTORES DE COMPONENTES

Los constructores de componentes se caracterizan por no tener ningún contacto con los usuarios. Para tratar de refinar los requerimientos del mismo, se enfocan básicamente al hardware, y no a las consideraciones de negocios.

Un grupo, que juega un papel importante, son los proveedores de software. Se les podría considerar como una extensión de los constructores de componentes.

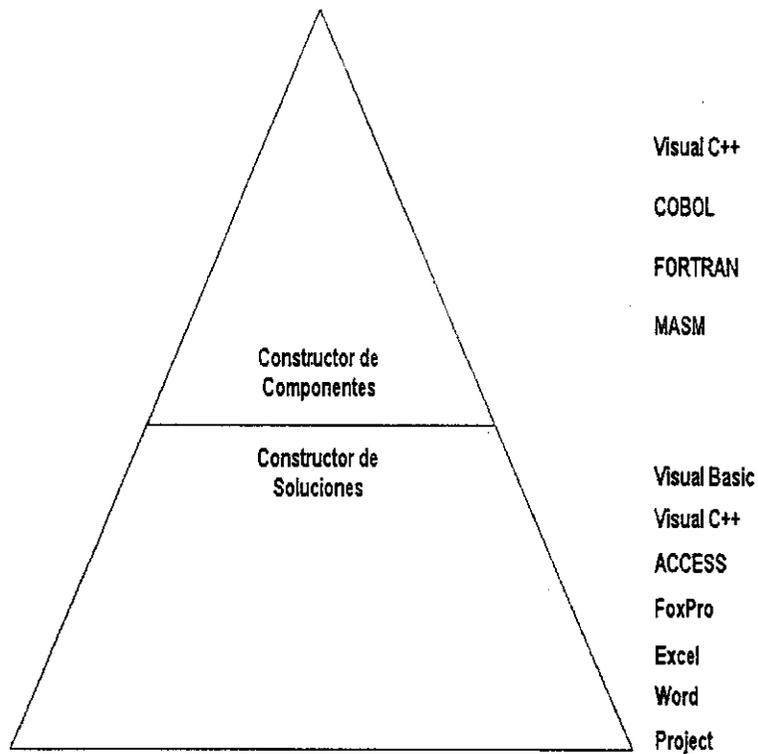


Figura 3.7.1. Constructores de componentes y constructores de soluciones.

GRUPO	ENFOQUE	CONJUNTO DE APTITUDES	CARACTERÍSTICAS DE LAS HERRAMIENTAS	HERRAMIENTAS TÍPICAS
Constructores de Soluciones	Definición de soluciones y explotación de componentes.	Orientado a soluciones de negocios: Relación con el usuario final; comprensión profunda del problema.	Iterativas, buenas como constructoras de interfaces y unión de componentes.	Power Builder Visual Basic Visual C++ Access Microsoft FoxPro Microsoft Excel Microsoft Word Microsoft Project
Constructores de Componentes	Construcción de componentes reutilizables.	Orientado a la máquina: Métodos de desarrollo tradicionales; dominio del análisis.	Producción rápida y compacta de código; pruebas rigurosas.	Visual C++ Cobol Fortran MASM
Proveedores Externos	Vendedor de Componentes	Desarrollo de mercadeo de componentes genéricos de alta calidad y de industrias específicas.	Producción de código de alta calidad.	

Figura 3.7.2. Tabla de características de constructores de componentes y soluciones.

➤ VISUAL BASIC

Este lenguaje está disponible para programadores de MS-DOS y Windows. Visual Basic es una herramienta rápida para crear aplicaciones generales de negocios. Cuenta con una pequeña curva de aprendizaje, requiere pocos conocimientos de programación y tiene una rápida integración y rehusos de componentes.

Proporciona gran flexibilidad para llevar a cabo llamadas a archivos con extensión DLL, y agregar una gran cantidad de controles, al gusto del cliente.

Visual Basic constituye un camino fácil para crear interfaces de usuario, utilizando componentes preconstruídas, por lo que se vuelve un lenguaje flexible que cuenta con un editor-verificador de sintaxis y herramientas de perfeccionamiento (debugging).

VENTAJAS

- Es un sistema productivo para crear soluciones en Windows.
- Controles visuales preconstruídos por terceros.
- Acceso a Bases de Datos.
- Manejo de Multimedia.
- Permite un ensamblaje fácil y rápido entre la interface de un usuario y los componentes prefabricados.
- Ofrece una gran capacidad y velocidad en su debugger sofisticado.
- Permite manipular otras aplicaciones para utilizarlas como componentes en aplicaciones propias (Ej. Word, Excel, Project, etc.), siempre y cuando dichas aplicaciones soporten OLE automation.

- Soporta variedad de Manejadores de Bases de Datos.
- Interface al usuario amigable.
- Manejo de ayudas en línea robustas.
- Intérprete poderoso que permite la detección de errores al momento de la edición del programa.
- Aplicaciones escritas que toman ventaja de grandes arreglos y un espacio ilimitado en strings (cadenas).
- Maneja aplicaciones Cliente-Servidor.
- Incluye como base de datos nativa a Access, la cual provee acceso simultáneo con FoxPro, Dbase, Paradox, SQL, etc.
- Un usuario inexperto puede ser productivo con algunas semanas de utilización de Visual Basic.
- Puede ser utilizado como un servidor de OLE Automation.

#### DESVENTAJAS

- Visual Basic se tiene que auxiliar de lenguajes más robustos, como "C" o Pascal, para hacer manipulación de memoria, accesos directos al Sistema Operativo, etc.
- No cuenta con la característica de herencia, que es muy importante dentro de la programación orientada a objetos (POO).
- Su intérprete es lento en la creación de ejecutables.
- Su código no es compatible , en plataformas diferentes a Windows.

#### ESPECIFICACIONES TÉCNICAS DE DESARROLLO

- Requiere de una computadora personal, que cuente con Windows 3.1 o una versión mayor.
- Al menos 4MB en memoria (8 MB o más es la recomendada).
- Disco duro con capacidad mínima de 30 MB.
- Monitor VGA o de una resolución mayor.

- Mouse.

#### ESPECIFICACIONES TÉCNICAS DEL USUARIO

- Requiere de una computadora personal, que cuente con Windows 3.1 o una versión mayor.
- Al menos 4MB de memoria disponible en RAM.
- Disco duro con capacidad mínima de 10 MB.
- Configuración mínima 486.
- Mouse.

A continuación, se muestran algunas de las ventajas anteriormente citadas.

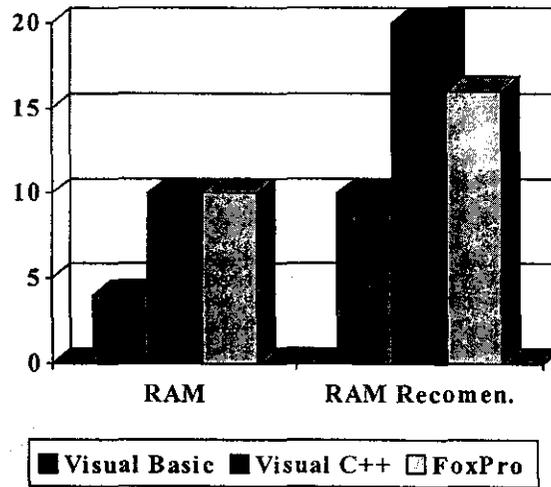


Figura 3.7.3. Requerimientos de Memoria RAM.

➤ VISUAL C++

Es una herramienta poderosa y flexible, para generar soluciones de negocios, además de ser una herramienta rápida para programadores que utilizan C y C++.

VENTAJAS

- Incluye gran variedad de librerías preconstruidas.
- Controles visuales.
- Permite la reutilización total del código.
- Manejo completo del esquema de la programación orientada a objetos (herencia, polimorfismo, etc.).
- Maneja controles VBX (Visual Basic eXtension), los cuales son componentes preconstruidos que aceleran el desarrollo de software, compatibles con la Ver, 1.0. de Visual Basic.
- Todo el manejo de dispositivos de Entrada/Salida (I/O), se hace a través de librerías.
- Cuenta con las características de un Visual Workbench (integra accesos a herramientas, como librerías, compiladores, editores, etc.; control de proyectos, etc.).
- Cuenta con compilador, no con intérprete.
- Se pueden crear clientes OLE, a través de él.
- Es el único lenguaje que, al salir una nueva versión de Windows, ya cuenta con las herramientas necesarias para trabajar en él.
- Maneja Multitarea.
- Lenguaje poderoso y flexible, ya que permite crear aplicaciones núcleo.
- Rapidez en ejecución de programas.
- Manejo de variables locales y globales.

- Permite obtener funcionalidades de otras aplicaciones, para utilizarlas como componentes en aplicaciones propias (Ej. Word, Excel, Project, etc.).
- Manejo de controles OLE.
- Libertad total para el manejo de memoria y accesos a sistema operativo.
- Permite la creación de servidores de OLE.

#### DESVENTAJAS

- Curva de aprendizaje alta.
- Tiempo de compilación largos (superable en nuevas versiones).

#### ESPECIFICACIONES TÉCNICAS DE DESARROLLO

- Requiere de una computadora personal, que cuente con Windows 3.1 o una versión mayor.
- Requiere de una computadora personal, que cuente con Windows NT o Windows 95.
- Al menos 8 MB en memoria (20 MB o más es la recomendada).
- Disco duro con capacidad mínima de 60 MB.
- Monitor VGA o de una resolución mayor.
- Mouse.

#### ESPECIFICACIONES TÉCNICAS DEL USUARIO

- Requiere de una computadora personal, que cuente con Windows 3.1 o una versión mayor.
- Al menos 4MB de memoria disponible en RAM.
- Disco duro con capacidad mínima de 10 MB, sin contar con el programa generado.

- Configuración mínima 486.
- Mouse.

A continuación, se muestran algunas de las ventajas anteriormente mencionadas.

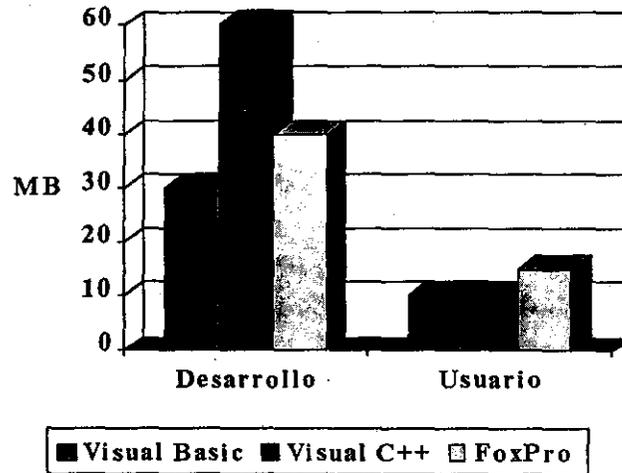


Figura 3.7.4. Requerimientos de Capacidad de Disco Duro.

- FOXPRO

FoxPro es el administrador de bases de datos más rápido para computadoras personales, permite hacer consultas para manejar grandes cantidades de datos.

#### VENTAJAS

- Maneja comandos de Xbase.

- El modelo de la programación orientada a objetos está completamente inmersa, es decir, se maneja polimorfismo, herencia, etc.
- Soporta el manejo Multiusuarios.
- Alta velocidad en la ejecución de consultas (queries).
- Las aplicaciones hechas en DOS, son fácilmente transportables a Windows.
- Los datos de FoxPro pueden ser exportados a ASCII, DBaseIII, WK1, WR1 y Excel.
- Permite explotar la tecnología OLE.
- FoxPro es una herramienta poderosa para el manejo de datos masivos.
- Cualquier manejo de interface al usuario, como formas y controles, pueden utilizarse como base para crear una nueva clase.
- Cuenta con un ambiente altamente integrado.
- Es una herramienta Multiplataforma.
- Permite la conexión a diferentes Bases de Datos.
- Permite la utilización de herramientas OLE.

#### DESVENTAJAS

- No puede ser utilizado como un servidor de OLE Automation, es decir, no puede manipular aplicaciones.
- Utilización de intérprete, en vez de compilador.
- FoxPro es poco flexible, ya que se tiene que apoyar en lenguajes más robustos como "C" o Pascal, para hacer manipulación de memoria, accesos directos al Sistema Operativo, etc.

## ESPECIFICACIONES TÉCNICAS DE DESARROLLO

- Requiere de una computadora personal, que cuente con Windows 3.1 o una versión mayor.
- Al menos 8 MB en memoria (16 MB es la recomendada).
- Disco duro con capacidad mínima de 40 MB.
- Monitor VGA o de una resolución mayor.
- Mouse.

## ESPECIFICACIONES TÉCNICAS DEL USUARIO

- Requiere de una computadora personal, que cuente con Windows 3.1 o una versión mayor.
- Al menos 8MB de memoria disponible en RAM.
- Disco duro con capacidad mínima de 15 MB.
- Configuración mínima 386.
- Mouse.

## ➤ POWER BUILDER

## VENTAJAS

- Permite el desarrollo de aplicaciones robustas, bajo ambientes multiplataforma, cuenta con un optimizador de código, posibilidad de distribución de objetos en un ambiente de red y manejadores nativos para diferentes bases de datos.
- Posee un soporte completo para ambiente Windows de 16 y 32 bits en plataformas Intel, incluyendo Microsoft Windows 3.X, Windows NT, Windows 95, Win OS/2, Mac y UNIX (Modify Open Look):
- Posee una familia de herramientas de desarrollo escalable que incrementan la productividad de las aplicaciones. La serie incluye

Power Builder Enterprise, Power Builder Team/PDBS, Power Builder Desktop, Infomaker y Power Builder Library for Lotus Notes.

- Su reporteador Infomaker, goza de versatilidad permitiendo múltiples tipos de reportes sin demasiada programación. Las consultas se realizan a través de un constructor gráfico y un quickselect multitabla. Sólo hay que salvar las consultas como objetos y entonces usarlos como fuentes de datos para una gran variedad de reportes.
- Es capaz de definir, compilar y corregir clases integradas de C++ basadas en el compilador Watcom C/C++ de alto rendimiento.
- Se ofrece un extenso lenguaje orientado a objetos con miles de funciones. Los desarrolladores pueden escribir sus propias funciones o utilizar las ya existentes escritas en C o en otros lenguajes. Han sido incluidos un compilador manejador y un debugger con muchas particularidades positivas.
- Incluye toda una gama de programación orientada a objetos, pero en Watcom C++ y en 3GL . PowerBuilder soporta la definición de clases para modelados visuales y objetos no visuales. Incluye herencia, encapsulamiento de datos y procesos lógicos y polimorfismo.
- Con respecto a la programación orientada a eventos, podemos decir que se basa en los eventos de Windows en su totalidad.
- Se incluye una biblioteca de objetos centralizada y un administrador de código fuente.
- Soporta desarrollo en grupo.
- El código se separa del lenguaje de transacción, optimizando la programación y el aprendizaje de la herramienta.
- La migración entre servidores SQL es fácil.
- Se pueden crear bases de datos desde PowerBuilder.
- Soporta OLE 2.0 a nivel servidor y cliente.
- Tiene un amplio manejo de gráficos de dos y tres dimensiones.

## DESVENTAJAS

- Tiene un ambiente de programación diferente al normal.

## ESPECIFICACIONES TÉCNICAS DE DESARROLLO

- Requiere de una computadora personal, que cuente con Windows 3.1 o una versión mayor.
- Al menos 4MB en memoria (8 MB o más es la recomendada).
- Disco duro con capacidad mínima de 30 MB.
- Monitor VGA o de una resolución mayor.
- Mouse.

## ESPECIFICACIONES TÉCNICAS DEL USUARIO

- Requiere de una computadora personal, que cuente con Windows 3.1 o una versión mayor.
- Al menos 4MB de memoria disponible en RAM.
- Disco duro con capacidad mínima de 10 MB.
- Configuración mínima 486.
- Mouse

Basándonos en el análisis elaborado anteriormente se deben de seleccionar las herramientas de desarrollo adecuadas para poder cumplir con las expectativas del cliente.

Debe asegurarse que las necesidades del usuario sean cubiertas de manera general, es decir, la solución no sólo debe de satisfacer las necesidades de un

determinado grupo de usuarios. También deben de tomarse en cuenta los siguientes puntos antes de elegir la solución óptima.

### TECNOLOGÍA

Desde el punto de vista técnico se debe de buscar que la solución a la necesidad de automatizar el manejo de la contabilidad de los aeropuertos este montada en una infraestructura de nueva tecnología, el pensar en desarrollar un sistema de tal magnitud en una tecnología que ya es obsoleta o que en poco tiempo lo será, es una decisión errada. Al pensar en tecnologías nuevas sabemos que se encontrarán las herramientas en el mercado con nuevas innovaciones, además de que se contará en todo momento con personal capaz de dar mantenimiento a nuestro sistema.

### HERRAMIENTAS DE APOYO

El iniciar el desarrollo de un nuevo sistema, no quiere decir que se tenga que empezar de cero, siempre en el mercado existen lenguajes de programación y manejadores de bases de datos que nos ofrecen un sinnúmero de ayudas que facilitarán nuestro trabajo y sobre todo acortarán los tiempos de desarrollo. Es importante recalcar que nuestra herramienta con la cual vamos a construir el front-end debe de contar con funciones ya construidas, además de manejo de ventanas y menús a fin de evitar que seamos nosotros los que desarrollemos estas utilerías.

### INNOVACIÓN

En el mundo de competencia en el cual vivimos, es importante darle al sistema que desarrollemos un aspecto innovador, a fin de que pueda competir con una gran cantidad de sistemas que existen en el mercado, pero el darle este aspecto, no debe de llevarnos demasiado tiempo y el costo de esto debe ser bajo, esto se

logra eligiendo una herramienta que cuente con características que de manera sencilla y rápida nos permita introducir al sistema aspectos con los cuales no cuenten los sistemas que se encuentren en el mercado.

#### INTERFACE AMIGABLE

El manejo de una contabilidad es un proceso ordenado el cual está conformado de pasos bien establecidos, los cuales deben de mostrarse de manera clara en el sistema a fin de que el tiempo de aprendizaje por parte de los usuarios no sea demasiado largo y el usuario sienta que el sistema es una ayuda y no una forma de hacer más complejo su trabajo.

#### RELACIÓN COSTO-BENEFICIO.

La plataforma de desarrollo y los recursos mínimos de operación para la aplicación muchas veces dependen de las herramientas en que ha sido desarrollada, es por ello que debemos de tomar en cuenta que existen usuarios con una infraestructura de equipos bastante aceptable y para los cuales se podría desarrollar una aplicación eficiente que consuma bastantes recursos, sin que ello represente un problema para estos usuarios, pero no hay que perder de vista a aquellos usuarios los cuales realizan los mismos procesos y no cuentan con equipos adecuados. Tomando en cuenta esto se debe de seleccionar una plataforma de desarrollo con la cual se pueda trabajar con el equipo con en cual cuenta actualmente la red aeroportuaria.

#### TIEMPOS DE RESPUESTA

Sea cual fuere la herramienta de desarrollo que se elegirá, es importante apuntar que el tiempo de respuesta siempre es un factor crítico que hace que una aplicación sea aceptada por el usuario o sea rechazada.

## COMPATIBILIDAD

La mayoría de los usuarios hoy en día además de utilizar algún sistema, manejan otro tipo de programas y aplicaciones que requieren para realizar su trabajo cotidiano, y que a pesar de utilizar un nuevo sistema, no dejarán de utilizar por diversas razones, este es el motivo por el cual el desarrollador tiene que asegurar que el sistema pueda en determinado momento trabajar con otros sistemas ya sea para obtener información de ellos o para proporcionárselas.

## PLATAFORMA DE DESARROLLO

Con respecto a la plataforma de desarrollo debemos analizar que la plataforma óptima para poder desarrollar este sistema, es aquella que en un futuro pueda estar en cualquier aeropuerto que conforma la red, por lo cual, para escoger nuestras herramientas de desarrollo debemos de tomar en cuenta que en cualquier aeropuerto se cuenta o se puede contar de manera inmediata con microcomputadoras Pentium con sistema operativo Windows 95, y servidores HP9000, por lo cual las herramientas de desarrollo, tanto para el front-end y la base de datos deben de ser soportadas en este tipo de plataformas.

## AMBIENTE OPERATIVO PARA LA APLICACIÓN

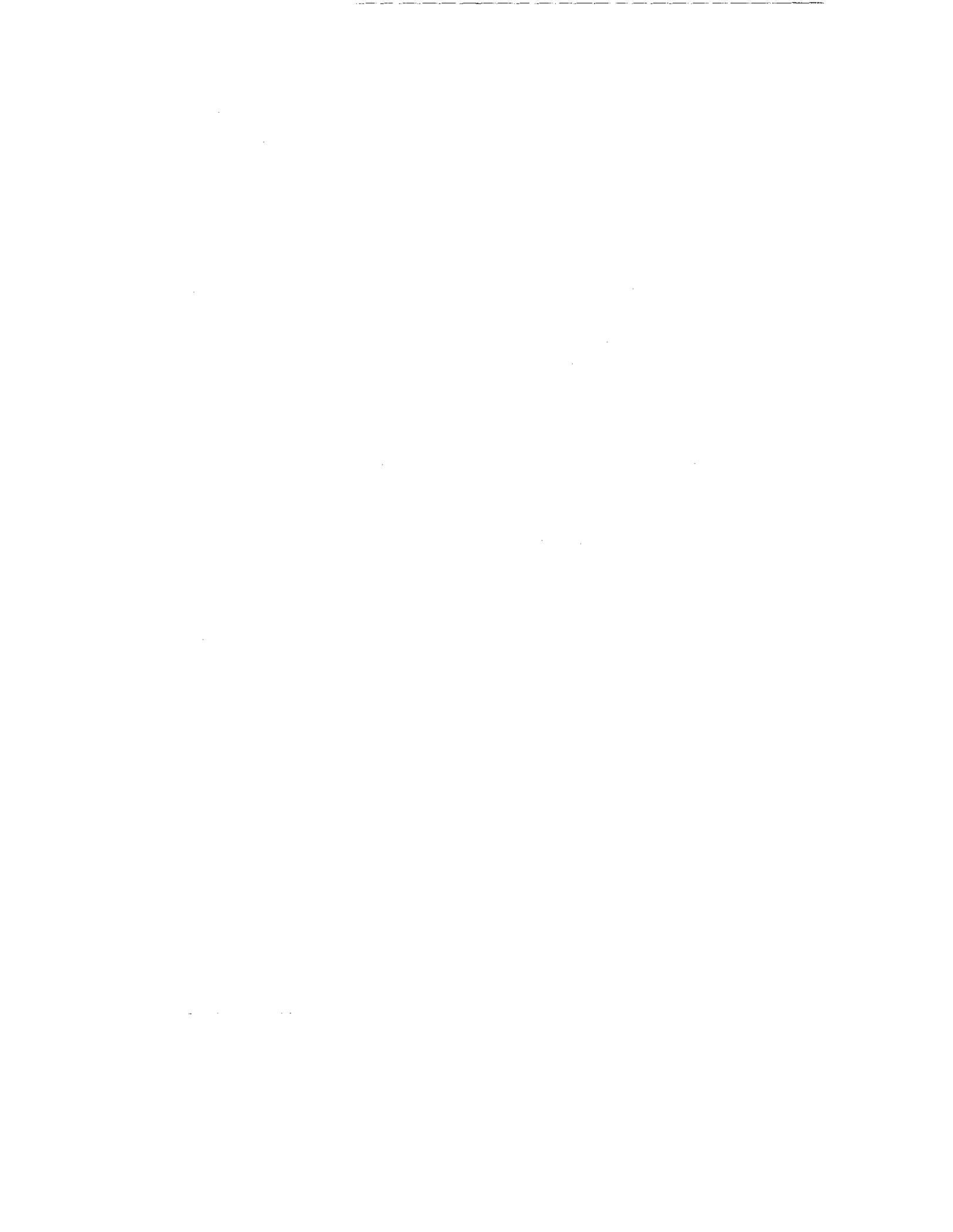
Hoy en día el sistema operativo para microcomputadoras que domina el mercado es Windows 95, y para los servidores del tipo HP9000 es el UNIX. Para los usuarios es muy importante que su aplicación corra sobre el sistema operativo Windows 95 ya que es el que tiene instalado en sus computadoras y es con el que están familiarizados. Con respecto al sistema operativo en el servidor, lo importante es que para los usuarios sea un punto transparente que no les afecte de ninguna manera. Estudiando la situación actual, y sabiendo que el sistema operativo de los servidores puede ser UNIX o Windows NT, es necesario que las

herramientas de desarrollo sean compatibles con este tipo de sistemas operativos.

Después de todo este análisis detallado de las herramientas y tomando en cuenta los puntos que se citaron en esta última parte podemos concluir que la solución óptima es la construcción de un sistema con arquitectura cliente/servidor utilizando en la parte del servidor el manejador de base de datos Sybase System 11 y en la parte del cliente la herramienta de desarrollo PowerBuilder. Debido a que es la única manera de que se pueda satisfacer plenamente las expectativas del cliente, además de que el sistema que se construirá no quedará en la obsolescencia en corto tiempo.

Los puntos en los que la tercera opción de solución sobrepasa en gran medida a las primeras dos opciones y que determinaron que se construyera un nuevo sistema con arquitectura cliente/servidor y utilizando las herramientas de desarrollo mencionadas en el párrafo anterior son:

- Funcionalidad del sistema
- Seguridad
- Desempeño
- Infraestructura de cómputo disponible. La institución cuenta con varios servidores HP9000 disponibles.
- Recursos humanos disponibles tanto en el mercado como en la institución con conocimientos de las herramientas de desarrollo.
- Capacidad de crecimiento.
- Disponibilidad del manejador de base de datos Sybase System 11 y la herramienta de desarrollo de clientes PowerBuilder 5.0 en la institución (Debido a una licitación que se dio anteriormente para la compra de herramientas de desarrollo de sistemas).



## 4.1 BACK END PARA CADA MÓDULO

### 4.1.1 Diagrama Entidad - Relación

Una de las tareas primordiales para el desarrollo de un sistema es el modelado del análisis, a fin de que se pueda describir en forma gráfica los requerimientos del cliente, se genere una base para la construcción de un sistema y se determinen los puntos importantes a validar una vez que el sistema se encuentre construido.

Primeramente hay que definir los objetos de datos, los cuales se muestran gráficamente en el diagrama entidad – relación.

En el modelado de un diagrama entidad relación se utilizan tres elementos que son los objetos de datos, los atributos de los objetos y las relaciones que es el elemento de conexión entre los objetos de datos.

Recordemos que la cardinalidad indica el número de instancias (una o muchas) de un objeto en relación a otro objeto y la modalidad nos indica si una relación es obligada o es opcional. Para el problema que nos ocupa se definieron varios objetos de datos y sus relaciones; en la figura 4.1.1.1 se muestran los símbolos que se utilizaron para representar la modalidad y la cardinalidad.

Símbolo	Modalidad	Cardinalidad	Descripción
	Obligada	Uno	Debe existir uno y solo uno
	Obligada	Muchos	Debe existir uno o muchos
	Opcional	Uno	Puede existir uno o muchos
	Opcional	Muchos	Puede existir uno o muchos o ninguno

Figura 4.1.1.1. Símbolos de representación de relaciones.

En el anexo A se muestra el diagrama entidad - relación para el sistema de contabilidad.

**Diccionario de datos**

Para la elaboración del diccionario de datos se utilizaron los siguientes tipos de datos:

- A(n) En donde A es una cadena alfanumérica y n el número de caracteres de esa cadena.
- VA(n) VA representa una cadena Alfanumérica variable y n el número máximo de caracteres de esa cadena.
- DT Tipo Fecha hora.
- N Tipo numérico.
- NEE,D Tipo numérico decimal en donde EE representa la longitud de la parte entera y D la longitud de la parte decimal.
- I Tipo entero.
- SI Tipo entero corto.

En las tablas el campo **I** determina si el campo es el identificador de la tabla, **M** si el campo debe de existir de forma obligada, y **DEP** si la relación es dependiente

**Entidad Aeropuertos**

**Descripción**

Contiene los aeropuertos que pertenecen a la red aeroportuaria.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Aeropuerto	asa_c_aeropuerto	A3	Si	Si
Región	asa_c_region	A3	No	Si
Estado	asa_c_estado	A10	No	Si
Nombre	Nombre_ae_str	VA60	No	No
Nombre	Nombre_ae_str	VA60	No	No
Alias	Aliaaerop ae str	VA40	No	No

**Campo de datos: asa\_c\_aeropuerto**

**Descripción**

Identificador del aeropuerto

**Campo de datos: asa\_c\_region**

**Descripción**

Región dentro de la cual está clasificado el aeropuerto.

**Campo de datos: asa\_c\_estado**

**Descripción**

Estado de la república en el cual se encuentra el aeropuerto.

**Campo de datos: nombre\_ae\_str**

**Descripción**

Nombre del aeropuerto.

**Campo de datos: aliasaerop\_ae\_str**

**Descripción**

Si el aeropuerto tiene algún alias, este se coloca en este campo, por ejemplo, casi todos los aeropuertos son nombrados dependiendo del lugar en donde se encuentran aunque sus nombres verdaderos son otros.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Catálogos Contables(ctb_c_cat_contable)	0,n	No	Ref_15226(ref_15226)

**Entidad Bitácora**

**Descripción**

Registro de la Bitácora de contabilizaciones.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Identificador contabilización	ctb_m_bitacora	NO	Si	Si
Fecha de contabilización	fec_contabiliza_bit_dt	DT	No	No
Pólizas correctas	pol_correctas_bit_i	I	No	No
Pólizas incorrectas	pol_incorrectas_bit_i	I	No	No
Total al debe	tot_debe_bit_i	N12,2	No	No
Total al haber	tot_haber_bit_i	N12,2	No	No
Error de contabilización	error_bid_str	VA255	No	No

**Campo de datos: IdContabilizacion**

**Descripción**

Identificador de la bitácora de contabilizaciones

**Campo de datos: FechaContabilizacion**

**Descripción**

Fecha en que se hace un proceso de contabilización

**Campo de datos: PolizasCorrectas**

**Descripción**

Número de pólizas que se contabilizaron de forma correcta

**Campo de datos: PolizasIncorrectas**

**Descripción**

Número de póliza que por algún motivo no se pudieron contabilizar

**Campo de datos: TotalDebe**

**Descripción**

Total al debe contabilizado

**Campo de datos: TotalHaber**

**Descripción**

Total al haber contabilizado

**Campo de datos: ErrorDeContabilización**

**Descripción**

Error detectado al momento de contabilizar

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Bitácora de pólizas (ctb_d_bit_poliza)	0,n	Si	Ref_17179(ref_17179)

**Entidad Bitácora de Movimientos**

**Descripción**

Bitácora de cada uno de los movimientos de las pólizas que se contabilizan

**Lista de atributos**

Nombre	Código	Tipo	I	M
Error de contabilización	error_bid_str	VA25 5	No	No

**Campo de datos: ErrorDeContabilización**

**Descripción**

Error detectado al momento de contabilizar

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Bitácora de pólizas (ctb_d_bit_poliza)	1,1	Si	Ref_17206(ref_17206)
Maestro de movimientos (ctb_m_movimiento)	1,1	Si	Ref_17235(ref_17235)
			5)

**Entidad Bitácora de pólizas**

**Descripción**

Bitácora de cada una de las pólizas que se contabilizan

**Lista de atributos**

Nombre	Código	Tipo	I	M
Error Póliza	error_bpo_str	VA255	No	No

**Campo de datos: ErrorDeContPoliza**

**Descripción**

Error de contabilización de la póliza

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Bitácora(ctb_m_bitacora)	1,1	Si	Ref_17179(ref_17179)
Maestro de polizas (ctb_m_poliza)	1,1	Si	Ref_17183(ref_17183)
Bitácora de Movimientos (ctb_d_bit_movtos)	0,n	Si	Ref_17206(ref_17206)

**Entidad Catálogo de áreas**

**Descripción**

Catálogo de áreas del organismo

**Lista de atributos**

Nombre	Código	Tipo	I	M
Area Organismo	asa_c_area	A10	Si	Si
Area Organismo	asa_c_area	A10	Si	Si
Descripción Area	descripcion_ar_str	VA60	No	No
Descripción Area	descripcion_ar_str	VA60	No	No

**Campo de datos: AreaOrganismo**

**Descripción**

Identificador del área

**Campo de datos: DescripcionArea**

**Descripción**

Descripción del área

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Centros de costo (asa_c_area_num)	0,n	No	Ref_26958(ref_26958)
Catálogos Contables (ctb_c_cat_contable)	1,n	No	Ref_4226(ref_4226)
Catálogo de cuentas contables (ctb_c_cuenta)	1,n	Si	rel_area_cat(rel_area_c at)

**Entidad Catálogo de cuentas contables**

**Descripción**

Entidad para almacenar los registros de las cuentas contables, aquí sólo se almacenan los datos generales de las cuentas, mas no sus saldos.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Identificador Cuenta	ctb_c_cuenta	I	Si	Si
Identificador Cuenta	ctb_c_cuenta	I	Si	Si
Nivel Cuenta	nivel_cat_si	SI	No	No
Nivel Cuenta	nivel_cat_si	SI	No	No

Nombre	Código	Tipo	I	M
Tipo1Cuenta	tipo1_cat_str	A1	No	No
Tipo2Cuenta	tipo2_cat_str	A1	No	No
Naturaleza cuenta	naturaleza_cat_str	A1	No	No
Cuenta contable	cuenta_cat_str	A15	No	No
Es detalle	es_det_cat_si	SI	No	No
Descripción Cuenta	descrip_cat_str	VA120	No	No
Cuenta bloqueada	bloquea_cat_str	A1	No	No
Propietario cuenta	prop_cat_str	A10	No	No
Permite sobregiro	sobregi_cat_str	A1	No	No
Nivel de sobregiro	niv_sob_cat_n	N15,2	No	No
Sobregiro total	sob_tot_cat_si	SI	No	No
Horario apertura	hr_aper_cat_dt	DT	No	No
Horario cierre	hr_cie_cat_dt	DT	No	No
Horario cierre	hr_cie_cat_dt	DT	No	No
Estado	estado_cat_str	A1	No	No
Estado	estado_cat_str	A1	No	No
AeropOrigen	asa_c_aerorig	A3	No	No
AeropOrigen	asa_c_aerorig	A3	No	No
AeropDestino	asa c aerodest	A3	No	No
AeropDestino	asa c aerodest	A3	No	No

**Campo de datos:** IDCuentaContable

**Descripción**

Identificador de la cuenta contable.

**Campo de datos: NivelCuenta**

**Descripción**

Nivel contable de la cuenta.

**Campo de datos: Tipo1Cuenta**

**Descripción**

En este campo se guardan los datos de la clasificación contable que tienen las cuentas Orden (O), Balance (B), Resultados (R).

**Campo de datos: Tipo2Cuenta**

**Descripción**

Se guarda la segunda identificación contable de una cuenta Activo (A), Pasivo (P), Capital (C) y Registro (R).

**Campo de datos: NaturalezaCuenta**

**Descripción**

Deudora (D), Acreedora(A)

**Campo de datos: CuentaContable**

**Descripción**

Se guarda la cuenta contable ejemplo: 123-002-00004

**Campo de datos: EsDetalle**

**Descripción**

Identifica si la cuenta es de detalle, si lo es contendrá un uno y si no contendrá un cero

**Campo de datos: DescripcionCuenta**

**Descripción**

Descripción contable de la cuenta

**Campo de datos: CuentaBloqueada**

**Descripción**

Campo para bloquear las cuentas, es decir que ya no se permita la aplicación contable sobre esa cuenta, si está bloqueada es uno y en caso contrario es cero.

**Campo de datos: PropietarioCuenta**

**Descripción**

Identificación de la persona propietaria de la cuenta, es decir quien sólo puede hacer aplicaciones contables sobre esa cuenta.

**Campo de datos: PermiteSobregiro**

**Descripción**

Determina si la cuenta se puede sobregirar o no, si tiene un uno, la cuenta se puede sobregirar y si tiene un cero la cuenta no se podrá sobregirar.

**Campo de datos: NiveldeSobregiro**

**Descripción**

En caso de que la cuenta tenga un nivel de sobregiro aquí se anota la cantidad máxima de sobregiro

**Campo de datos: SobregiroTotal**

**Descripción**

Si este campo contiene un uno la cuenta se puede sobregirar sin ningún nivel.

**Campo de datos: HorarioApertura**

**Descripción**

Las cuentas cuentan con horarios, este campo indica el horario inicial en el cual la cuenta se encuentra activa

**Campo de datos: HorarioCierre**

**Descripción**

Horario en el cual la cuenta deja de estar activa y no se pueden hacer más aplicaciones contables sobre ella.

**Campo de datos: Estado**

**Descripción**

Si la cuenta contiene la letra B, la cuenta está dada de baja pero tiene saldo

**Campo de datos: AeropOrigen**

**Descripción**

Campos para procesos de reaplicación.

**Campo de datos: AeropDestino**

**Descripción**

Campos para procesos de reaplicación.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Iconos	0,1	No	Ref_1613(ref_1613)
Catálogos Contables (ctb_c_cat_contable)	1,1	Si	Ref_2939(ref_2939)
Divisas (ctb_c_divisas)	0,1	No	Ref_5878(ref_5878)
Histórico de movimientos (ctb_h_movimiento)	1,n	No	ref_936(ref_936)
Maestro de cuentas contables (ctb_m_cuenta)	1,1	Si	Ref_cat_mtro_cta(ref_cat_mt ro_cta)
Catálogo de areas (asa_c_area)	1,1	Si	rel_area_cat(rel_area_cat)
Catálogo de cuentas contables (ctb_c cuenta)	0,1	No	rel_padre_hijo(rel_padre_hij o)
Catálogo de cuentas contables (ctb c cuenta)	0,1	No	rel_padre_hijo(rel_padre_hij o)

**Entidad Catálogos Contables**

**Descripción**

Control de los catálogos contables de cada uno de los aeropuertos que pertenecen a la red aeroportuaria.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Catálogo	ctb_c_cat_contable	I	Si	Si
Año	anio_cct_si	SI	No	Si
Descripción	descrip_cct_str	A80	No	No
Nivel	nivel_cct_si	SI	No	No
Catálogo padre	padre_cct_si	SI	No	No
Nivel concentración	niv_con_cct_si	SI	No	No
Normativo	norma_cct_si	SI	No	No
Concentrador	concen_cct_si	SI	No	No
Concentrador aeropuerto	concenae_cct_si	SI	No	No
Contabilidad ASA	contasa_cct_si	SI	No	No
AeropOrigen	asa_c_aerorig	A3	No	No
AeropDestino	asa_c_aerodest	A3	No	No
AeropDestino	asa_c_aerodest	A3	No	No

**Campo de datos: ctb\_c\_cat\_contable**

**Descripción**

Identificador del catálogo contable

**Campo de datos: año**

**Descripción**

Año contable del catálogo, hay que recordar que en la base de datos se deben de tener almacenados al menos los últimos cinco años de la contabilidad.

**Campo de datos: descripcion**

**Descripción**

Descripción del catálogo, la descripción debe de tener el nombre del aeropuerto al que pertenece.

**Campo de datos: nivel**

**Descripción**

Nivel del catálogo, los niveles son los siguientes:

- 1.-catálogo del aeropuerto
- 2.- catálogo concentrador de la unidad de neocios
- 3.- catálogo normativo de la red aeroportuaria

**Campo de datos: catalogoPadre**

**Descripción**

Catálogo en el cual se concentra el catálogo del registro.

**Campo de datos: NivelConcentracion**

**Descripción**

Nivel contable en el cual se encuentra la primera cuenta acumulativa.

**Campo de datos: normativo**

**Descripción**

Determina si el catálogo es normativo, es importante recalcar que sólo debe de haber un catálogo normativo-concentrador por cada base de datos.

**Campo de datos: concentrador**

**Campo de datos: ConcentradorAeropuerto**

**Descripción**

Determina si el catálogo concentra la contabilidad de algún aeropuerto.

**Campo de datos: contabilidadASA**

**Descripción**

Por los procesos de desincorporación, se debe de tener contemplado la posibilidad de manejar otro tipo de contabilidades, por lo que en este campo se verifica si es la contabilidad de las oficinas generales la que se está manejando o es otro tipo de contabilidad.

**Campo de datos: AeropOrigen**

**Descripción**

Campos para procesos de reaplicación.

**Campo de datos: AeropDestino**

**Descripción**

Campos para procesos de reaplicación.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Aeropuertos (asa_c_aeropuerto)	0,1	No	Ref_15226(ref_15226)
Maestro de movimientos (ctb_m_movimiento)	0,n	No	Ref_24818(ref_24818)
Temporal maestro de polizas (ctb_m_car_poliza)	0,n	Si	Ref_25410(ref_25410)
Temporal maestro de movimientos (ctb_m_car_movimiento)	0,n	No	Ref_25430(ref_25430)
Centros de costo (asa_c_area_num)	1,1	No	Ref_26963(ref_26963)
Centros de costo (asa_c_area_num)	1,1	No	Ref_26963(ref_26963)
Catálogo de cuentas contables (ctb_c_cuenta)	1,n	Si	Ref_2939(ref_2939)
Catálogo de cuentas contables (ctb_c_cuenta)	1,n	Si	Ref_2939(ref_2939)
Catálogo de areas	0,1	No	Ref_4226(ref_4226)

Entidad	Cardinalidad	Dep.	Relación
(asa_c_area) Maestro de polizas (ctb_m_poliza)	0,n	Si	Ref_8148(ref_8148)
Histórico de pólizas (ctb_h_poliza)	1,n	Si	Ref_8174(ref_8174)
Cierres contables (ctb_m_cierre)	1,n	Si	Ref_9411(ref_9411)
Tipos de catálogo (maestro) (ctb_c_tpcat)	1,1	No	Ref_tct_cct(ref_tct_cct)

**Entidad Centros de costo**

**Descripción**

Centro de costos que maneja el sistema

**Lista de atributos**

Nombre	Código	Tipo	I	M
Area Organismo Numérica	asa_c_area_num	SI	Si	Si

**Campo de datos: AreaOrganismoNumerica**

**Descripción**

Número que identifica el centro de costos

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Catálogo de areas (asa_c_area)	0,1	No	Ref_26958(ref_26958)
Catálogo de areas (asa_c_area)	0,1	No	Ref_26958(ref_26958)
Catálogos Contables (ctb_c_cat_contable)	1,n	No	Ref_26963(ref_26963)
Catálogos Contables (ctb_c_cat_contable)	1,n	No	Ref_26963(ref_26963)
Maestro de movimientos	0,n	No	Ref_27848(ref_27848)

Entidad	Cardinalidad	Dep.	Relación
(ctb_m_movimiento)			
Temporal maestro de movimientos (ctb_m_car_movimiento)	0,n	No	Ref_28468(ref_28468)

**Entidad Cierres contables**

**Descripción**

Controla los periodos contables para cada catálogo, dichos periodos son mensuales.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Año	anio_cie_si	SI	Si	Si
Mes	mes_cie_si	SI	Si	Si
Estado cierre	edo_cie_si	SI	No	No
Fecha de cierre	fecha_cie_dt	DT	No	No
Cerro normativo	cerro_nor_si	SI	No	No
Fecha cierre normativo	f_cie_nor_dt	DT	No	No
Usuario que cierra	sys_usuarios	A10	No	No

**Campo de datos: Año**

**Descripción**

Año contable.

**Campo de datos: Mes**

**Descripción**

Mes contable.

**Campo de datos: Estado cierre**

**Descripción**

Estado (cerrado/abierto) del periodo contable.

**Campo de datos: Fecha de cierre**

**Descripción**

Fecha en que se realizó el cierre contable.

**Campo de datos: Cerro normativo**

**Descripción**

En caso de que alguien de la contabilidad normativa cerrara el periodo contable, se marca el periodo cerrado por el normativo.

**Campo de datos: Fecha cierre normativo**

**Descripción**

En caso de que el normativo haya cerrado el periodo contable se almacena la fecha de cierre.

**Campo de datos: Usuario que cierra**

**Descripción**

Usuario que cierra el periodo contable.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Catálogos Contables (ctb_c_cat_contable)	1,1	Si	Ref_9411(ref_9411)

**Entidad Divisas**

**Descripción**

Almacena las divisas con las cuales puede trabajar el sistema

**Lista de atributos**

Nombre	Código	Tipo	I	M
Identificador divisa	ctb_c_divisas	SI	Si	Si
Descripción divisas	descrip_div_str	VA20	No	No
Descripción divisas	descrip_div_str	VA20	No	No

**Campo de datos: IdDivisas**

**Descripción**

Identificador de la divisa

**Campo de datos: DescripcionDivisas**

**Descripción**

Descripción de la divisa

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Tipos de Cambio (ctb_c_tipo_cambio)	1,n	Si	Ref_5032(ref_5032)
Catálogo de cuentas contables (ctb_c_cuenta)	1,n	No	Ref_5878(ref_5878)

**Entidad Iconos**

**Descripción**

Contiene los iconos contenidos en el menú.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Iconos del menú	sys_iconos	SI	Si	Si
Nombre del icono	nombre_del_icono	VA30	No	No

**Campo de datos: iconos del menú**

**Descripción**

Corresponde a los iconos contenidos en el menú.

**Campo de datos: nombre del icono**

**Descripción**

Nombre del icono del menú.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Catálogo de cuentas contables (ctb_c_cuenta)	1,n	No	Ref_1613(ref_1613)

**Entidad Maestro de cuentas contables**

**Descripción**

En esta entidad se guardan los saldos de las cuentas contables

**Lista de atributos**

Nombre	Código	Tipo	I	M
SaldoEneroApertura	ene_si_cta_n	N15,2	No	No
SaldoDebeEnero	ene_d_cta_n	N15,2	No	No
SaldoHaberEnero	ene_h_cta_n	N15,2	No	No
SaldoEneroCierre	ene_sf_cta_n	N15,2	No	No
SaldoFebreroApertura	feb_si_cta_n	N15,2	No	No
SaldoDebeFebrero	feb_d_cta_n	N15,2	No	No
SaldoHaberFebrero	feb_h_cta_n	N15,2	No	No
SaldoFebreroCierre	feb_sf_cta_n	N15,2	No	No
SaldoMarzoApertura	mar_si_cta_n	N15,2	No	No
SaldoDebeMarzo	mar_d_cta_n	N15,2	No	No
SaldoHaberMarzo	mar_h_cta_n	N15,2	No	No
SaldoMarzoCierre	mar_sf_cta_n	N15,2	No	No
SaldoAbrilApertura	abr_si_cta_n	N15,2	No	No
SaldoDebeAbril	abr_d_cta_n	N15,2	No	No
SaldoDebeAbril	abr_d_cta_n	N15,2	No	No
SaldoHaberAbril	abr_h_cta_n	N15,2	No	No
SaldoHaberAbril	abr_h_cta_n	N15,2	No	No
SaldoAbrilCierre	abr_sf_cta_n	N15,2	No	No
SaldoAbrilCierre	abr_sf_cta_n	N15,2	No	No
SaldoMayoApertura	may_si_cta_n	N15,2	No	No
SaldoMayoApertura	may_si_cta_n	N15,2	No	No
SaldoDebeMayo	may_d_cta_n	N15,2	No	No
SaldoDebeMayo	may_d_cta_n	N15,2	No	No
SaldoHaberMayo	may_h_cta_n	N15,2	No	No
SaldoHaberMayo	may_h_cta_n	N15,2	No	No
SaldoMayoCierre	may_sf_cta_n	N15,2	No	No
SaldoMayoCierre	may_sf_cta_n	N15,2	No	No
SaldoJunioApertura	jun_si_cta_n	N15,2	No	No

Nombre	Código	Tipo	I	M
SaldoDebeJunio	jun_d_cta_n	N15,2	No	No
SaldoHaberJunio	jun_h_cta_n	N15,2	No	No
SaldoJunioCierre	jun_sf_cta_n	N15,2	No	No
SaldoJulioApertura	jul_si_cta_n	N15,2	No	No
SaldoDebeJulio	jul_d_cta_n	N15,2	No	No
SaldoHaberJulio	jul_h_cta_n	N15,2	No	No
SaldoJulioCierre	jul_sf_cta_n	N15,2	No	No
SaldoAgostoApertura	ago_si_cta_n	N15,2	No	No
SaldoDebeAgosto	ago_d_cta_n	N15,2	No	No
SaldoHaberAgosto	ago_h_cta_n	N15,2	No	No
SaldoAgostoCierre	ago_sf_cta_n	N15,2	No	No
SaldoSeptiembreApertura	sep_si_cta_n	N15,2	No	No
SaldoDebeSeptiembre	sep_d_cta_n	N15,2	No	No
SaldoHaberSeptiembre	sep_h_cta_n	N15,2	No	No
SaldoSeptiembreCierre	sep_sf_cta_n	N15,2	No	No
SaldoOctubreApertura	oct_si_cta_n	N15,2	No	No
SaldoDebeOctubre	oct_d_cta_n	N15,2	No	No
SaldoHaberOctubre	oct_h_cta_n	N15,2	No	No
SaldoOctubreCierre	oct_sf_cta_n	N15,2	No	No
SaldoNoviembreApertura	nov_si_cta_n	N15,2	No	No
SaldoDebeNoviembre	nov_d_cta_n	N15,2	No	No
SaldoHaberNoviembre	nov_h_cta_n	N15,2	No	No
SaldoNoviembreCierre	nov_sf_cta_n	N15,2	No	No
SaldoDiciembreApertura	dic_si_cta_n	N15,2	No	No
SaldoDiciembreApertura	dic_si_cta_n	N15,2	No	No
SaldoDebeDiciembre	dic_d_cta_n	N15,2	No	No
SaldoDebeDiciembre	dic_d_cta_n	N15,2	No	No
SaldoHaberDiciembre	dic_h_cta_n	N15,2	No	No
SaldoHaberDiciembre	dic_h_cta_n	N15,2	No	No
SaldoDiciembreCierre	dic_sf_cta_n	N15,2	No	No
SaldoDiciembreCierre	dic_sf_cta_n	N15,2	No	No
SaldoDebeTotal	total_d_cta_n	N15,2	No	No
SaldoDebeTotal	total_d_cta_n	N15,2	No	No
SaldoHaberTotal	total_h_cta_n	N15,2	No	No
SaldoHaberTotal	total_h_cta_n	N15,2	No	No
HayMovimiento	movi_cta_n	SI	No	No
HayMovimiento	movi_cta_n	SI	No	No
AeropOrigen	asa_c_aerorig	A3	No	No
AeropOrigen	asa_c_aerorig	A3	No	No
AeropDestino	asa_c_aerodest	A3	No	No
AeropDestino	asa_c_aerodest	A3	No	No

Campo de datos: SaldoEneroApertura

Descripción

Saldo inicial del mes de enero.

**Campo de datos: SaldoDebeEnero**

**Descripción**

Saldo al debe del mes de enero

**Campo de datos: SaldoHaberEnero**

**Descripción**

Saldo al haber del mes de enero

**Campo de datos: SaldoEneroCierre**

**Descripción**

Saldo final del mes de enero

**Nota:** para los demás campos de saldos la descripción es semejante

**Campo de datos: SaldoDebeTotal**

**Descripción**

Saldo anual al debe de la cuenta

**Campo de datos: SaldoHaberTotal**

**Descripción**

Saldo anual al haber de la cuenta

**Campo de datos: HayMovimiento**

**Descripción**

Determina si la cuenta ya fue afectada en alguna ocasión

**Campo de datos: AeropOrigen**

**Descripción**

Campos para procesos de reaplicación.

**Campo de datos: AeropDestino**

**Descripción**

Campos para procesos de reaplicación.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Catálogo de cuentas contables (ctb_c_cuenta)	1,1	Si	Ref_cat_mtro_cta(ref_cat_mtro_cta)

**Entidad Maestro de movimientos**

**Descripción**

Identificador en donde se guardan los movimientos pertenecientes a una póliza.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Secuencial Por póliza	sec_mov_si	SI	Si	Si
Area Cuenta	area_cta_pol_str	A10	No	No
Identificador cuenta contable	ctb_c_cuenta	I	No	No
Referencia	refer_cta_str	VA15	No	No
Concepto movimiento	concepto_mov_str	VA120	No	No
Cheque referencia	chq_ref_mov_str	VA15	No	No
Importe al debe	debe_mov_n	N15,2	No	No
Importe al haber	haber_mov_n	N15,2	No	No
AeropOrigen	asa_c_aerorig	A3	No	No
AeropDestino	asa_c_aerodest	A3	No	No

**Campo de datos: SecuencialPorPoliza**

**Descripción**

Secuencia del movimiento

**Campo de datos: AreaCuenta**

**Descripción**

Área de la cuenta contable a la cual afecta el movimiento

**Campo de datos: IDCuentaContable**

**Descripción**

Identificador de la cuenta contable.

**Campo de datos: referencia**

**Descripción**

Nombre de la cuenta contable a la que afecta el movimiento.

**Campo de datos: ConceptoMovimiento**

**Descripción**

Concepto del movimiento contable

**Campo de datos: ChequeReferencia**

**Descripción**

Referencia por la cual se está haciendo el movimiento por lo regular es un número de cheque

**Campo de datos: ImporteAlDebe**

**Descripción**

Cantidad por la cual se afecta al debe a la cuenta

**Campo de datos: ImporteAlHaber**

**Descripción**

Cantidad por la cual se afecta al haber a la cuenta

**Campo de datos: AeropOrigen**

**Descripción**

Campos para procesos de reaplicación.

**Campo de datos: AeropDestino**

**Descripción**

Campos para procesos de reaplicación.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Bitácora de Movimientos (ctb_d_bit_movtos)	0,n	Si	Ref_17235(ref_17235)
Catálogos Contables (ctb_c_cat_contable)	0,1	No	Ref_24818(ref_24818)
Centros de costo (asa_c_area_num)	0,1	No	Ref_27848(ref_27848)
Maestro de polizas (ctb_m_poliza)	1,1	Si	rel_mpol_mmov(rel_mpol_mmov)

**Entidad Maestro de pólizas**

**Descripción**

Entidad para guardar los encabezados de las póliza contables

**Lista de atributos**

Nombre	Código	Tipo	I	M
Tipo póliza	Tipo_pol_str	A1	Si	Si
Mes póliza	Mes_pol_si	SI	Si	Si
Número poliza	Num_pol_i	I	Si	Si
Area generadora póliza	area_gen_pol_str	A10	No	Si
Fecha registro póliza	f_reg_pol_dt	DT	No	Si
Fecha aplicación póliza	f_apl_pol_dt	DT	No	No
Fecha Contabilización póliza	f_cont_pol_dt	DT	No	No
UsrPropPoliza	sys_usuarios	VA10	No	No
UsrPropPoliza	sys_usuarios	VA10	No	No
UsrRevPoliza	u_rev_pol_str	VA10	No	No
UsrRevPoliza	u_rev_pol_str	VA10	No	No
UsrAutPoliza	u_aut_pol_str	VA10	No	No
UsrAutPoliza	u_aut_pol_str	VA10	No	No
ConceptoPoliza	concepto_pol_str	VA120	No	No
ConceptoPoliza	concepto_pol_str	VA120	No	No
Cifras Control Poliza	cif_con_pol_str	VA25	No	No
Cifras Control Poliza	cif_con_pol_str	VA25	No	No
Numero Movimientos Poliza	num_mov_pol_si	SI	No	No
Numero Movimientos Poliza	num_mov_pol_si	SI	No	No
Total debe póliza	tot debe pol n	N15,2	No	No
Total debe póliza	tot debe pol n	N15,2	No	No
Total haber póliza	tot haber pol n	N15,2	No	No
Total haber póliza	tot haber pol n	N15,2	No	No
Revisada	rev_pol_si	SI	No	No

Nombre	Código	Tipo	I	M
Autorizada	aut_pol_si	SI	No	No
Contabilizada	conta_pol_si	SI	No	No
AeropOrigen	asa_c_aeroorig	A3	No	No
AeropDestino	asa_c_aerodest	A3	No	No

**Campo de datos: TipoPoliza**

**Descripción**

Tipo de póliza Diario (D), Ingresos (I), Egresos (E), Única (U)

**Campo de datos: MesPoliza**

**Descripción**

Mes en el cual afecta contablemente la póliza

**Campo de datos: NumeroPoliza**

**Descripción**

Número de la póliza

**Campo de datos: AreaGeneradoraPoliza**

**Descripción**

Área que genera la póliza

**Campo de datos: FechaRegistroPoliza**

**Descripción**

Fecha en la cual se captura la póliza

**Campo de datos: FechaAplicacionPoliza**

**Descripción**

Fecha en que la póliza afecta los libros contables

**Campo de datos: FechaContabPoliza**

**Descripción**

Fecha valor de la operación contable

**Campo de datos: UsrPropPoliza**

**Descripción**

Usuario que capturó la póliza

**Campo de datos:** UsrRevPoliza

**Descripción**

Usuario que revisó la póliza

**Campo de datos:** UsrAutPoliza

**Descripción**

Usuario que autorizó la póliza para que se contabilice

**Campo de datos:** ConceptoPoliza

**Descripción**

Concepto de la póliza

**Campo de datos:** CifrasControlPoliza

**Descripción**

Cifras de control de la póliza, estas se forman por la sumatoria de todas las cuentas contables a las cuales afecta la póliza

**Campo de datos:** NumeroMovimientosPoliza

**Descripción**

Número de movimiento que tiene la póliza

**Campo de datos:** totaldebepoliza

**Descripción**

Total al debe de la póliza, para poder ser aplicada la póliza esta cifra debe de ser igual al total del haber de la póliza

**Campo de datos:** totalhaberpoliza

**Descripción**

Total al haber de la póliza, para poder ser aplicada la póliza esta cifra debe de ser igual al total del debe de la póliza

**Campo de datos: revisada**

**Descripción**

1 = póliza revisada, 0 = póliza sin revisar

**Campo de datos: Autorizada**

**Descripción**

1 = póliza autorizada, 0 = póliza sin ser autorizada

**Campo de datos: contabilizada**

**Descripción**

1 = Póliza contabilizada, esta póliza no debe de ser borrada, 0 = Póliza sin contabilizar

**Campo de datos: AeropOrigen**

**Descripción**

Campos para procesos de reaplicación.

**Campo de datos: AeropDestino**

**Descripción**

Campos para procesos de reaplicación.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Bitácora de pólizas (ctb_d_bit_poliza)	0,n	Si	Ref_17183(ref_17183)
Catálogos Contables (ctb_c_cat_contable)	1,1	Si	Ref_8148(ref_8148)
Maestro de movimientos (ctb_m_movimiento)	0,n	Si	rel_mpol_mmov(rel_mpol_mmov)

**Entidad Tipos de Cambio**

**Descripción**

Entidad para almacenar los tipos de cambio con los cuales puede trabajar el sistema.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Identificador del tipo de cambio	ctb_c_tipo_cambio	SI	Si	Si
Fecha del tipo de cambio	fec_reg_tca_dt	DT	No	Si
Descripción	desc_tca_str	VA20	No	No
Equivalencia	equiv_tca_n	N9,4	No	No

**Campo de datos: ctb\_c\_tipo\_cambio**

**Descripción**

Identificador del tipo de cambio

**Campo de datos: FechaRegistro**

**Descripción**

Fecha en que se dio de alta el tipo de cambio

**Campo de datos: descripcion**

**Descripción**

Descripción del tipo de cambio

**Campo de datos: equivalencia**

**Descripción**

Equivalencia numérica con pesos mexicanos

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Divisas(ctb_c_divisas)	1,1	Si	Ref_5032(ref_5032)

**Entidad Tipos de catálogo (maestro)**

**Descripción**

En esta entidad se almacenan los tipos de catálogos contables que el sistema puede manejar

**Lista de atributos**

Nombre	Código	Tipo	I	M
Tipo de catálogo	ctb_c_tpocat	SI	Si	Si
Número de niveles	numniv_tct_si	SI	No	No
Descripción tipo catálogo	descrip_tcp_str	VA120	No	No

**Campo de datos: Tipo de catálogo****Descripción**

A una estructura de una cuenta contable se le asigna un número para diferenciarla de otras.

**Campo de datos: Número de niveles****Descripción**

Número de Niveles de los que consta el catálogo.

**Campo de datos: Descripción tipo catálogo****Descripción**

Breve descripción del tipo del catálogo.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Catálogos Contables (ctb_c_cat_contable)	1,n	No	Ref_tct_cct(ref_tct_cct)
Tipos de Catálogos (detalle) (ctb_d_tpocat)	1,n	Si	Ref_tct_dtc(ref_tct_dtc)

**Entidad Tipos de Catálogos (detalle)****Descripción**

Se almacena el detalle de la estructura de la cuentas de los catálogos contables que el sistema puede manejar.

**Lista de atributos**

Nombre	Código	Tipo	I	M
Nivel	nivel_dtc_si	SI	Si	Si
Número de caracteres	num_carac_dtc_si	SI	No	No
Nombre del nivel	nombre_dtc_str	VA80	No	No

**Campo de datos: Nivel**

**Descripción**

Una cuenta contable puede estar compuesta por diferentes niveles contables, para cada nivel se le asigna un número. Ejemplo la cuenta 100-003-00005 tiene tres niveles y el primer nivel es el 100, el segundo es el 003 y el 00005 es el tercer nivel.

**Campo de datos: Número de caracteres**

**Descripción**

Almacena el número de caracteres que conforman un nivel contable.

**Campo de datos: Nombre del nivel**

**Descripción**

Se almacena el nombre del nivel contable, los nombres son cuenta, subcuenta, subsubcuenta, subsubcuenta, etc.

**Lista de referencias**

Entidad	Cardinalidad	Dep.	Relación
Tipos de catálogo (maestro) (ctb_c_tpocat)	1,1	Si	Ref_tct_dtc(ref_tct_dtc)

Para las entidades Histórico de póliza e histórico de movimientos, la explicación es similar que para las entidades maestro de póliza y maestro de movimientos, al igual que las tablas para un almacenamiento temporal de pólizas y movimientos.

#### 4.1.2 Diagrama contextual del sistema

Un diagrama contextual representa con una sola burbuja el sistema, y nos muestra la relación que tiene este con entidades externas. En la figura 4.1.2.1 se muestra el diagrama contextual del sistema de contabilidad.

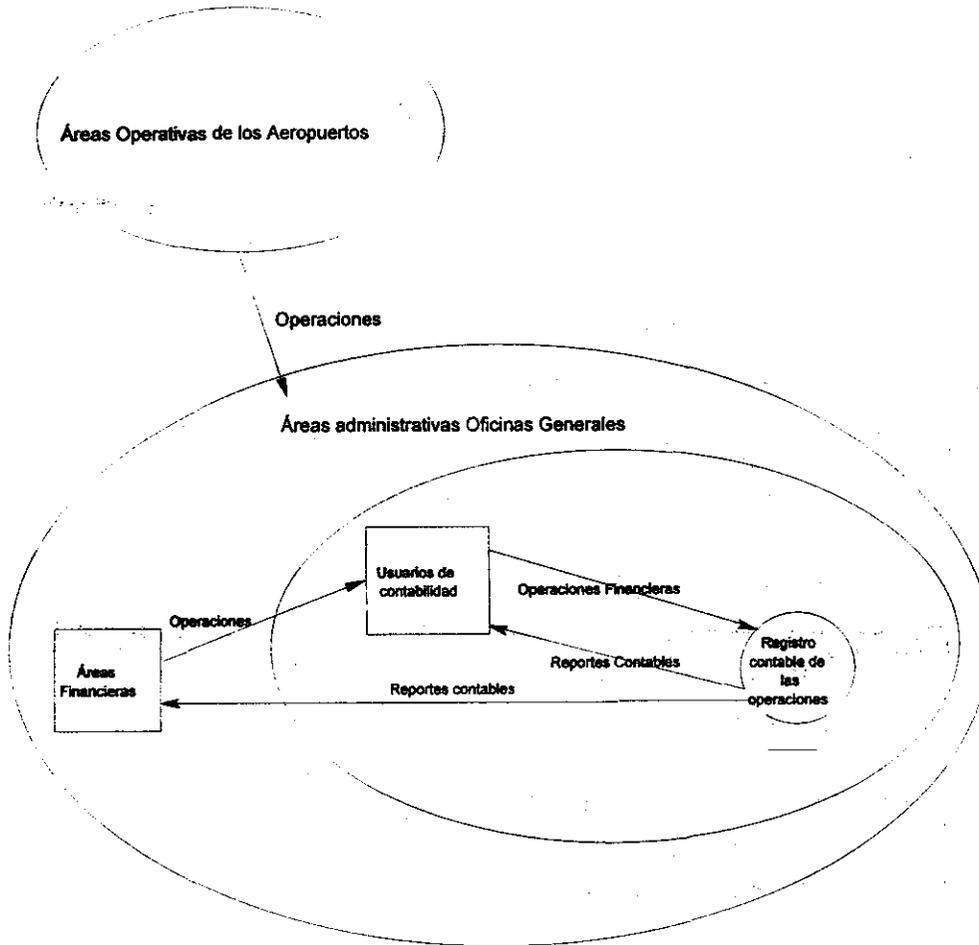


Figura 4.1.2.1. Diagrama contextual del sistema de contabilidad.

Notificar de la apertura o cierre [5.6]

Contabilización de las operaciones [2]

Autorización y contabilización de la póliza [2.3]

Abrir periodo contable [2.3.2]

### **4.1.3 Diagrama de procesos .**

Una vez que se tiene el diagrama de contexto del sistema, se empieza a descomponer en niveles mas bajos a fin de obtener los diagramas de procesos que nos identifican los procesos más importantes que se determinaron en el análisis el árbol de procesos es una representación sencilla que nos auxilia en el análisis de las gráficas.

#### **Árbol de Procesos**

Registro contable de las operaciones [0]

Alta de cuentas en el catálogo contable [4]

    Análisis de la cuenta a dar de alta [4.1]

    Captura de la cuenta a dar de alta [4.2]

    Incorporación al catálogo contable correspondiente [4.3]

Análisis de las operaciones a contabilizar [1]

    Complemento de los datos de las operaciones [1.6]

    Envío a la oficina contable correspondiente [1.4]

    Identificación de aeropuerto de origen de las operaciones [1.3]

    Identificación del tipo de operación [1.2]

    Verificación de los datos de las operaciones [1.5]

    Recepción o devolución de operaciones [1.1]

Cierre o apertura del ejercicio contable [5]

    Abrir el periodo contable [5.3]

    Abrir periodos en los catálogos padres [5.4]

    Cerrar el periodo contable [5.5]

    Determinar la acción a realizar [5.1]

    Determinar si los periodos contables en los catálogos padres están abiertos [5.2]

Generación de reportes contables [3]

    Determinación del periodo que abarca el reporte [3.2]

    Determinación del reporte a realizar [3.1]

    Generación del reporte [3.4]

    Procesamiento de la información requerida [3.3]

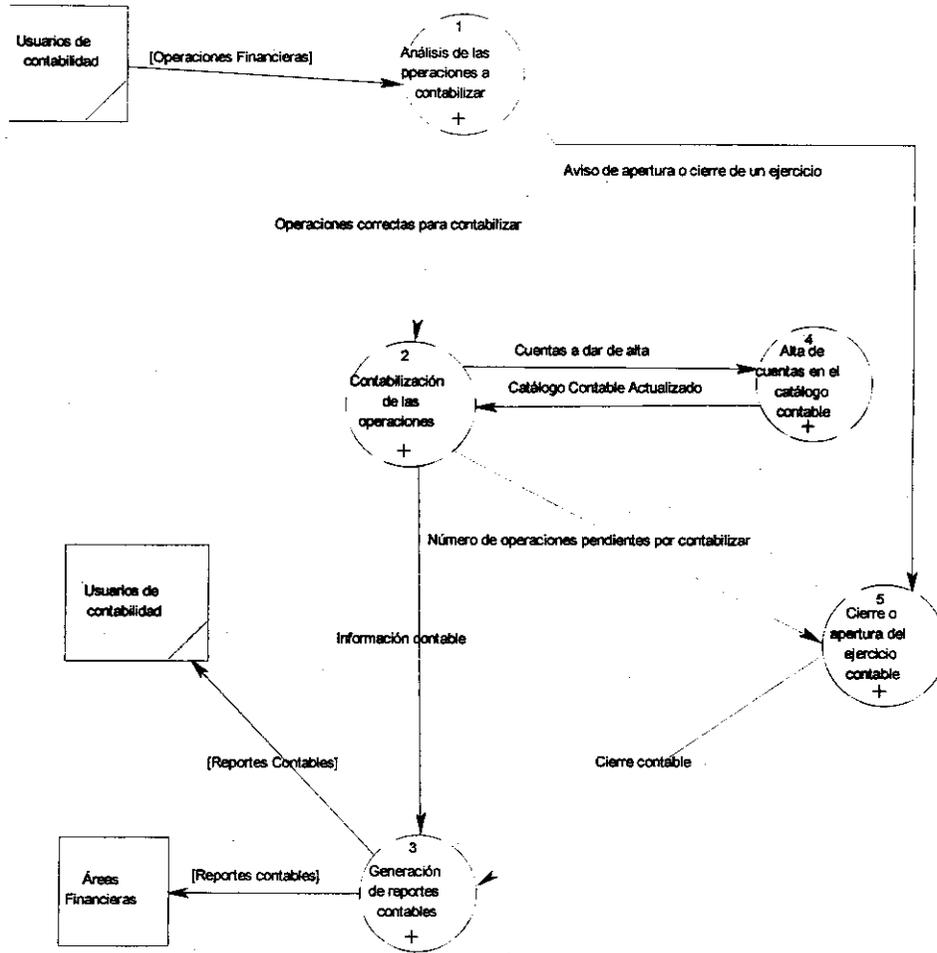


Figura 4.1.3.1. Subproceso registro contable de las operaciones

## Explicación de subprocesos

### *Subproceso Alta de cuentas en el catálogo contable*

#### Descripción del proceso

Dependiendo de las operaciones, es necesario dar de alta nuevas cuentas, por ejemplo una operación con un nuevo cliente, nos lleva a dar de alta la cuenta de ese cliente en la cuenta de mayor de clientes.

### *Subproceso Análisis de las operaciones a contabilizar*

#### Descripción del proceso

Una vez que las operaciones llegan al área contable, se procede a hacer un análisis de cada una de ellas para determinar la forma en que afectarán a la contabilidad

### *Subproceso Cierre o apertura del ejercicio contable*

#### Descripción del proceso

La fecha de contabilización de las operaciones determina si un periodo contable se debe de abrir, y una vez que todas las operaciones de un determinado mes o periodo contable se han introducido al sistema, se puede proceder a cerrar ese periodo.

### *Subproceso Contabilización de las operaciones*

#### Descripción del proceso

Una vez que las operaciones han sido analizadas y clasificadas, se proceden a hacer los asientos contables correspondientes

### *Subproceso Generación de reportes contables*

#### Descripción del proceso

Una vez que en el sistema de contabilidad se ha hecho el asiento correspondiente se procede a emitir los reportes financieros correspondientes

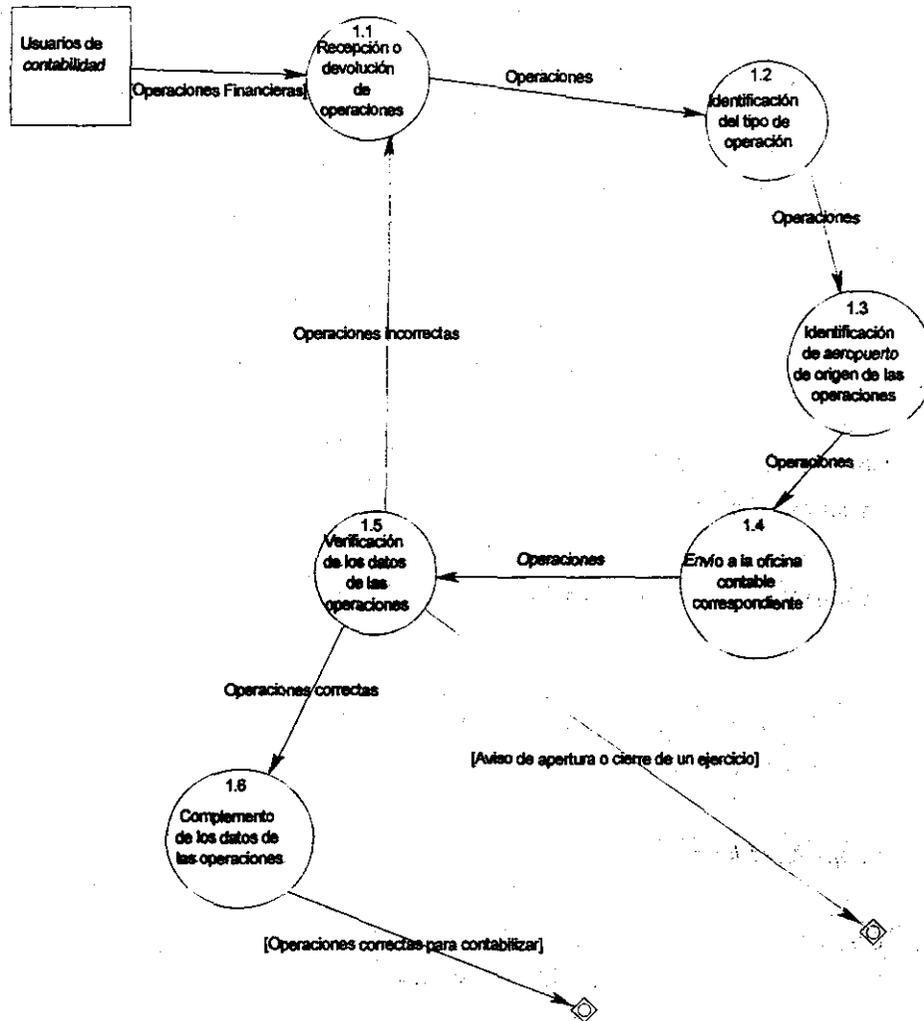


Figura 4.1.3.2. Subproceso Análisis de las operaciones a contabilizar

## **Explicación de subprocesos**

### *Subproceso Complemento de los datos de las operaciones*

#### **Descripción del proceso**

Una operación es susceptible de necesitar complementar los datos que trae, como por ejemplo en algunas operaciones se debe de anotar los precios vigentes de los combustibles que se venden o las tasas de interés que se manejan en un determinado banco, etc.

### *Subproceso Envío a la oficina contable correspondiente*

#### **Descripción del proceso**

Una vez que se ha determinado el aeropuerto y el tipo de operación se procede a enviar esta operación a la oficina contable correspondiente

### *Subproceso Identificación de aeropuerto de origen de las operaciones*

#### **Descripción del proceso**

Cada operación debe de ser identificada tanto por su tipo como por el aeropuerto en donde se origina

### *Subproceso Identificación del tipo de operación*

#### **Descripción del proceso**

El área contable debe de identificar el tipo de operación que llega para ser contabilizada

### *Subproceso Recepción o devolución de operaciones*

#### **Descripción del proceso**

Antes de proceder a elaborar una póliza el área contable debe de analizar y verificar los datos que se encuentren asentados en las operaciones.

*Subproceso Verificación de los datos de las operaciones*

Descripción del proceso

Cada operación debe de contener un número específico de datos y estos deben de ser correctos

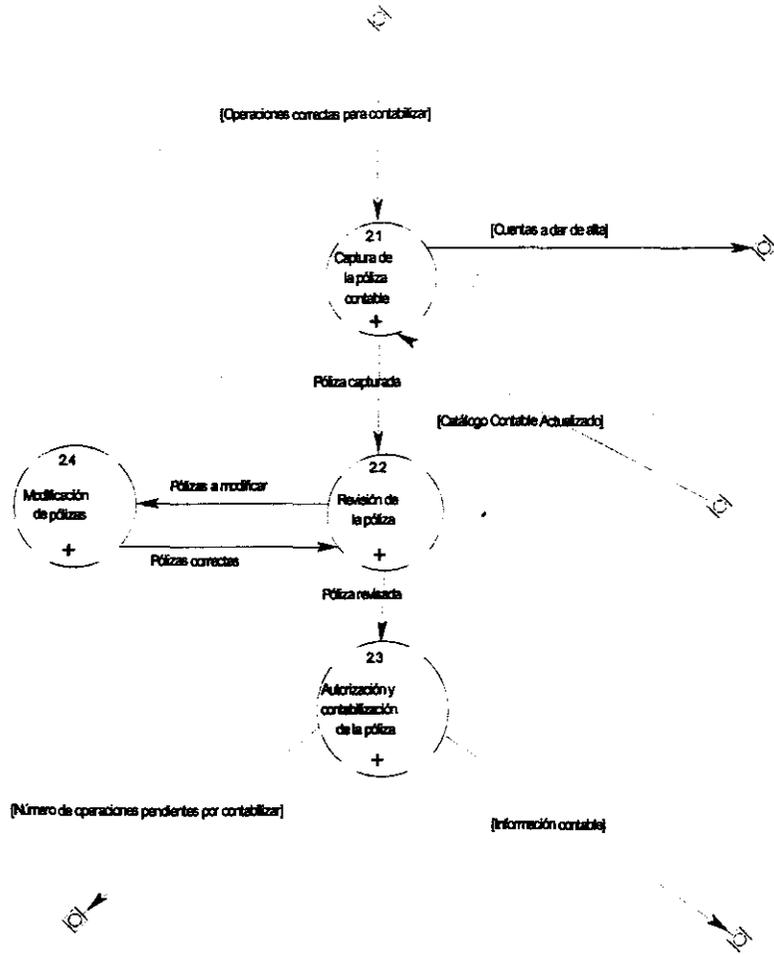


Figura 4.1.3.3. Subproceso Contabilización de las operaciones

## Explicación de subprocesos

### *Subproceso Autorización y contabilización de la póliza*

#### Descripción del proceso

Una vez que la póliza se ha revisado se procede a contabilizarla y marcarla como autorizada y contabilizada.

### *Subproceso Captura de la póliza contable*

#### Descripción del proceso

La unidad contable fundamental de una contabilidad es la póliza contable, la captura de estas es el inicio para la generación de los reportes contables.

### *Subproceso Modificación de pólizas*

#### Descripción del proceso

En este proceso se hacen las correcciones pertinentes a la póliza a fin de que los asientos contables que la conforman sean los correctos.

### *Subproceso Revisión de la póliza*

#### Descripción del proceso

El proceso de revisión es el que determina si los datos que han sido asentados en la póliza son correctos e incluso si la póliza en general debe de ser capturada, puede darse el caso de que alguna operación no genere contabilización y se quiera capturar por error una póliza.

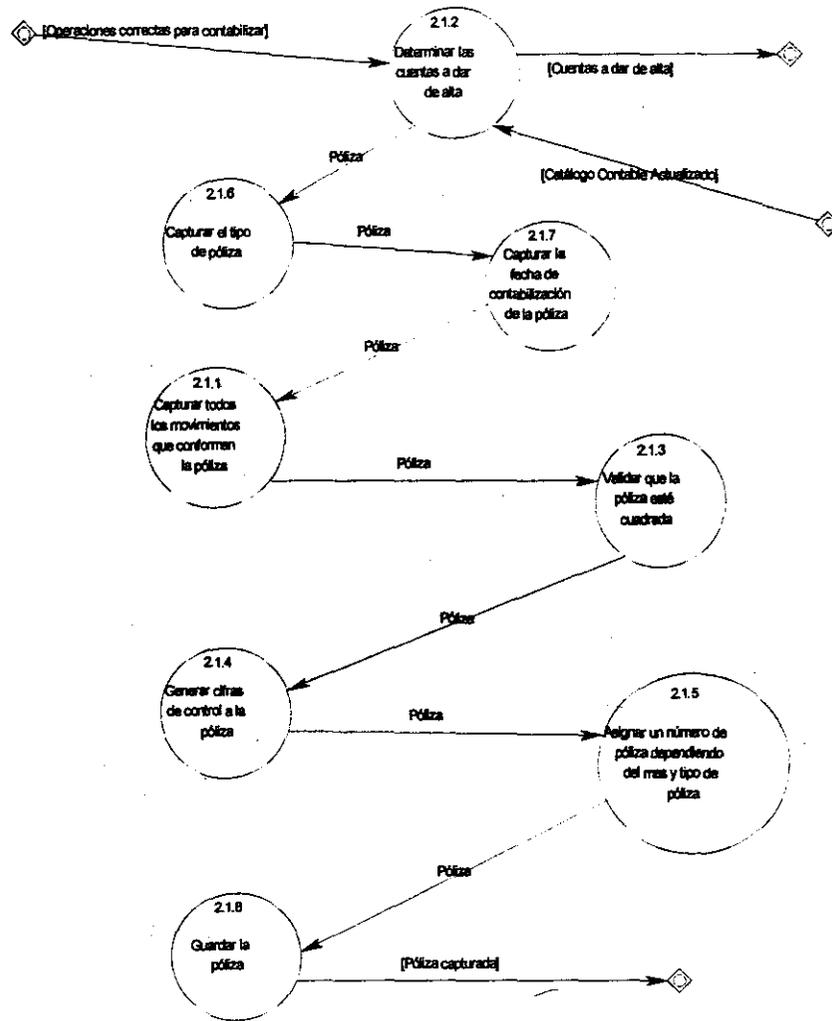


Figura 4.1.3.4. Subproceso captura de la póliza contable

## Explicación de subprocesos

### *Subproceso Capturar el tipo de póliza*

#### Descripción del proceso

Existen tres tipos de póliza, Diario, Ingresos y Egresos.

### *Subproceso Capturar la fecha de contabilización de la póliza*

#### Descripción del proceso

La fecha de contabilización es la que determina a que periodo contable afectará la póliza.

### *Subproceso Capturar todos los movimientos que conforman la póliza*

#### Descripción del proceso

Los movimientos contables que conforman la póliza son los datos que alimentarán a los reportes.

### *Subproceso Generar cifras de control a la póliza*

#### Descripción del proceso

Las cifras de control se formarán por la sumatoria de las cuentas a las que afectará la póliza.

### *Subproceso Validar que la póliza esté cuadrada*

#### Descripción del proceso

Para poder respetar la ley de la partida doble es necesario que la póliza contable se encuentre totalmente cuadrada, es decir que la suma de cargos sea igual a la suma de abonos.

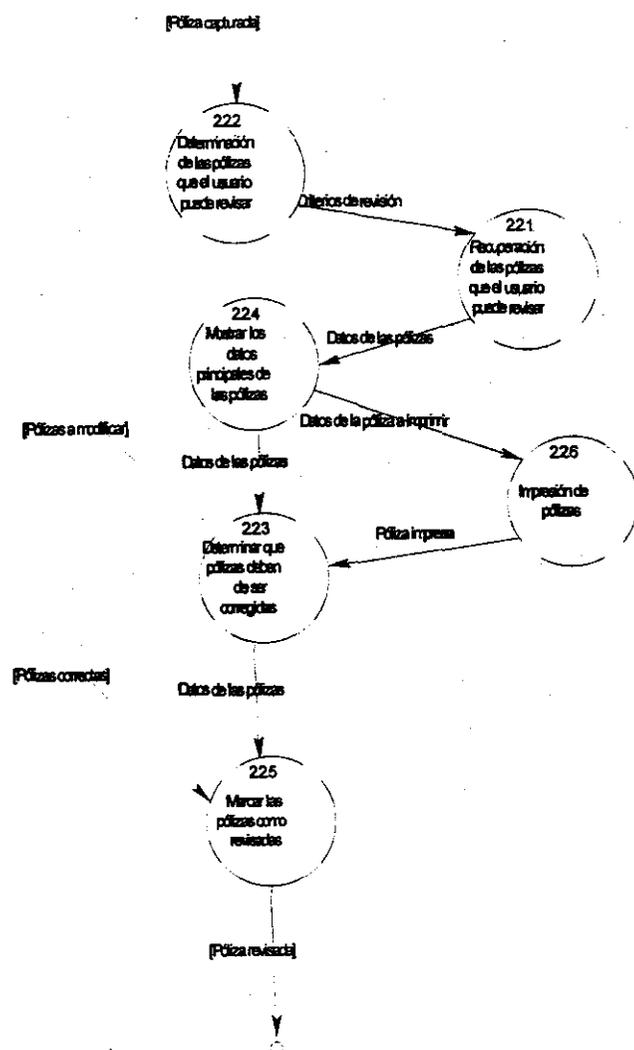


Figura 4.1.3.5. Subproceso revisión de la póliza.

**Explicación de subprocesos**

*Subproceso Determinación de las pólizas que el usuario puede revisar*

**Descripción del proceso**

Un usuario sólo podrá revisar pólizas que el no haya capturado

*Subproceso Mostrar los datos principales de las pólizas*

**Descripción del proceso**

Los datos principales de cada póliza se mostrarán en pantalla pero son necesarios otros datos para pasar al proceso de impresión de todos los datos de la póliza.

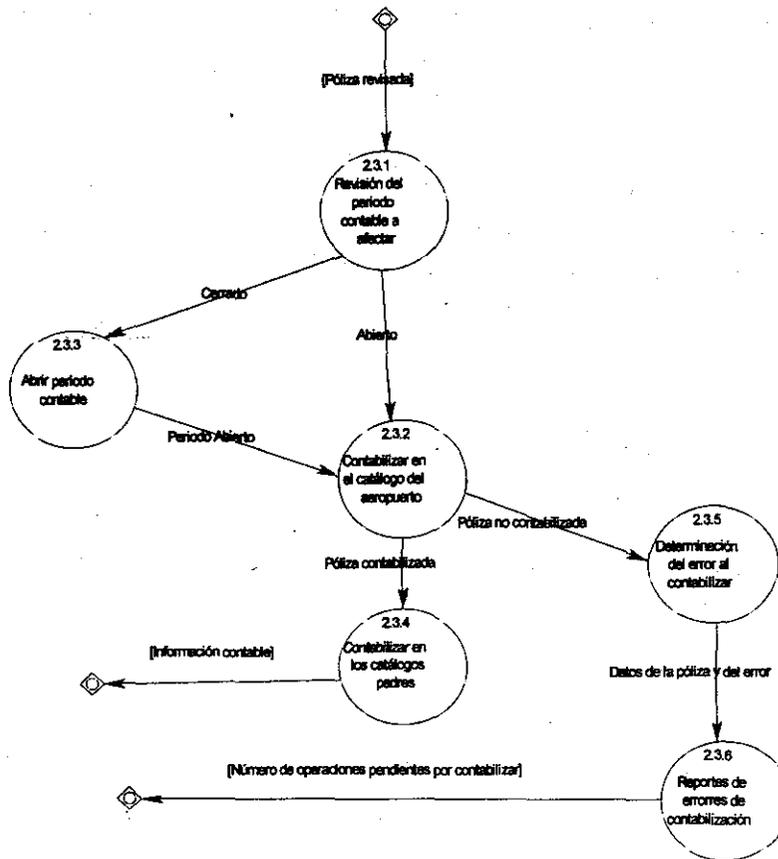


Figura 4.1.3.6. Subproceso autorización y contabilización de la póliza

## **Explicación de subprocesos**

### *Subproceso Contabilizar en los catálogos padres*

#### Descripción del proceso

Los catálogos padres de un catálogo de un aeropuerto son los catálogos concentradores de las regionales y el catálogo normativo de toda la red aeroportuaria.

### *Subproceso Determinación del error al contabilizar*

#### Descripción del proceso

Puede darse el caso de que una póliza no se contabilice, por ejemplo cuando esa póliza sobregire una cuenta o si el periodo contable al que afecta se encuentra cerrado.

### *Subproceso Revisión del periodo contable a afectar*

#### Descripción del proceso

Se tiene que verificar el periodo contable al que se quiere afectar, ya que una vez que un periodo se ha cerrado no se permite que se modifiquen los saldos por medio de la aplicación de una póliza.



Figura 4.1.3.6. Subproceso modificación de pólizas

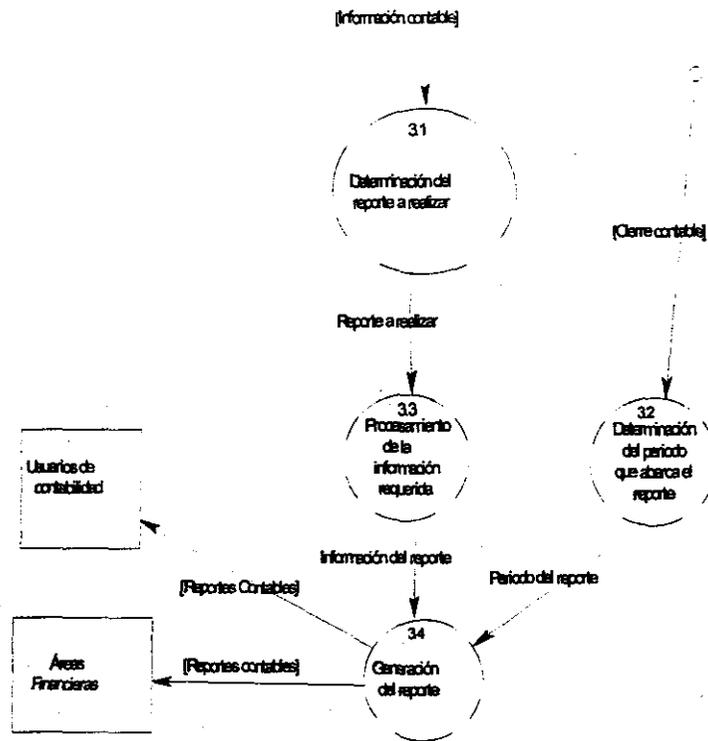


Figura 4.1.3.7. Subproceso Generación de reportes contables

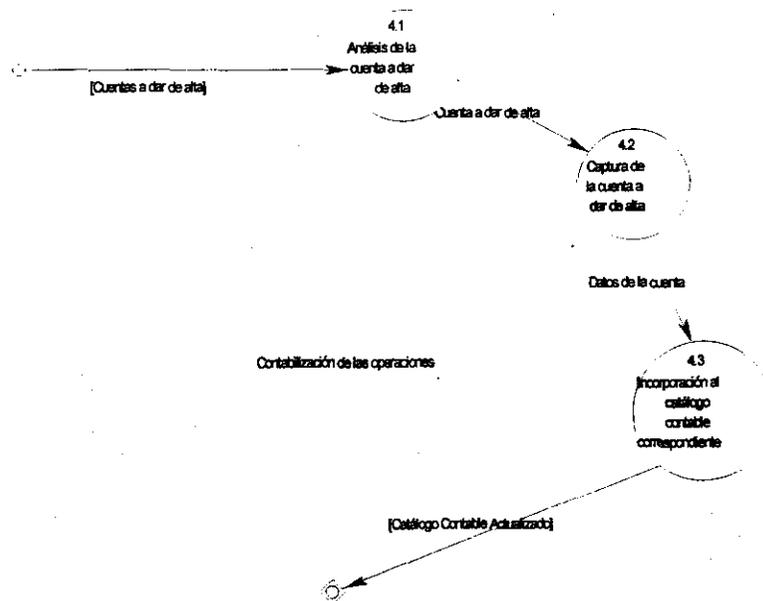


Figura 4.1.3.8. Subproceso alta de cuentas en el catálogo contable

### Explicación de subprocesos

#### *Subproceso Análisis de la cuenta a dar de alta*

##### Descripción del proceso

La cuenta a dar de alta en el catálogo es analizada para determinar su estructura contable.

#### *Subproceso Captura de la cuenta a dar de alta*

##### Descripción del proceso

Una vez que se ha determinado la estructura de la cuenta contable se procede a capturar sus datos.

Subproceso Incorporación al catálogo contable correspondiente

Descripción del proceso

Una vez que la cuenta es capturada, el sistema la incorpora al catálogo contable correspondiente.

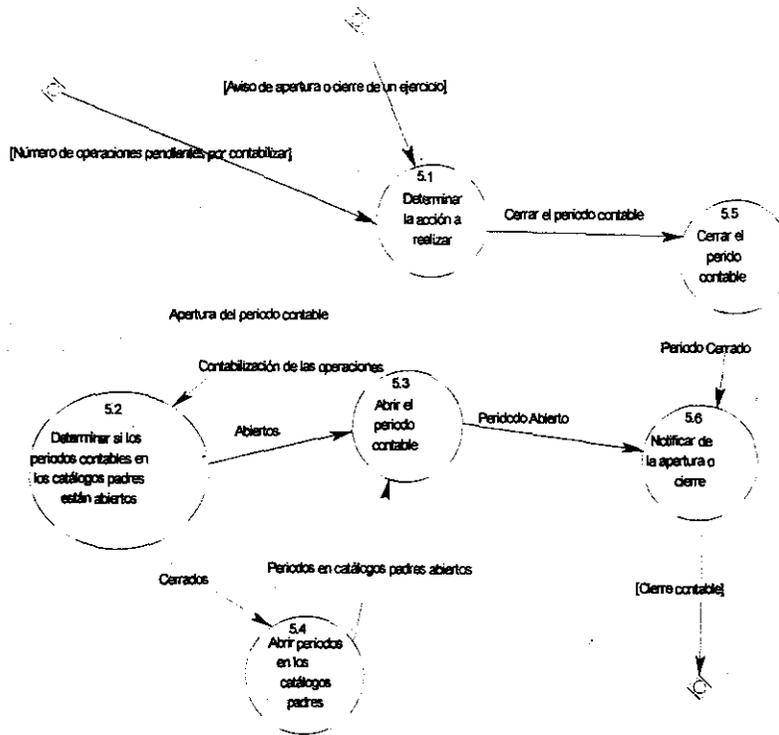


Figura 4.1.3.8. Subproceso cierre o apertura del ejercicio contable

4.1.4 Diagramas de flujo de datos.

Basándonos en los diagramas de procesos, se construyeron los diagramas de flujo de datos agregándoles las entidades en las cuales se hacía un almacenamiento o modificación de datos durante los procesos, sólo se presentan los diagramas en los cuales se hace referencia a las entidades del modelo entidad-relación.

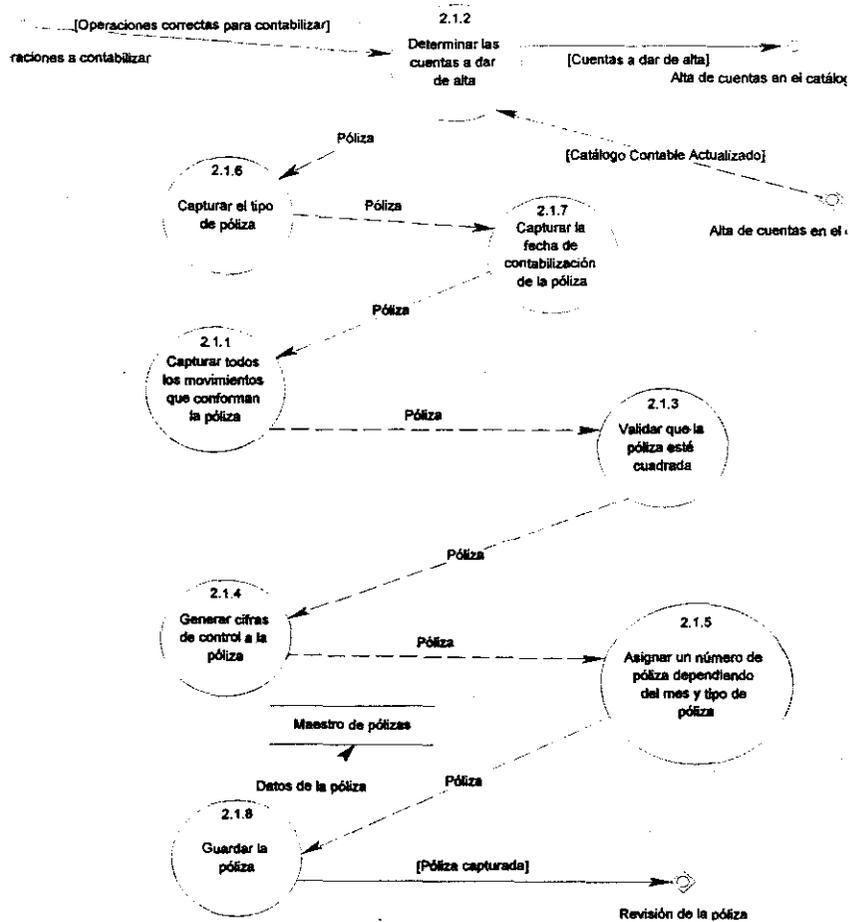


Figura 4.1.4.1. Subproceso Captura de la póliza contable.

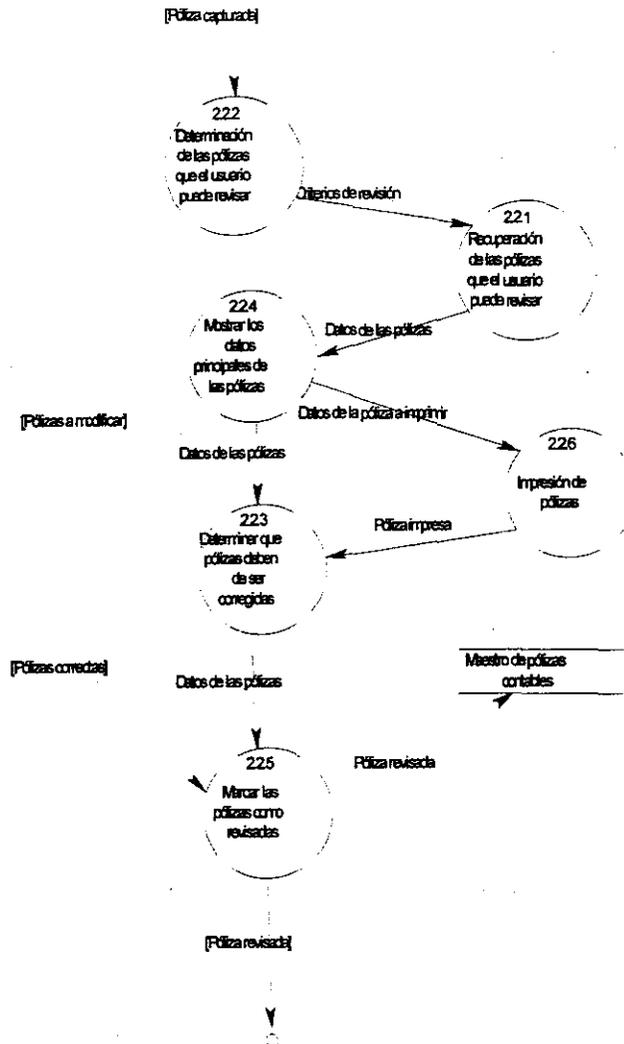


Figura 4.1.4.2. Subproceso Revisión de la póliza

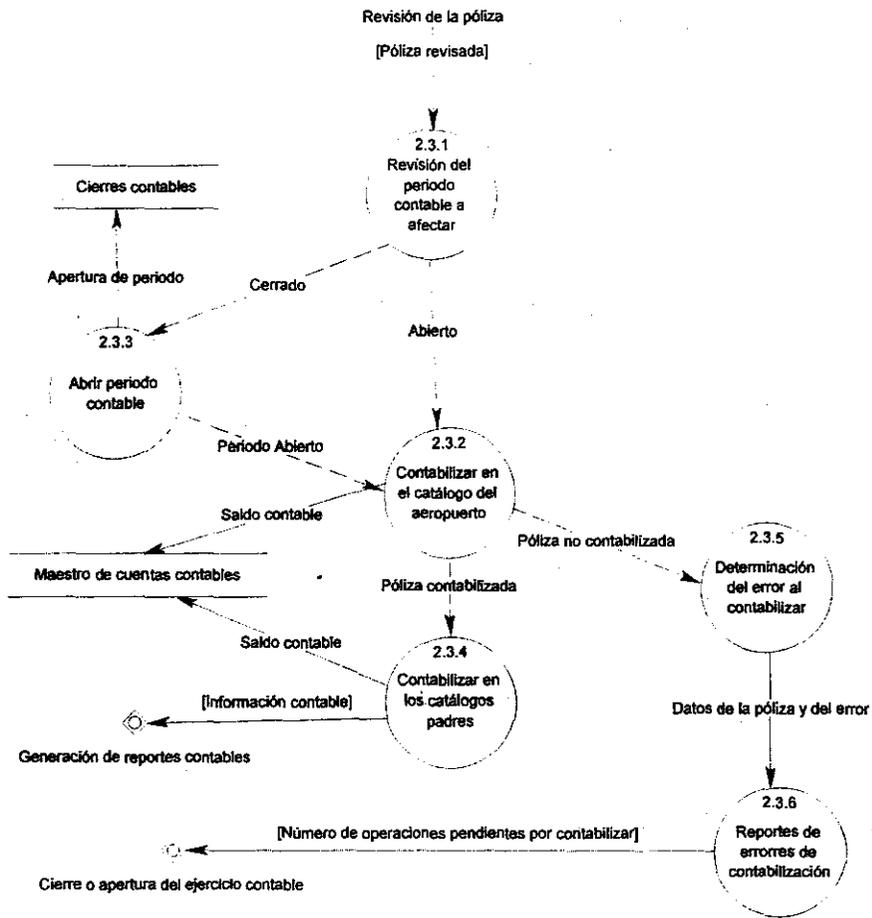


Figura 4.1.4.3. Subproceso autorización y contabilización de la póliza

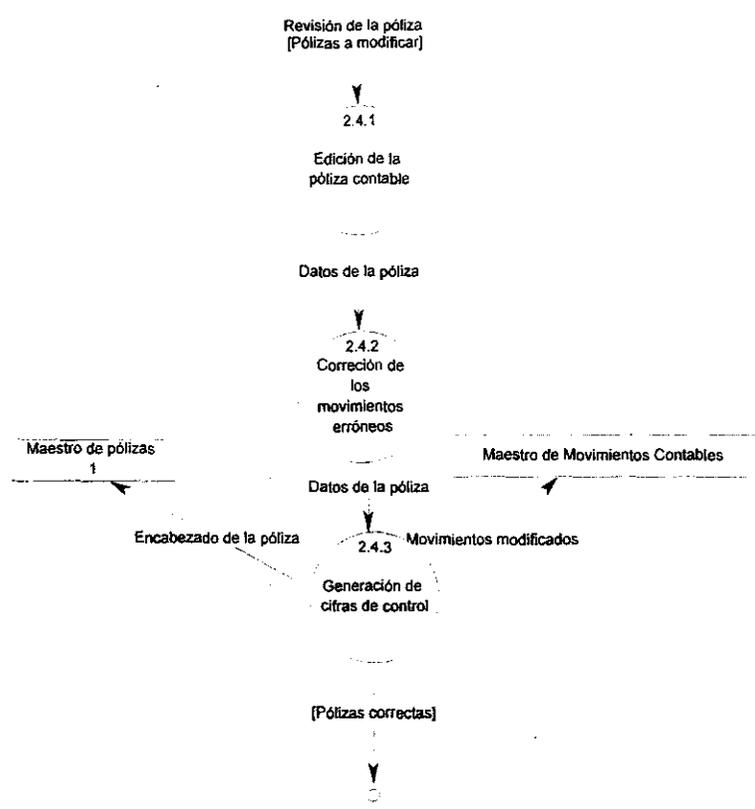


Figura 4.1.4.4. Subproceso modificación de pólizas

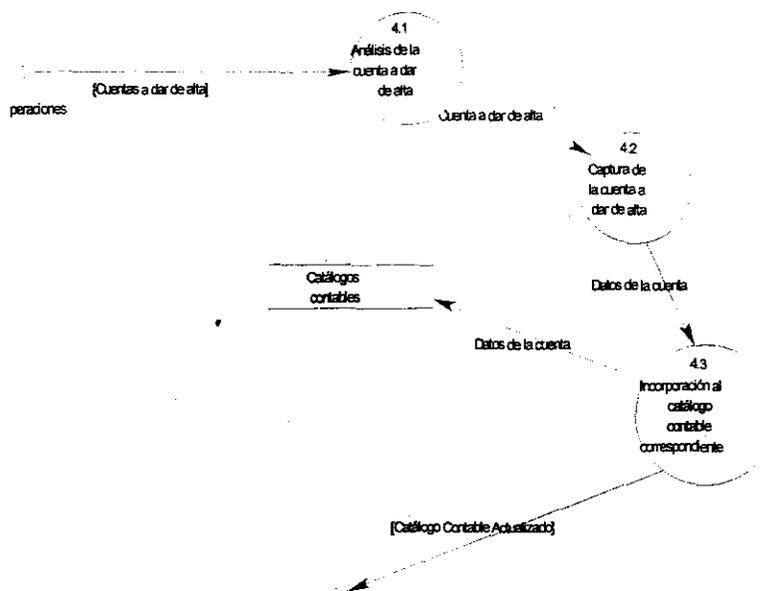


Figura 4.1.4.5. Subproceso alta de cuentas en el catálogo contable

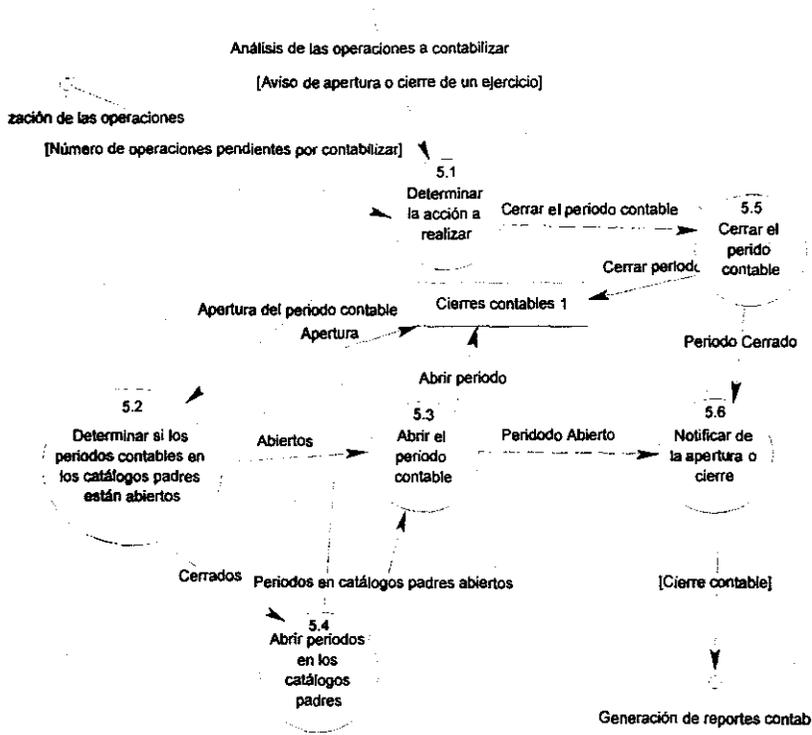


Figura 4.1.4.6. Subproceso Cierre o apertura del ejercicio contable

#### **4.1.5 Diagramas de eventos**

Otras herramientas dentro del desarrollo de sistemas, son los diagramas de eventos en los cuales se identifica cómo los diferentes eventos de la aplicación nos llevarán de un objeto a otro, además de mostrarnos el comportamiento de los objetos durante el procesamiento.

Al realizar el análisis de un sistema es importante que el desarrollador identifique plenamente la forma en que los eventos van formando un flujo que lleva de un objeto a otro.

Además de los diagramas de eventos, es importante que en el análisis se realicen los diagramas de transición de estados los cuales nos muestran la forma en que se comportará el sistema como consecuencia de sucesos externos. En estos diagramas se muestran los estados y los sucesos que generan que el sistema cambie de un estado a otro.

En términos generales, mientras que en el diagrama de eventos se muestra la forma en que los eventos nos llevan de un objeto a otro, en el de estados podemos observar cómo el sistema cambia de un estado a otro influenciado por diversos sucesos.

A continuación se mostrarán los principales diagramas de eventos de la aplicación.

En la figura 4.1.5.1 podemos observar los eventos principales que actúan sobre los objetos del módulo de seguridad.

La seguridad al momento de acceder un sistema es de vital importancia, cuando se trabajará en un ambiente multiusuario, debido a que cada usuario debe de poder ver sólo las opciones a las que tenga permiso. En este caso tenemos que el sistema presentará una pantalla en donde el usuario tecleará su clave de

acceso a fin de que el sistema valide si es una clave correcta, la clave que el sistema introduzca será encriptada, debido a que el sistema encripta la clave de acceso al dar de alta un nuevo usuario. Cuando la contraseña ha sido validada y se sabe que es una contraseña correcta, se procede a dejar ver al usuario el menú principal, a fin de que pueda seleccionar la opción con la que quiera trabajar.

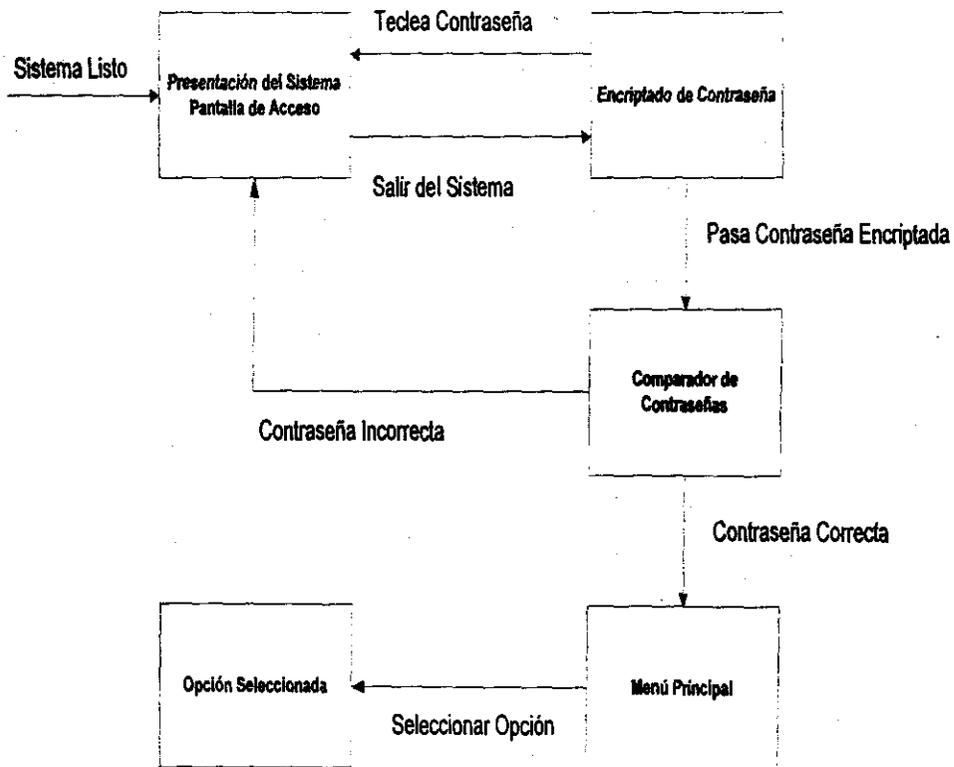


Figura 4.1.5.1. Diagrama de eventos módulo de seguridad

Para el módulo de manejo de cuentas contable, presentamos la gráfica de la figura 4.1.5.2., en la cual se muestra la forma en que los determinados eventos que el usuario activa, lo van llevando por los diferentes objetos que conforman el módulo.

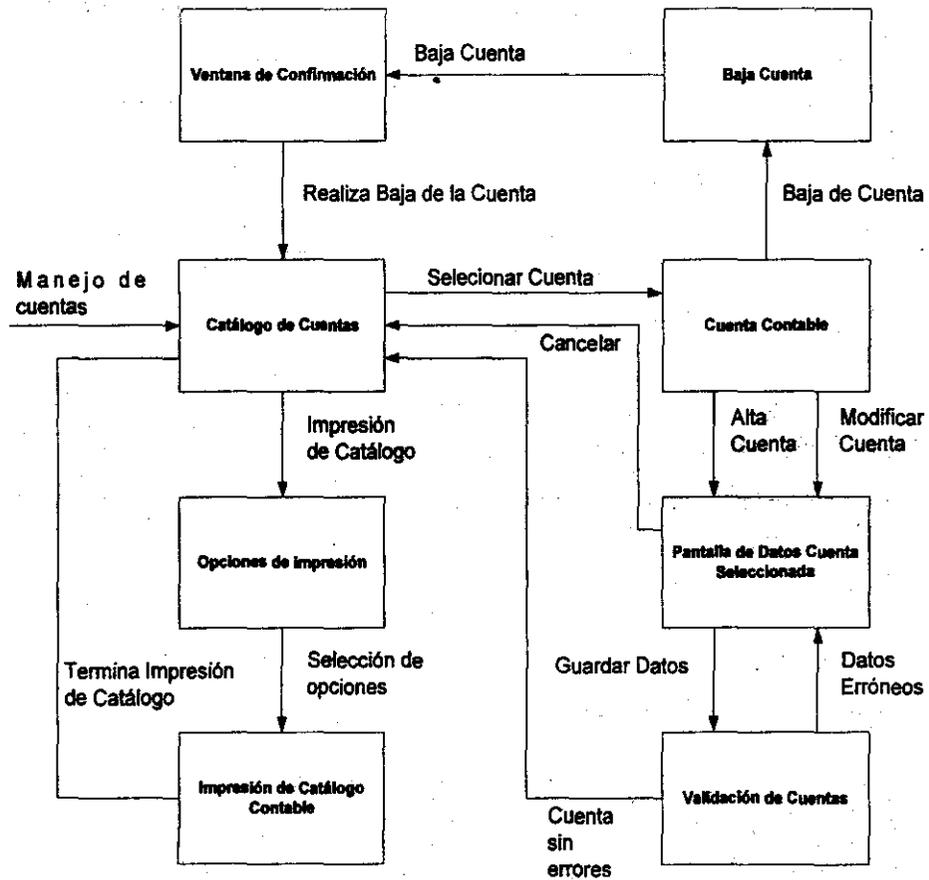


Figura 4.1.5.2. Diagrama de eventos módulo de manejo de cuentas contables.

Una vez que el usuario seleccionó desde el menú principal el módulo de manejo de cuentas contables, el sistema lo lleva a visualizar el catálogo correspondiente a su clave de acceso, hay que recordar que hay un catálogo contable por cada aeropuerto, por lo que el sistema le presentará el catálogo al que pertenezca el usuario; una vez en esta opción, el usuario puede seleccionar cualquier cuenta del catálogo, a la cual le podrá hacer una modificación o darla de baja, en caso de que se requiera modificar los datos de la cuenta, se presenta la pantalla en donde aparecen todos los datos de la cuenta, a fin de poder modificar el deseado, una vez que se ha modificado el o los datos pertinentes, el usuario al aceptar los datos, hace que se active la rutina de validación de la cuenta, la cual nos retornará a la pantalla de captura de datos en caso de encontrar alguna anomalía o nos llevará de nuevo a visualizar el catálogo contable; en caso de que lo que se requiera es dar de baja una cuenta, se pasa a una ventana de confirmación a fin de evitar que se den de baja cuentas en forma errónea.

Dentro de las opciones del manejo de cuentas contables, también encontramos el alta de cuentas, esta opción, al ser activada muestra una pantalla para la captura de los datos de la cuenta y una vez que el usuario procede a aceptar los datos, el sistema los valida, y si son correctos se procede a guardar la cuenta y regresar a visualizar el catálogo de cuentas.

En la figura 4.1.5.3 observamos el diagrama de eventos para el módulo de captura de pólizas contables.

El usuario debe de seleccionar la opción de administración de pólizas para poder realizar cualquier acción que conlleve una póliza contable, una vez accedida la opción, se tiene la posibilidad de dar de alta una nueva póliza seleccionando el alta de una póliza, con lo que se presentará la pantalla de captura de pólizas, en donde se capturarán los datos que conforman la póliza, para después al momento de salvar la información se validará a fin de regresar a

la pantalla de captura si alguno de los datos de la póliza son erróneos, o falta alguno, o bien regresar a la pantalla de administración de pólizas.

En la pantalla de administración de pólizas se muestran las pólizas que un usuario ha capturado, estas pueden ser susceptibles de modificación siempre y cuando no hayan sido ya contabilizadas, porque en ese caso sólo se mostrarán sus datos, mas no se podrán modificar. Si se selecciona una póliza para modificar, se mostrarán la pantalla de captura de pólizas, con los datos de la póliza a modificar, a fin de que el usuario haga las modificaciones pertinentes, y una vez realizadas estas, al momento de guardar la información se procede a validar esta, y si la validación es correcta, se regresa a la pantalla de administración de pólizas, y si no lo es, se regresa a la pantalla de captura de datos de la póliza.

Una póliza que no ha sido contabilizada puede ser borrada, para esto cuando el usuario selecciona la opción de borrar una póliza, se presenta una pantalla de confirmación de borrado, con la cual se evita que se borren accidentalmente pólizas.

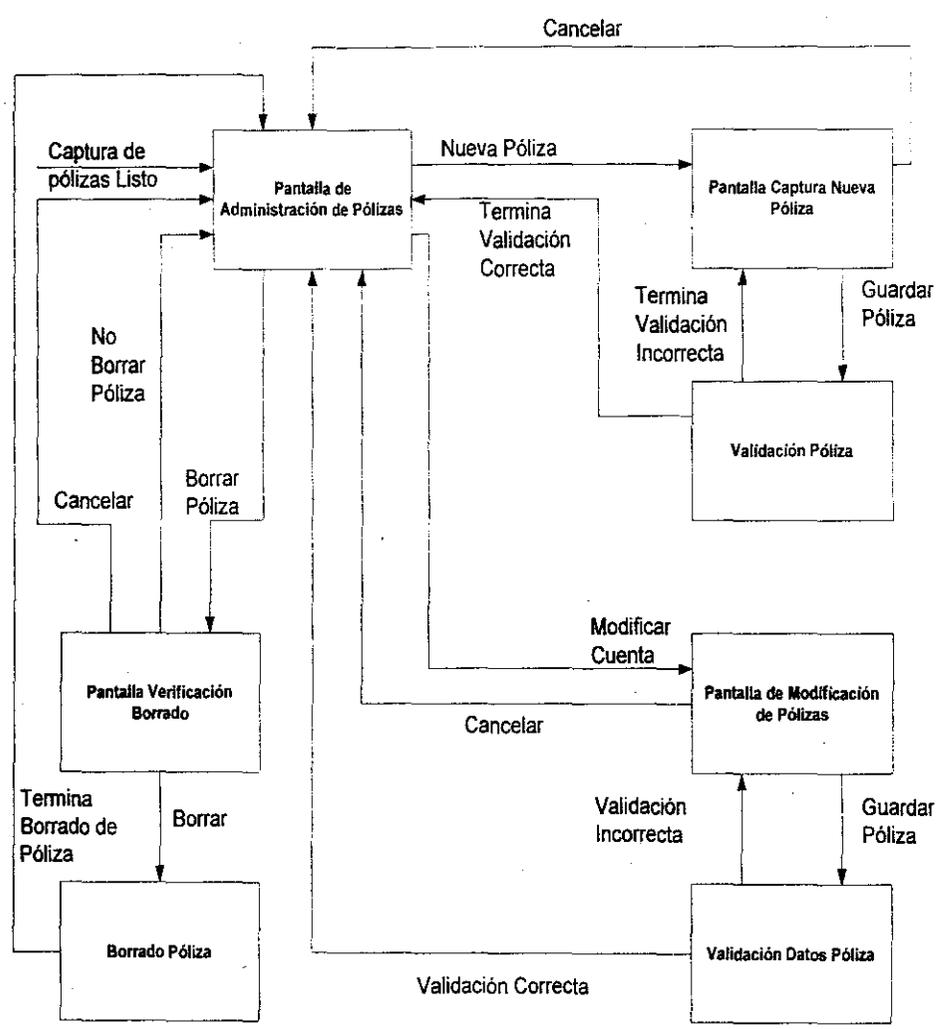


Figura 4.1.5.3. Diagrama de eventos módulo de captura de pólizas contables.

En la siguiente gráfica podemos observar el módulo de aplicaciones contables en el cual se muestran las pólizas que son susceptibles de ser contabilizadas, estas son marcadas por el usuario y contabilizadas. En caso de que alguna de las pólizas por algún motivo no pueda contabilizarse se genera un reporte que muestra la póliza que no pudo contabilizarse y el motivo, a fin de que se corrija el error.

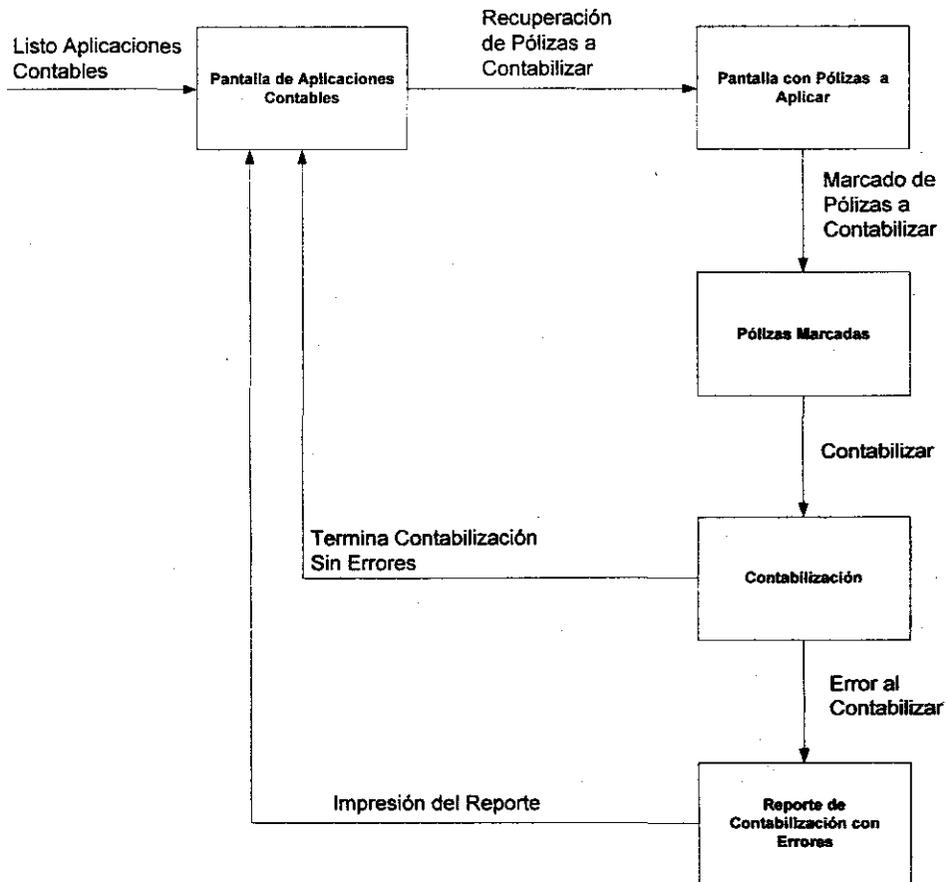


Figura 4.1.5.4. Diagrama de eventos módulo de aplicaciones contables.

En el módulo de consulta de cuentas, se generan la mayoría de los reportes contables con los que cuenta el sistema, el usuario sólo tiene que activar la lista de los reportes disponibles y seleccionar el reporte a generar para que el sistema active la lanzadora del reporte determinado, en la cual se pedirán algunos datos para la generación del reporte, y en el momento que el usuario los haya introducido, se generará el reporte.

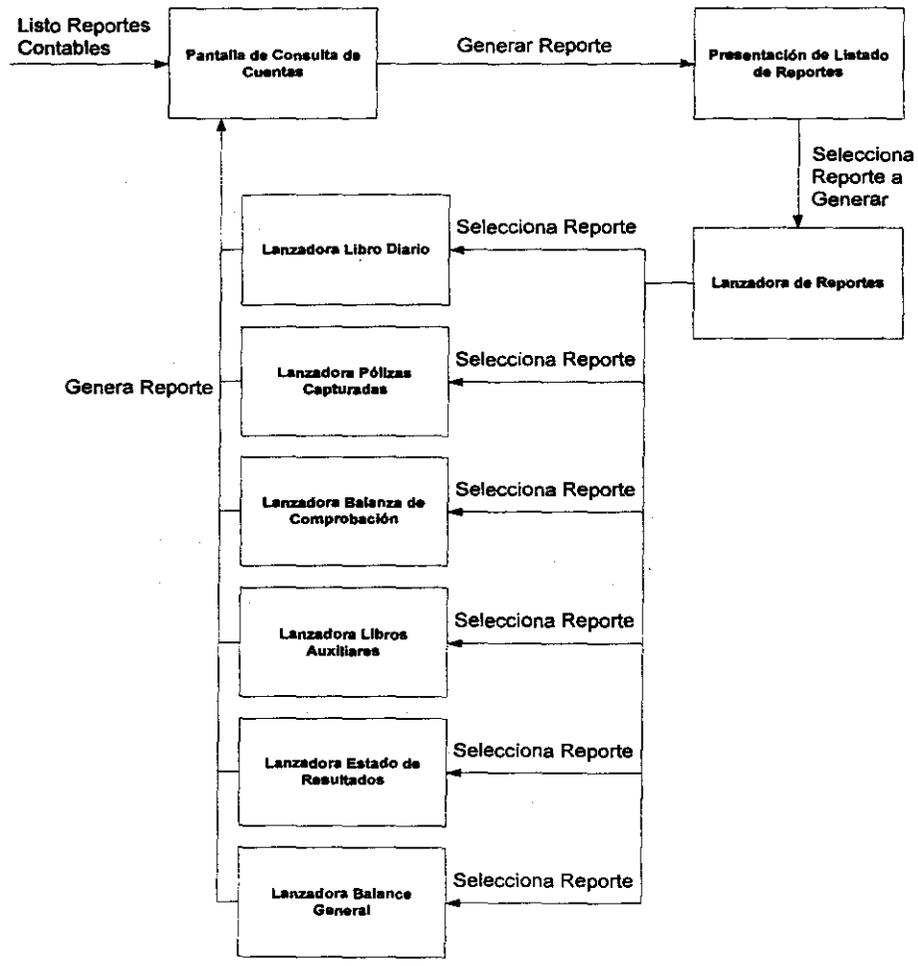


Figura 4.1.5.5. Diagrama de eventos módulo de reportes contables.

En el módulo de cierres contables, el usuario, selecciona la contabilidad a cerrar y el sistema le muestra el estado de la contabilidad en cuestión, a fin de que le permita al usuario abrir o cerrar un determinado período de la contabilidad.

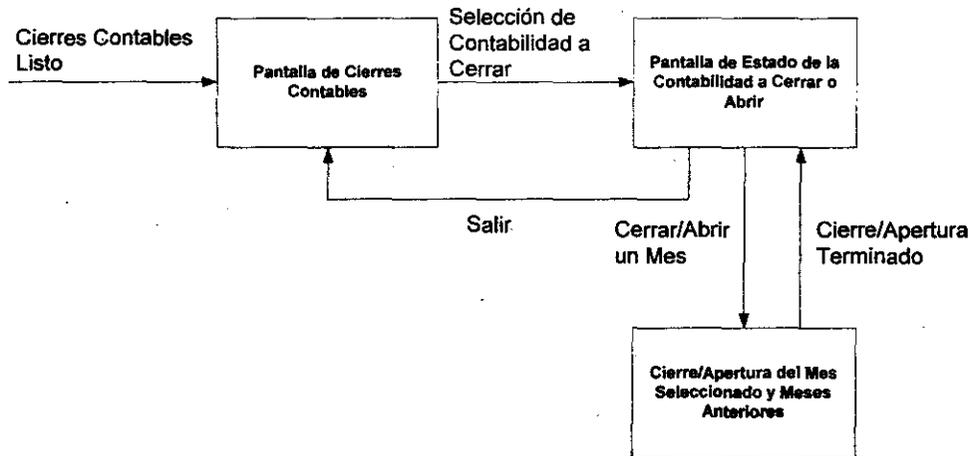


Figura 4.1.5.6. Diagrama de eventos módulo de cierres contables.

#### 4.1.6 Diagramas de estados.

En el punto anterior se explicó la importancia de los diagramas de estados y sus relaciones y diferencias con los diagramas de eventos. A continuación se muestran los diagramas de estados del sistema de contabilidad.

Primeramente mostramos el diagrama de estados del módulo de seguridad para el acceso de usuario a las opciones del menú.

En la pantalla de acceso al sistema se espera a que el usuario introduzca su contraseña y entonces se pasa al estado de encriptado de la misma, para después buscar la contraseña en la base de datos y si se encuentra esta se accesa el

sistema, y se muestran las opciones del menú, a fin de que se pueda seleccionar alguna de ellas.

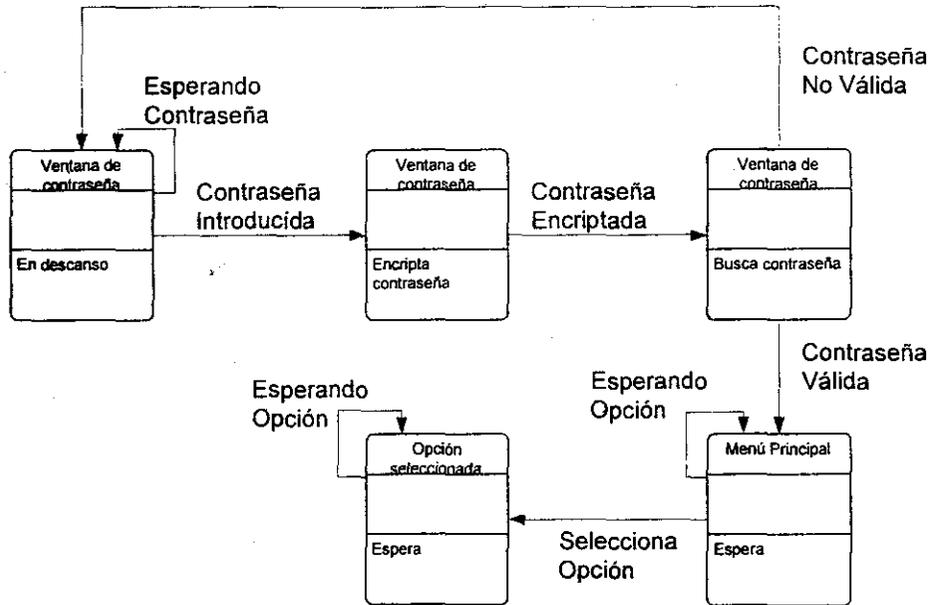


Figura 4.1.6.1. Diagrama de eventos para la pantalla de acceso al sistema del módulo de seguridad.

En la siguiente figura se muestra el diagrama de transición de estados del objeto catálogo contable, con el cual se maneja el alta, baja y modificación de cuentas.

Una vez que el usuario a entrado a la opción de manejo del catálogo contable tiene tres opciones, el hacer una alta, una modificación y una baja.

Analizaremos primero que es lo que sucede al hacer una baja de una cuenta contable. Cuando una determinada cuenta contable ha sido marcada y se ha

seleccionado la opción de baja de cuenta, el sistema empieza a determinar el saldo de la cuenta, debido a que una cuenta que tenga saldo diferente de cero no puede darse de baja, a fin de respetar las leyes contables, si la cuenta cumple con este requerimiento se pasa a determinar si tiene cuentas hijas, esto es que la cuenta forma parte del grupo de las normativas, esto es importante ya que una cuenta que tenga cuentas que dependan de ella, no podrá ser dada de baja, pero si la cuenta tiene un saldo de cero y no tiene cuentas dependientes, el sistema procede a dar de baja la cuenta.

Cuando se ha seleccionado la opción de alta de una cuenta, se presenta una ventana de captura de datos, en donde el usuario definirá las características de la cuenta que quiere dar de alta, una vez hecho esto, el sistema determina el tipo de cuenta a dar de alta, esto es, que sea una cuenta normativa o una cuenta de detalle, ya que la primera debe de existir en todos los catálogos contables de la red aeroportuaria, y la segunda sólo en el catálogo que está activo.

Al igual que el alta de una cuenta, la modificación de cuentas respeta el tipo de cuentas, es decir las modificaciones en una cuenta normativa se aplica en todas las contabilidades de todos los aeropuertos, mientras que la modificación de una cuenta de detalle, solo se hace en el catálogo contable activo.

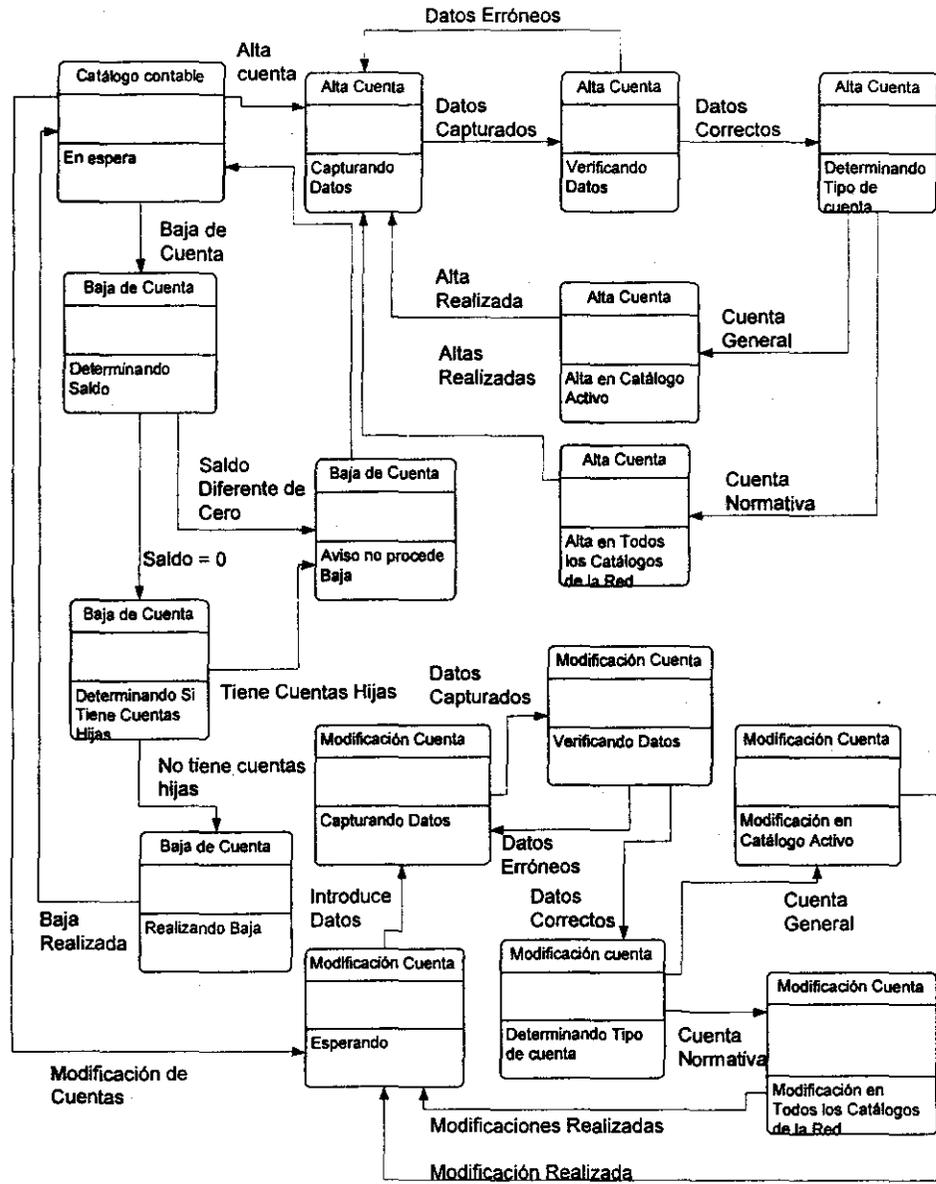


Figura 4.1.6.2. Diagrama de estados módulo de manejo de cuentas.

Para la aplicación de pólizas que es uno de los procesos mas importantes del sistema de contabilidad debido a que es en este momento en donde se afectan los libros de contabilidad, es necesario que se explique como trabaja el objeto de contabilización de pólizas.

Para contabilizar una o varias pólizas es necesario accesar la pantalla de contabilización de pólizas, y determinar los parámetros de recuperación de las pólizas, esto es a que mes perteneces, de que tipo son, el usuario que las creó o la fecha en que fueron creadas etc. Y una vez que estos parámetros han sido cambiados se procede a recuperar las pólizas, las cuales son susceptibles de ser marcadas para que inicie el proceso de contabilización, el iniciar el proceso, se empiezan a modificar los libros contables y la balanza de comprobación, si las pólizas no tuvieron ningún error al contabilizarse , el sistema emite un aviso de que se ha terminado el proceso de contabilización de manera normal, y en caso de que se detecte alguna anormalidad, se procede a presentar el reporte con las observaciones pertinentes.

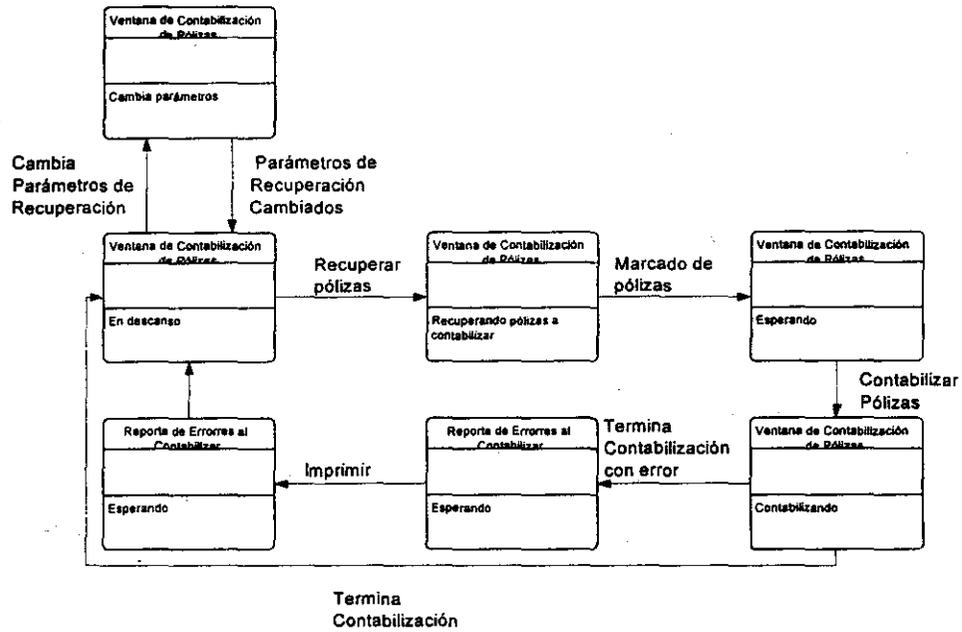


Figura 4.1.6.3. Diagrama de estados del módulo de contabilización de pólizas.

En la figura 4.1.6.4. podemos observar el diagrama de eventos del módulo de administración de pólizas, en donde empieza el flujo cuando se indica que se va crear una nueva póliza, para que la pantalla de captura de pólizas se habilite, y el usuario esté en posibilidad de capturar los datos de la póliza, y una vez que se ha capturado la información se valida la misma a fin de evitar errores tales como registros sin cuentas, sin montos, pólizas descuadradas, etc. , y si el sistema detecta que no hay ningún error la póliza es guardada. Dentro de la administración de cuentas también podemos encontrar la modificación de una póliza la cual sigue el mismo flujo que la alta de una nueva póliza, ya que los datos de la póliza modificada deben de ser validados. Por último tenemos la baja de pólizas contables, que verifica si la póliza a dar de baja se encuentra contabilizada, ya que

si esto es verdadero, no se puede borrar una póliza para no corromper la información de los libros contables, y en caso de que no se haya contabilizado aún sólo se verifica si es verdadera la petición de borrado y en caso afirmativo se procede a borrar la póliza.

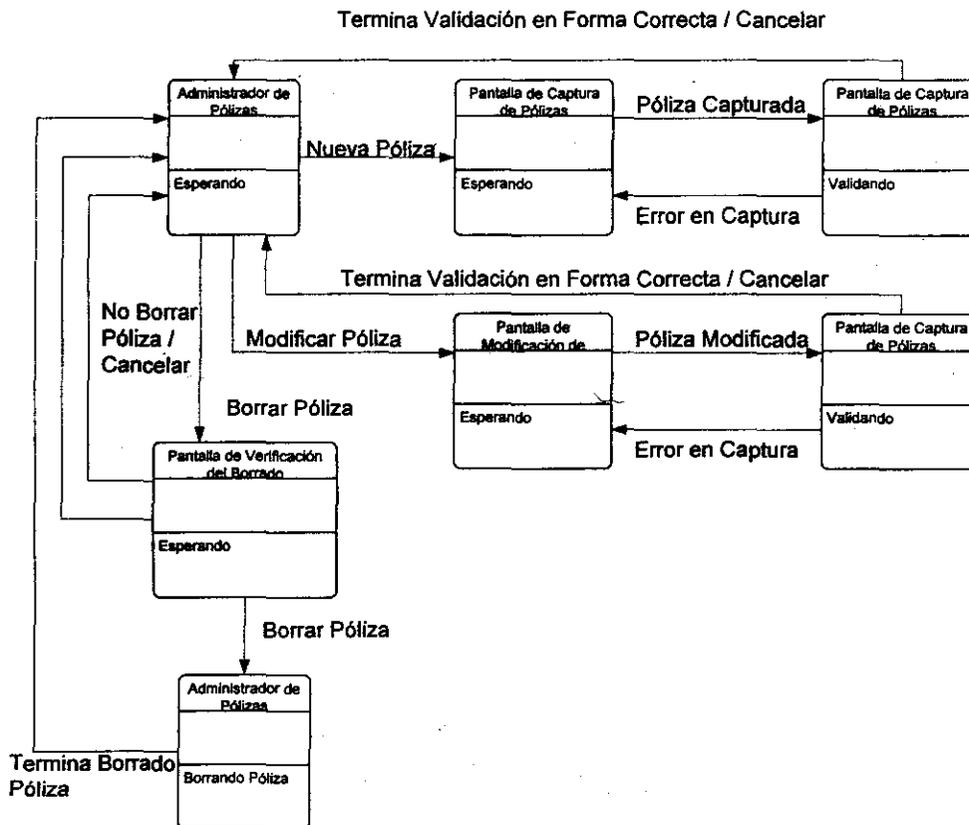


Figura 4.1.6.4 Diagrama de estados del módulo de administración de pólizas.

El diagrama de estados del módulo de reportes se encuentra esquematizado en la figura 4.1.6.5. en la cual podemos observar que el objeto de reportes de cuentas espera hasta que se de una petición de generación de un reporte, para esto hay que recordar que todos los reportes contables se generan con un mismo objeto, el cual analiza el tipo de reporte que se ha seleccionado a fin de establecer un primer conjunto de datos que servirán como parámetros del reporte, para con ellos poder abrir la ventana que funciona como lanzadora de reportes, en la cual se establecen un conjunto de parámetros que servirán para la generación del reporte el cual se dispara con la lanzadora y se muestra en pantalla, para después esperar la opción de impresión o de cancelación.

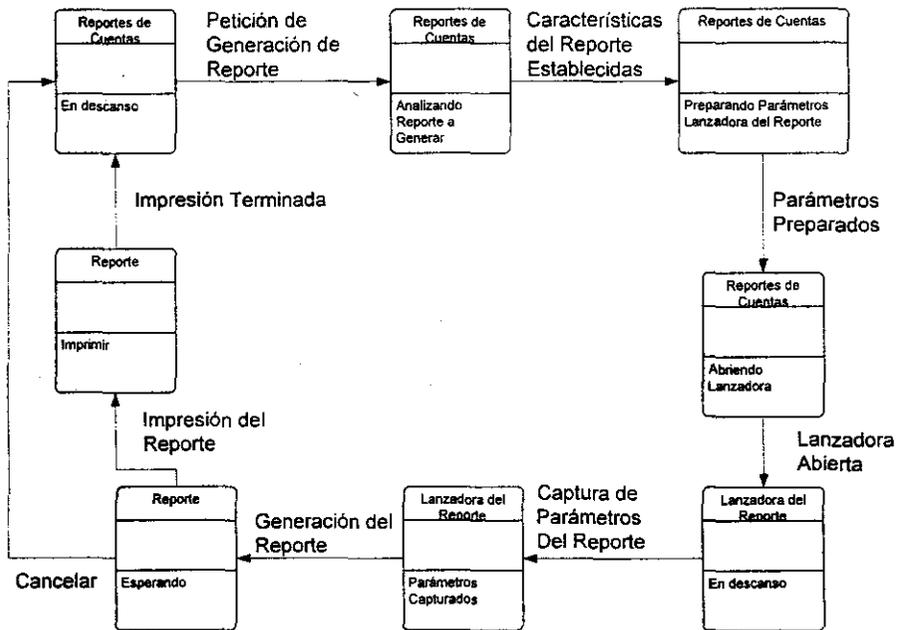


Figura 4.1.6.5 Diagrama de estados del módulo de reportes

## 4.2 GENERACIÓN DE CÓDIGO PARA EL PROCESAMIENTO DE LA INFORMACIÓN.

A continuación se presentan las rutinas más importantes que componen el sistema en la parte del cliente.

Código de las funciones más importantes del sistema.

//Función para obtener el mes actual

```
Datetime ldt_fecha_hora
String ls_fecha_hora
Integer li_año
```

```
ldt_fecha_hora = ctb_f_fecha_hora_servidor()
ls_fecha_hora = String(ldt_fecha_hora,"dd/mm/yyyy")
li_año = Integer(Mid(ls_fecha_hora,7,4))
```

```
Return li_año
end function
```

//Función para obtener los años contables que se tienen cargados en el sistema

```
Integer año,li_item_seleccionado,li_item_insertado,li_contador_años
Boolean lb_termino
```

```
DECLARE años_contables CURSOR FOR
SELECT año_cct_si
FROM ctb_c_cat_contable
GROUP BY año_cct_si;
```

```
OPEN años_contables;
```

```
lb_termino = false
li_contador_años = 0
```

```
Do WHILE NOT(lb_termino)
  FETCH años_contables INTO :año ;
  if SQLCA.sqlcode <> 0 then
    if SQLCA.sqlcode <> 100 then
      MessageBox("ERROR","No puedo determinar los años contables~r" + &
```

```

                                "Error: " + String(SQLCA.sqlcode) + "~r" + &
                                "Error DB: " + String(SQLCA.sqldbcode) + "~r" + &
                                SQLCA.sqlerrtext)
                                end if
                                lb_termino = true
                                else
                                li_contador_anios ++
                                aa_anios[li_contador_anios] = anio
                                end if
LOOP

CLOSE anios_contables;
ai_num_anios = li_contador_anios

Return

end subroutine

//Función para generar en la tabla de cierres contables, los meses contables de una
contabilidad

Boolean lb_realizo

li_estado = 0

for li_contador = li_mes_inicial to li_mes_final
    INSERT INTO ctb_m_cierre
    ( ctb_c_cat_contable,
      anio_cie_si,
      mes_cie_si)
    VALUES (:li_catalogo,
             :li_anio,
             :li_contador) ;
next
Return lb_realizo
end function

//Función para dar de alta las cuentas de un catálogo.
//Recibe como parámetro el catalogo que hay que dar de alta y el catálogo del cual
se van a copiar las cuentas

String ls_area_catalogo_nuevo,ls_area,ls_area_padre
Integer li_catalogo,li_cat_padre
long ll_cuenta,ll_cta_padre
Boolean lb_realizo,lb_termino
```

```

select asa_c_area
into :ls_area_catalogo_nuevo
from ctb_c_cat_contable
where ctb_c_cat_contable = :ai_catalogo_nuevo;

DECLARE cuentas_machote CURSOR FOR
SELECT ctb_c_cuenta.ctb_c_cat_contable,
       ctb_c_cuenta.asa_c_area,
       ctb_c_cuenta.ctb_c_cuenta,
       ctb_c_cuenta.cat_padre_i,
       ctb_c_cuenta.area_padre_str,
       ctb_c_cuenta.cta_padre_i
FROM ctb_c_cuenta
where ctb_c_cat_contable = :ai_catalogo_machote and
      (((es_det_cat_si = 1 and :ai_detalle = 1) or
        (es_det_cat_si = 0 and :ai_detalle = 1)) or
        (es_det_cat_si = 0 and :ai_detalle = 0))
ORDER BY cat_padre_i,area_padre_str,cta_padre_i;

OPEN cuentas_machote;
lb_termino = false
do while SQLCA.sqlcode = 0 and not(lb_termino)
  fetch cuentas_machote INTO
:li_catalogo,:ls_area,:li_cuenta,:li_cat_padre,:ls_area_padre,:ll_cta_padre;
  if SQLCA.sqlcode <> 0 then
    if SQLCA.sqlcode <> 100 then
      MessageBox("ERROR","Error al leer las cuentas del catálogo Machote ~r"
+ &
        "Error: " + String (SQLCA.sqlcode) + "~r" + &
        "Error: " + String (SQLCA.sqldbcode) + "~r" + &
        SQLCA.sqlerrtext)
    else
      lb_termino = true
    end if
  else
    lb_realizo = ctb_f_inserta_en_ctb_c_cuenta(ai_catalogo_nuevo, &
      ls_area_catalogo_nuevo,
      li_catalogo,&
      ls_area,&
      li_cuenta,&
      li_cat_padre,&
      ls_area_padre,&
      ll_cta_padre)
  if not (lb_realizo) then

```

```

        lb_termino = true
    end if
end if
loop

```

```

CLOSE cuentas_machote;

```

```

Return lb_realizo

```

```

end function

```

```

//Función para determinar qué catálogos dependen de un determinado catálogo
Boolean lb_hay_hijos,lb_continua_busqueda
Integer li_catalogos_hijos[],li_padres[],li_hijos[],li_total_insertados,li_apuntador
Integer li_catalogos_ordenados[],li_catalogo_auxiliar,li_contador_2
String ls_areas[],ls_area_ordenada[],ls_area_auxiliar
Integer
li_contador,li_contador_padres,li_contador_hijos,li_un_hijo,li_contador_catalogos_hijos
os
ctb_s_catalogo_area ls_areas_catalogos[],ls_auxiliar

```

```

lb_continua_busqueda = true
li_contador_catalogos_hijos = 0
li_contador_padres = 1
li_padres[1] = ai_catalogo
li_contador_hijos = 0

```

```

DO WHILE lb_continua_busqueda
    li_contador_hijos = 0
    FOR li_contador = 1 TO li_contador_padres
        DECLARE busca_hijos CURSOR FOR
        SELECT ctb_c_cat_contable.ctb_c_cat_contable
        FROM ctb_c_cat_contable
        WHERE ctb_c_cat_contable.padre_cct_si = :li_padres[li_contador];

```

```

    OPEN busca_hijos;

```

```

        lb_hay_hijos = true
        DO WHILE lb_hay_hijos
            FETCH busca_hijos INTO :li_un_hijo;
            if SQLCA.sqlcode = 0 then
                li_contador_hijos ++
                li_hijos[li_contador_hijos] = li_un_hijo
            elseif SQLCA.sqlcode = 100 then

```

```

        lb_hay_hijos = false
    else
        MessageBox("ERROR", "No puedo encontrar los hijos de un
catálogo~r" + &
                "Error: " + String(SQLCA.sqlcode) + "~r" + &
                "ErrorDB: " +
String(SQLCA.sqldbcode) + "~r" + &
                SQLCA.sqlerrtext)
        lb_hay_hijos = false
    end if
LOOP

    CLOSE busca_hijos;
NEXT
//todos los hijos encontrados se almacenan en el arreglo de catalogos hijos
FOR li_contador = 1 TO li_contador_hijos
    li_contador_catalogos_hijos ++
    li_catalogos_hijos[li_contador_catalogos_hijos] = li_hijos[li_contador]
NEXT
//Limpiamos el arreglo de padres, ya que los hijos encontrados ahora son los
padres
FOR li_contador=1 TO li_contador_padres
    SetNull(li_padres[li_contador])
NEXT
//Pasamos los hijos encontrados del arreglo hijos al de padres
li_contador_padres = 0
FOR li_contador=1 TO li_contador_hijos
    li_contador_padres ++
    li_padres[li_contador_padres] = li_hijos[li_contador]
NEXT
//y por último limpiamos el arreglo de hijos
FOR li_contador = 1 TO li_contador_hijos
    SetNull(li_hijos[li_contador])
NEXT
li_contador_hijos = 0

IF li_contador_padres > 0 THEN
    lb_continua_búsqueda = True
ELSE
    lb_continua_búsqueda = False
END IF
LOOP

//hasta aquí tenemos los catalogos hijos y el catálogo padre por separado

//hay que obtener las áreas de los catálogos hijos
for li_contador = 1 to li_contador_catalogos_hijos

```

```

Select asa_c_area
  into :ls_areas_catalogos[li_contador].area
  from ctb_c_cat_contable
  where ctb_c_cat_contable = :li_catalogos_hijos[li_contador];
if SQLCA.sqlcode <> 0 then
  MessageBox("ERROR", "No puedo encontrar los la áreas de los catálogos
para poder ordenarlos~r" + &
              "Error: " + String(SQLCA.sqlcode) + "~r" + &
              "ErrorDB: " +
              String(SQLCA.sqldbcode) + "~r" + &
              SQLCA.sqlerrtext)
end if
  ls_areas_catalogos[li_contador].catalogo = li_catalogos_hijos[li_contador]
next

//Ahora hay que ordenar todos los registros de los catálogos hijos
for li_contador = 1 to li_contador_catalogos_hijos - 1
  for li_contador_2 = 1 to li_contador_catalogos_hijos - 1
    if ls_areas_catalogos[li_contador_2].area >
ls_areas_catalogos[li_contador_2 + 1].area then
      ls_auxiliar = ls_areas_catalogos[li_contador_2]
      ls_areas_catalogos[li_contador_2] =
ls_areas_catalogos[li_contador_2 + 1]
      ls_areas_catalogos[li_contador_2 + 1] = ls_auxiliar
    end if
  next
next

for li_contador = 1 to li_contador_catalogos_hijos
  li_catalogos_hijos[li_contador] = ls_areas_catalogos[li_contador].catalogo
next

//colocamos el catálogo que llegó como argumento de la función al principio del
arreglo
FOR li_contador = li_contador_catalogos_hijos + 1 TO 1 STEP -1
  if li_contador = 1 then
    li_catalogos_hijos[li_contador] = ai_catalogo
  else
    li_catalogos_hijos[li_contador] = li_catalogos_hijos[li_contador - 1]
  end if
NEXT

ai_num_cat      = li_contador_catalogos_hijos + 1
aa_catalogos_hijos[] = li_catalogos_hijos[]
Return

```

```
end subroutine
```

```
//Función para determinar si una determinada contabilidad es normativa
Integer li_catalogo
```

```
SELECT ctb_c_cat_contable.ctb_c_cat_contable
INTO :li_catalogo
FROM ctb_c_cat_contable
WHERE ( ctb_c_cat_contable.asa_c_area = :as_area ) AND
( ctb_c_cat_contable.norma_cct_si = 1 ) ;
```

```
if SQLCA.sqlcode = 100 then
    lb_es_normativo = false
elseif SQLCA.sqlcode = 0 then
    lb_es_normativo = True
else
    MessageBox("ERROR","No puedo determinar si el area es normativa")
end if
```

```
REturn lb_es_normativo
end function
```

```
//Función para determinar si una contabilidad es concentradora
```

```
integer li_concentrador
```

```
Select concen_cct_si
into :li_concentrador
from ctb_c_cat_contable
where ctb_c_cat_contable = :ai_id_catalogo;
```

```
return (li_concentrador = 1)
end function
```

```
//Función para extraer la fecha y la hora del servidor
```

```
Datetime ldt_fechahora_servidor
```

```
DECLARE fecha_servidor PROCEDURE FOR ctb_sp_fechahora_servidor ;
```

```
EXECUTE fecha_servidor;
```

```
FETCH fecha_servidor INTO :ldt_fechahora_servidor;
if SQLCA.sqlcode <> 0 then
```

```
Messagebox("ERROR","Error al obtener la fecha del servidor~r" + &
"Error: " + String(SQLCA.sqlcode) + "~r" + &
"ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
SQLCA.sqlerrtext)
end if

CLOSE fecha_servidor;
Return Idt_fechahora_servidor;
end function
```

```
//Función para insertar un registro en la tabla de cuentas
boolean lb_realizo
```

```
lb_realizo = true

if not(isnull(ai_cat_padre)) then
    ai_cat_padre = ai_catalogo_nuevo
end if
if not(isnull(as_area_padre)) then
    as_area_padre = as_area_catalogo_nuevo
end if
```

```
INSERT INTO ctb_c_cuenta
( ctb_c_cat_contable,
  asa_c_area,
  ctb_c_cuenta,
  cat_padre_i,
  area_padre_str,
  cta_padre_i,
  ctb_c_divisas,
  sys_iconos,
  nivel_cat_si,
  tipo1_cat_str,
  tipo2_cat_str,
  naturaleza_cat_str,
  cuenta_cat_str,
  es_det_cat_si,
  descrip_cat_str,
  bloquea_cat_str,
  prop_cat_str,
  sobregi_cat_str,
  niv_sob_cat_n,
  hr_aper_cat_dt,
  hr_cie_cat_dt,
```

```

    estado_cat_str,
    asa_c_aerorig,
    asa_c_aerodest )
SELECT :ai_catalogo_nuevo,
       :as_area_catalogo_nuevo,
       ctb_c_cuenta.ctb_c_cuenta,
       :ai_cat_padre,
       :as_area_padre,
       :ai_cta_padre,
       ctb_c_cuenta.ctb_c_divisas,
       ctb_c_cuenta.sys_iconos,
       ctb_c_cuenta.nivel_cat_si,
       ctb_c_cuenta.tipo1_cat_str,
       ctb_c_cuenta.tipo2_cat_str,
       ctb_c_cuenta.naturaleza_cat_str,
       ctb_c_cuenta.cuenta_cat_str,
       ctb_c_cuenta.es_det_cat_si,
       ctb_c_cuenta.descrip_cat_str,
       ctb_c_cuenta.bloquea_cat_str,
       ctb_c_cuenta.prop_cat_str,
       ctb_c_cuenta.sobregi_cat_str,
       ctb_c_cuenta.niv_sob_cat_n,
       ctb_c_cuenta.hr_aper_cat_dt,
       ctb_c_cuenta.hr_cie_cat_dt,
       ctb_c_cuenta.estado_cat_str,
       ctb_c_cuenta.asa_c_aerorig,
       ctb_c_cuenta.asa_c_aerodest
FROM ctb_c_cuenta
where ctb_c_cat_contable = :ai_cat_machote and
      asa_c_area         = :as_area_cat_machote and
      ctb_c_cuenta       = :ai_cta_cat_machote;

if SQLCA.sqlcode <> 0 then
  MessageBox("ERROR","Error al dar de alta las cuentas de un nuevo catalogo
(ctb_c_cuenta)~r" + &
    "Catalogo: " + String(ai_cat_machote) + "~r" + &
    "Area: "    + String(as_area_cat_machote) + "~r" + &
    "Cuenta "  + String(ai_cta_cat_machote) + "~r" + &
    "Error: "  + String (SQLCA.sqlcode) + "~r" + &
    "Error: "  + String (SQLCA.sqldbcode) + "~r" + &
    SQLCA.sqlerrtext)
  lb_realizo = false
end if

if lb_realizo then
  insert into ctb_m_cuenta
    (ctb_m_cuenta.ctb_c_cat_contable,

```

```

        ctb_m_cuenta.asa_c_area,
        ctb_m_cuenta.ctb_c_cuenta)
    select ctb_c_cuenta.ctb_c_cat_contable,
    ctb_c_cuenta.asa_c_area,
        ctb_c_cuenta.ctb_c_cuenta
    from ctb_c_cuenta
    where ctb_c_cat_contable = :ai_catalogo_nuevo and
          asa_c_area       = :as_area_catalogo_nuevo and
          ctb_c_cuenta     = :al_cta_cat_machote;
    if SQLCA.sqlcode <> 0 then
    MessageBox("ERROR", "Error al dar de alta las cuentas de un nuevo catalogo
(ctb_m_cuenta)~r" + &
        "Catalogo: " + String(ai_cat_machote) + "~r" + &
        "Area: "   + String(as_area_cat_machote) + "~r" + &
        "Cuenta "  + String(al_cta_cat_machote) + "~r" + &
        "Error: "  + String(SQLCA.sqlcode) + "~r" + &
        "Error: "  + String(SQLCA.sqldbcode) + "~r" + &
        SQLCA.sqlerrtext)
        lb_realizo = false
    end if
end if

return lb_realizo
end function

```

//Función para determinar la longitud de una cuenta contable  
Integer li\_longitud\_cuenta

```

SELECT sum(ctb_d_tpocat.num_carac_dtc_si)
INTO :li_longitud_cuenta
FROM ctb_c_cat_contable,
     ctb_d_tpocat
WHERE ( ctb_c_cat_contable.ctb_c_tpocat = ctb_d_tpocat.ctb_c_tpocat ) and
      ( ( ctb_c_cat_contable.ctb_c_cat_contable = :ai_catalogo ) ) ;

if SQLCA.sqlcode <> 0 then
    MessageBox("ERROR", "No puedo determinar la longitud de las cuentas ~r" + &
        "Error: " + String(SQLCA.sqldbcode) + "~r" + &
        SQLCA.sqlerrtext)
end if

Return li_longitud_cuenta

```

```

end function

//Función para obtener la máscara que se utilizará para presentar las cuentas
//en la pantalla

integer li_tipo_de_catalogo,li_numero_de_niveles,contador
integer li_numero_de_caracteres
String mascara

Select ctb_c_tpocat
  into :li_tipo_de_catalogo
  from ctb_c_cat_contable
  where ctb_c_cat_contable = :ai_num_catalogo;

Select numniv_tct_si
  into :li_numero_de_niveles
  from ctb_c_tpocat
  where ctb_c_tpocat = :li_tipo_de_catalogo;

mascara = ""

for contador = 1 to li_numero_de_niveles
  Select num_carac_dtc_si
    into :li_numero_de_caracteres
    from ctb_d_tpocat
    where ((nivel_dtc_si = :contador) and (ctb_c_tpocat =
:li_tipo_de_catalogo));
  mascara = mascara + fill(as_caracter,li_numero_de_caracteres)
  if contador <> li_numero_de_niveles then
    mascara = mascara + '-'
  end if
next

return mascara
end function

//Función para obtener el mes actual
Datetime ldt_fecha_hora
String ls_fecha_hora
Integer li_mes

ldt_fecha_hora = ctb_f_fecha_hora_servidor()
ls_fecha_hora = String(ldt_fecha_hora,"dd/mm/yyyy")

```

```
li_mes = Integer(Mid(ls_fecha_hora,4,2))
```

```
Return li_mes  
end function
```

```
//Función para obtener el mes como una cadena  
String ls_mes
```

```
CHOOSE CASE ia_mes  
CASE 1  
    ls_mes = "ENERO"  
CASE 2  
    ls_mes = "FEBRERO"  
CASE 3  
    ls_mes = "MARZO"  
CASE 4  
    ls_mes = "ABRIL"  
CASE 5  
    ls_mes = "MAYO"  
CASE 6  
    ls_mes = "JUNIO"  
CASE 7  
    ls_mes = "JULIO"  
CASE 8  
    ls_mes = "AGOSTO"  
CASE 9  
    ls_mes = "SEPTIEMBRE"  
CASE 10  
    ls_mes = "OCTUBRE"  
CASE 11  
    ls_mes = "NOVIEMBRE"  
CASE 12  
    ls_mes = "DICIEMBRE"
```

```
END CHOOSE
```

```
Return ls_mes  
end function
```

```
//Función para obtener los niveles contables de una determinada contabilidad  
Integer li_catalogo,li_niveles_catalogo  
Boolean lb_nivel_correcto
```

```

lb_nivel_correcto = false

//Obtenemos el número de niveles que maneja el catálogo al cual se le quiere hacer
el
//alta
SELECT ctb_c_tpocat.numniv_tct_si
      INTO :li_niveles_catalogo
      FROM ctb_c_cat_contable,ctb_c_tpocat
      WHERE (ctb_c_tpocat.ctb_c_tpocat = ctb_c_cat_contable.ctb_c_tpocat) and
            (ctb_c_cat_contable.ctb_c_cat_contable = :ai_id_catalogo);

return li_niveles_catalogo
end function

//Función para obtener un nuevo número de póliza
long nuevo_numero_poliza

select max(num_pol_i) + 1
      into :nuevo_numero_poliza
      from ctb_m_poliza
      where ctb_c_cat_contable = :ai_catalogo and
            tipo_pol_str      = :as_tipo_poliza and
            mes_pol_si       = :ai_mes;

if !null(nuevo_numero_poliza) then nuevo_numero_poliza = 1

if SQLCA.sqlcode <> 0 then
  MessageBox("ERROR","No puedo determinar el número de póliza" + "~r" + &
    "Error: " + String(SQLCA.sqlcode)+ "~r" + &
    "Error DB" + String(SQLCA.sqldbcode) + "~r" + &
    SQLCA.sqlerrtext)
  nuevo_numero_poliza = -1
end if

return nuevo_numero_poliza
end function

//Función que en base al catálogo que se pasa como parámetro se devuelve los
catálogos de esa misma área y del mismo tipo de catálogo pero de años anteriores

```

```

//*****
//El siguiente es el código que hay que colocar para que la función trabaje
//
//Any li_catalogos_anteriores[],ls_areas_ant[]
//Integer li_cat_anteriores[],li_num_cat,li_catalogo
//String ls_areas_anteriores[]
//
//ctb_f_obten_cat_anios_anteriores(li_catalogo,li_num_cat,li_catalogos_anteriores[])
//li_cat_anteriores[] = li_catalogos_anteriores[]
//ls_areas_anteriores[] = ls_areas_ant[]
//*****

String ls_asa_c_area,ls_asa_c_aeropuerto,ls_area,ls_areas_anteriores[]
Integer
li_ctb_c_tpocat,li_anio_cct_si,li_norma_cct_si,li_concen_cct_si,li_contasa_cct_si
Integer li_catalogos_anteriores[],li_numero_de_catalogos,li_catalogo
Boolean lb_termino,lb_realizo

//Primero obtenemos todos los datos que identifican al catálogo que se pasa como
parámetro

lb_realizo = true
Select asa_c_area,
      ctb_c_tpocat,
      asa_c_aeropuerto,
      anio_cct_si,
      norma_cct_si,
      concen_cct_si,
      contasa_cct_si
into :ls_asa_c_area,
     :li_ctb_c_tpocat,
     :ls_asa_c_aeropuerto,
     :li_anio_cct_si,
     :li_norma_cct_si,
     :li_concen_cct_si,
     :li_contasa_cct_si
from ctb_c_cat_contable
where ctb_c_cat_contable = :ai_catalogo;

if SQLCA.sqlcode <> 0 then
MessageBox("ERROR", "Error al determinar los datos del catálogo del cual se
quieren ~r" + &"obtener los catálogos similares de años anteriores ~r" + &
"Error: " + String(SQLCA.sqlcode) + "~r" + &
"Errordb: " + String(SQLCA.sqldbcode) + "~r" + &
SQLCA.sqlerrtext)
lb_realizo = false

```

```
end if
```

```
//una vez que se han determinado las características del catálogo que se pasó como
//parámetro, se procede a determinar cuales son los catálogos de las contabilidades
//de años anteriores
```

```
DECLARE contabilidades CURSOR FOR
```

```
  Select ctb_c_cat_contable,
         asa_c_area
  from ctb_c_cat_contable
  where asa_c_area      = :ls_asa_c_area and
         ctb_c_tpcat   = :li_ctb_c_tpcat and
         asa_c_aeropuerto = :ls_asa_c_aeropuerto and
         norma_cct_si   = :li_norma_cct_si and
         concen_cct_si  = :li_concen_cct_si and
         contasa_cct_si = :li_contasa_cct_si;
```

```
OPEN contabilidades;
```

```
lb_termino = false
```

```
li_numero_de_catalogos = 0
```

```
DO WHILE SQLCA.sqlcode = 0 and not(lb_termino)
```

```
  FETCH contabilidades INTO :li_catalogo,:ls_area;
```

```
  if SQLCA.sqlcode <> 0 then
```

```
    if SQLCA.sqlcode = 100 then
```

```
      lb_termino = true
```

```
    else
```

```
      MessageBox("ERROR", "Error al determinar los catálogos de años anteriores ~r" + &
```

```
        "Error: " + String(SQLCA.sqlcode) + "~r" + &
```

```
        "Errordb: " + String(SQLCA.sqldbcode) + "~r" + &
```

```
        SQLCA.sqlerrtext)
```

```
      lb_realizo = false
```

```
      lb_termino = true
```

```
    end if
```

```
  else
```

```
    li_numero_de_catalogos ++
```

```
    li_catalogos_anteriores[li_numero_de_catalogos] = li_catalogo
```

```
    ls_areas_anteriores[li_numero_de_catalogos] = ls_area
```

```
  end if
```

```
loop
```

```
CLOSE contabilidades;
```

```
ai_num_cat = li_numero_de_catalogos
```

```
aa_catalogos[] = li_catalogos_anteriores[]
```

```
aa_areas[] = ls_areas_anteriores[]
```

```
RETURN lb_realizo
```

```
end function
```

```
//Función para obtener los catálogos a los cuales se va a contabilizar dependiendo  
del área  
//que se pasa como parámetro. Se regresa como resultado los números de catálogos  
a los  
//cuales aplicar en el arreglo que se pasó como referencia  
//y si no se encuentra ninguno se retorna un -1
```

```
//se pasa como parámetro el área de la cual se quiera saber qué catálogo le  
corresponde  
//no todas las áreas tienen un catálogo al cual aplicar pero por el lugar donde se  
//encuentren dentro de la estructura, se determina el catálogo, ya que a lo mejor ellas  
no  
//tienen catálogo, pero sus ascendientes sí
```

```
//se parte de la premisa de que la dependencia de áreas se da por la nomenclatura de  
cada  
//área esto es:  
// ejemplo:
```

```
// SD02 es el padre de SD0201 y este a su vez es el padre de SD020101
```

```
// se van tomando de dos en dos los caracteres de la derecha y se busca si el área  
// resultante se encuentra en la tabla de catálogos, cuando la longitud de la cadena  
llega  
// a dos sólo se toma un caracter
```

```
*****  
//Instrucciones que deben colocarse en el código de donde se llame esta función  
*****  
//Integer li_num_cat, contador, li_cat[]  
//Any li_catalogos[]  
//String ls_area  
//li_num_cat = ctb_f_obten_catálogo_a_contabilizar(ls_area, li_catalogos[])  
//li_cat = li_catalogos  
//  
//for contador = 1 to li_num_cat  
//  MessageBox("AVISO", li_cat[contador])  
//next  
*****
```

```
String ls_area_ceros,ls_area
Boolean lb_procesa,lb_tiene_catalogo,lb_fin_de_cadena,lb_termino
Integer li_catalogo_a_aplicar,li_quitar,li_catalogos_obtenidos[],li_contador_cat
Integer li_anio_actual
```

```
ls_area      = as_area
ls_area_ceros  = as_area
lb_tiene_catalogo = false
lb_fin_de_cadena = false
lb_procesa    = true
li_anio_actual = ctb_f_anio_actual()
```

```
do while not(lb_fin_de_cadena) and not(lb_tiene_catalogo) and lb_procesa
```

```
  declare catalogos_a_aplicar cursor for
  Select ctb_c_cat_contable
  from ctb_c_cat_contable
  where (asa_c_area = :ls_area or
  asa_c_area = :ls_area_ceros) and
  concen_cct_si = 0 and
  anio_cct_si = :li_anio_actual;
```

```
  open catalogos_a_aplicar;
```

```
  li_contador_cat = 0
```

```
  lb_termino = false
```

```
  do while not(lb_termino)
```

```
    fetch catalogos_a_aplicar into :li_catalogos_obtenidos[li_contador_cat + 1];
```

```
    if SQLCA.sqlcode <> 0 then
```

```
      if SQLCA.sqlcode <> 100 then
```

```
        MessageBox("ERROR", "Error al acceder la tabla de catálogos para determinar ~r" +
        &
```

```
          "el catálogo al cual se aplica ~r" + &
```

```
          "Error: " + String(SQLCA.sqlcode) + &
```

```
          SQLCA.sqlerrtext)
```

```
        lb_procesa = false
```

```
        lb_termino = true
```

```
      else
```

```
        lb_termino = true
```

```
      end if
```

```
    else
```

```
      lb_tiene_catalogo = true
```

```

        li_contador_cat ++
    end if
Loop

close catalogos_a_aplicar;

if len(ls_area) >= 4 then
    li_quitar = 2
else
    li_quitar = 1
end if

if (len(ls_area) - li_quitar) > 0 then
    ls_area = Mid(ls_area,1,len(ls_area)-li_quitar)
    ls_area_ceros = Replace(ls_area_ceros,len(ls_area) + 1 ,li_quitar,"00")
else
    lb_procesa = false
end if
loop

if li_contador_cat = 0 or Isnull(li_contador_cat) then
    li_contador_cat = -1
end if

aa_catalogos_obtenidos = li_catalogos_obtenidos

Return li_contador_cat
end function

```

//Función que obtiene los catálogos contables de un aeropuerto  
//En base al catálogo que se pasa como parámetro se devuelven los catálogos del  
//aeropuerto y se regresan de todos los años que se tengan cargados en el sistema

```

//*****
//El siguiente es el código que hay que colocar para que la función trabaje
//
//Any li_catalogos_anteriores[],ls_areas_ant[]
//Integer li_cat_anteriores[],li_num_cat,li_catalogo
//String ls_areas_anteriores[]
//
//ctb_f_obten_cat_anios_anteriores(li_catalogo,li_num_cat,li_catalogos_anteriores[])
//li_cat_anteriores[] = li_catalogos_anteriores[]
//ls_areas_anteriores[] = ls_areas_ant[]
//*****

```

```

String ls_asa_c_area,ls_asa_c_aeropuerto,ls_area,ls_areas_anteriores[]
Integer
li_ctb_c_tpcat,li_anio_cct_si,li_norma_cct_si,li_concen_cct_si,li_contasa_cct_si
Integer li_catalogos_anteriores[],li_numero_de_catalogos,li_catalogo
Boolean lb_termino,lb_realizo

//Primero obtenemos de que aeropuerto es el catálogo que se pasa como parámetro

lb_realizo = true
Select asa_c_aeropuerto
into :ls_asa_c_aeropuerto
from ctb_c_cat_contable
where ctb_c_cat_contable = :ai_catalogo;

if SQLCA.sqlcode <> 0 then
Messagebox("ERROR", "Error al determinar los datos del catálogo del cual se
quieren ~r" + &
"obtener los catálogos similares de años anteriores ~r" + &
"Error: " + String(SQLCA.sqlcode) + "~r" + &
"Errordb: " + String(SQLCA.sqldbcode) + "~r" + &
SQLCA.sqlerrtext)
lb_realizo = false
end if

//una vez que se han determinado cual es el aeropuerto del catálogo que se pasó
como
//parámetro, se procede a determinar cuales son los catálogos que están en la tabla
//de catálogos y que pertenecen al mismo aeropuerto
DECLARE contabilidades CURSOR FOR
Select ctb_c_cat_contable,
asa_c_area
from ctb_c_cat_contable
where asa_c_aeropuerto = :ls_asa_c_aeropuerto and
concen_cct_si <> 1;

OPEN contabilidades;

lb_termino = false
li_numero_de_catalogos = 0

DO WHILE SQLCA.sqlcode = 0 and not(lb_termino)
FETCH contabilidades INTO :li_catalogo,:ls_area;
if SQLCA.sqlcode <> 0 then
if SQLCA.sqlcode = 100 then
lb_termino = true
else

```

```
MessageBox("ERROR", "Error al determinar los catálogos de años anteriores ~r" + &
```

```
    "Error: " + String(SQLCA.sqlcode) + "~r" + &
    "Errordb: " + String(SQLCA.sqldbcode) + "~r" + &
    SQLCA.sqlerrtext)
    lb_realizo = false
    lb_termino = true
```

```
end if
```

```
else
```

```
    li_numero_de_catalogos ++
    li_catalogos_anteriores[li_numero_de_catalogos] = li_catalogo
    ls_areas_anteriores[li_numero_de_catalogos] = ls_area
```

```
end if
```

```
loop
```

```
CLOSE contabilidades;
```

```
ai_num_cat = li_numero_de_catalogos
aa_catalogos[] = li_catalogos_anteriores[]
aa_areas[] = ls_areas_anteriores[]
```

```
RETURN lb_realizo
```

```
end function
```

```
//Función para generar la póliza de concelación de un mes de contable de un
aeropuerto
```

```
integer ai_mes_pol_si);Integer li_num_pol_i,li_dia,li_anio
Datetime ldt_fecha_hoy,ldt_fecha_contabilizacion
Boolean lb_continuar,lb_cargos,lb_abonos,lb_inserta_registro
integer li_ctb_c_cat_contable,li_contador,li_num_movimientos,li_asa_c_area_num
String ls_asa_c_area,ls_refer_cta_str
double ld_cifra_control
long ll_ctb_c_cuenta
decimal {2} ld_debe_mov_n, ld_haber_mov_n,ln_tot_debe_pol_n,ln_tot_haber_pol_n
String ls_cifras_de_control
```

```
decimal {2} ld_ene_d_cta_n, &
    ld_ene_h_cta_n, &
    ld_feb_d_cta_n, &
    ld_feb_h_cta_n, &
    ld_mar_d_cta_n, &
    ld_mar_h_cta_n, &
```

```

ld_abr_d_cta_n, &
ld_abr_h_cta_n, &
ld_may_d_cta_n, &
ld_may_h_cta_n, &
ld_jun_d_cta_n, &
ld_jun_h_cta_n, &
ld_jul_d_cta_n, &
ld_jul_h_cta_n, &
ld_ago_d_cta_n, &
ld_ago_h_cta_n, &
ld_sep_d_cta_n, &
ld_sep_h_cta_n, &
ld_oct_d_cta_n, &
ld_oct_h_cta_n, &
ld_nov_d_cta_n, &
ld_nov_h_cta_n, &
ld_dic_d_cta_n, &
ld_dic_h_cta_n

```

```

SELECT ctb_c_cat_contable.asa_c_area,
       ctb_c_cat_contable.asa_c_area_num
INTO :ls_asa_c_area,
     :li_asa_c_area_num
FROM ctb_c_cat_contable
WHERE ctb_c_cat_contable = :ai_ctb_c_cat_contable;

```

```

If SQLCA.sqlcode <> 0 then
  MessageBox("ERROR","Error al determinar las áreas del catálogo~r" + &
             "Error: " + String(SQLCA.sqlcode) + "~r" + &
             "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
             SQLCA.sqlerrtext)
Return false
end if

```

```

li_num_pol_i = 5000
ldt_fecha_hoy = ctb_f_fecha_hora_servidor ()

```

```

//armamos la fecha de contabilización
li_dia = Day(Date(ldt_fecha_hoy))
li_anio = Year(Date(ldt_fecha_hoy))

```

```

ldt_fecha_contabilizacion = Datetime(Date ( li_anio, ai_mes_pol_si,li_dia))

```

```

INSERT INTO dbo.ctb_m_poliza
( ctb_c_cat_contable,

```

```
tipo_pol_str,  
mes_pol_si,  
num_pol_i,  
area_gen_pol_str,  
f_reg_pol_dt,  
f_apl_pol_dt,  
f_cont_pol_dt,  
sys_usuarios,  
u_rev_pol_str,  
u_aut_pol_str,  
concepto_pol_str,  
cif_con_pol_str,  
num_mov_pol_si,  
tot_debe_pol_n,  
tot_haber_pol_n,  
rev_pol_si,  
aut_pol_si,  
conta_pol_si,  
auto_pol_si,  
asa_c_aeroorig,  
asa_c_aerodest )  
VALUES ( :ai_ctb_c_cat_contable,  
"D",  
:ai_mes_pol_si,  
:li_num_pol_i,  
:ls_asa_c_area,  
:ldt_fecha_hoy,  
null,  
:ldt_fecha_contabilizacion,  
"xrojel",  
"xrojel",  
null,  
"Poliza de cancelación del mes",  
null,  
null,  
null,  
null,  
1,  
0,  
0,  
0,  
null,  
null ) ;
```

```
If SQLCA.sqlcode <> 0 then
```

```

    MessageBox("ERROR", "Error al insertar el encabezado de la póliza de
cancelación~r" + &
    "Error: " + String(SQLCA.sqlcode) + "~r" + &
    "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
    SQLCA.sqlerrtext)

```

```

    Return false
end if

```

```

//Una vez que el registro de la poliza está insertado procedemos a insertar los
//movimientos, declaramos un cursor

```

```

//se barren todas las cuentas dos veces la primera para detectar los abonos y la
segunda
//para detectar los abonos

```

```

li_num_movimientos = 1
ln_tot_debe_pol_n = 0
ln_tot_haber_pol_n = 0
ld_cifra_control = 0
for li_contador = 1 to 2
    if li_contador = 1 then
        lb_cargos = true
        lb_abonos = false
    else
        lb_cargos = false
        lb_abonos = true
    end if

```

```

DECLARE cuentas_con_movimiento CURSOR FOR

```

```

SELECT dbo.ctb_m_cuenta.ctb_c_cat_contable,
    dbo.ctb_m_cuenta.asa_c_area,
    dbo.ctb_m_cuenta.ctb_c_cuenta,
    dbo.ctb_m_cuenta.ene_d_cta_n,
    dbo.ctb_m_cuenta.ene_h_cta_n,
    dbo.ctb_m_cuenta.feb_d_cta_n,
    dbo.ctb_m_cuenta.feb_h_cta_n,
    dbo.ctb_m_cuenta.mar_d_cta_n,
    dbo.ctb_m_cuenta.mar_h_cta_n,
    dbo.ctb_m_cuenta.abr_d_cta_n,
    dbo.ctb_m_cuenta.abr_h_cta_n,
    dbo.ctb_m_cuenta.may_d_cta_n,
    dbo.ctb_m_cuenta.may_h_cta_n,
    dbo.ctb_m_cuenta.jun_d_cta_n,
    dbo.ctb_m_cuenta.jun_h_cta_n,
    dbo.ctb_m_cuenta.jul_d_cta_n,
    dbo.ctb_m_cuenta.jul_h_cta_n,

```

```

        dbo.ctb_m_cuenta.ago_d_cta_n,
        dbo.ctb_m_cuenta.ago_h_cta_n,
        dbo.ctb_m_cuenta.sep_d_cta_n,
        dbo.ctb_m_cuenta.sep_h_cta_n,
        dbo.ctb_m_cuenta.oct_d_cta_n,
        dbo.ctb_m_cuenta.oct_h_cta_n,
        dbo.ctb_m_cuenta.nov_d_cta_n,
        dbo.ctb_m_cuenta.nov_h_cta_n,
        dbo.ctb_m_cuenta.dic_d_cta_n,
        dbo.ctb_m_cuenta.dic_h_cta_n,
        dbo.ctb_c_cuenta.cuenta_cat_str
FROM dbo.ctb_m_cuenta,dbo.ctb_c_cuenta
WHERE ( dbo.ctb_m_cuenta.ctb_c_cat_contable =
dbo.ctb_c_cuenta.ctb_c_cat_contable) and
        ( dbo.ctb_m_cuenta.asa_c_area      = dbo.ctb_c_cuenta.asa_c_area)
and
        ( dbo.ctb_m_cuenta.ctb_c_cuenta    = dbo.ctb_c_cuenta.ctb_c_cuenta)
and
        ( dbo.ctb_m_cuenta.ctb_c_cat_contable = :ai_ctb_c_cat_contable ) AND
        ( dbo.ctb_m_cuenta.ctb_c_cuenta >= 30000 ) ;

If SQLCA.sqlcode <> 0 then
    MessageBox("ERROR","Error al declarar el cursor~r" + &
        "Error: " + String(SQLCA.sqlcode) + "~r" + &
        "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
        SQLCA.sqlerrtext)
    Return false
End if

OPEN cuentas_con_movimiento;
If SQLCA.sqlcode <> 0 then
    MessageBox("ERROR","Error al declarar el cursor~r" + &
        "Error: " + String(SQLCA.sqlcode) + "~r" + &
        "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
        SQLCA.sqlerrtext)
    Return False
End if

lb_continuar = true
do While lb_continuar
    Fetch cuentas_con_movimiento INTO : li_ctb_c_cat_contable, &
        : ls_asa_c_area, &
        : ll_ctb_c_cuenta, &
        : ld_ene_d_cta_n, &
        : ld_ene_h_cta_n, &
        : ld_feb_d_cta_n, &
        : ld_feb_h_cta_n, &

```

```

:ld_mar_d_cta_n, &
:ld_mar_h_cta_n, &
:ld_abr_d_cta_n, &
:ld_abr_h_cta_n, &
:ld_may_d_cta_n, &
:ld_may_h_cta_n, &
:ld_jun_d_cta_n, &
:ld_jun_h_cta_n, &
:ld_jul_d_cta_n, &
:ld_jul_h_cta_n, &
:ld_ago_d_cta_n, &
:ld_ago_h_cta_n, &
:ld_sep_d_cta_n, &
:ld_sep_h_cta_n, &
:ld_oct_d_cta_n, &
:ld_oct_h_cta_n, &
:ld_nov_d_cta_n, &
:ld_nov_h_cta_n, &
:ld_dic_d_cta_n, &
:ld_dic_h_cta_n, &
:ls_refer_cta_str;

If SQLCA.sqlcode <> 0 then
  If SQLCA.sqlcode <> 100 then
    MessageBox("ERROR","Error en el fetch de cursor~r" + &
      "Error: " + String(SQLCA.sqlcode) + "~r" + &
      "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
      SQLCA.sqlerrtext)

    Return False
  else
    //es el último registro
    lb_continuar = false
  end if
else
  //Trajo un registro que hay que insertar en la tabla de movimientos

  CHOOSE CASE ai_mes_pol_si
    CASE 1
      if lb_cargos then
        ld_debe_mov_n = ld_ene_d_cta_n * -1
        ld_haber_mov_n = 0
        if ld_debe_mov_n <> 0 then
          lb_inserta_registro = true
        else
          lb_inserta_registro = false
        end if
      end if

```

```

else
    ld_debe_mov_n = 0
    ld_haber_mov_n = ld_ene_h_cta_n * -1
    if ld_haber_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
CASE 2
if lb_cargos then
ld_debe_mov_n = ld_feb_d_cta_n * -1
    ld_haber_mov_n = 0
    if ld_debe_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
else
    ld_debe_mov_n = 0
    ld_haber_mov_n = ld_feb_h_cta_n * -1
    if ld_haber_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
CASE 3
if lb_cargos then
    ld_debe_mov_n = ld_mar_d_cta_n * -1
    ld_haber_mov_n = 0
    if ld_debe_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
else
    ld_debe_mov_n = 0
    ld_haber_mov_n = ld_mar_h_cta_n * -1
    if ld_haber_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
CASE 4
if lb_cargos then

```

```

        ld_debe_mov_n = ld_abr_d_cta_n * -1
        ld_haber_mov_n = 0
        if ld_debe_mov_n <> 0 then
            lb_inserta_registro = true
        else
            lb_inserta_registro = false
        end if
    else
        ld_debe_mov_n = 0
        ld_haber_mov_n = ld_abr_h_cta_n * -1
        if ld_haber_mov_n <> 0 then
            lb_inserta_registro = true
        else
            lb_inserta_registro = false
        end if
    end if
CASE 5
    if lb_cargos then
        ld_debe_mov_n = ld_may_d_cta_n * -1
        ld_haber_mov_n = 0
        if ld_debe_mov_n <> 0 then
            lb_inserta_registro = true
        else
            lb_inserta_registro = false
        end if
    else
        ld_debe_mov_n = 0
        ld_haber_mov_n = ld_may_h_cta_n * -1
        if ld_haber_mov_n <> 0 then
            lb_inserta_registro = true
        else
            lb_inserta_registro = false
        end if
    end if
CASE 6
    if lb_cargos then
        ld_debe_mov_n = ld_jun_d_cta_n * -1
        ld_haber_mov_n = 0
        if ld_debe_mov_n <> 0 then
            lb_inserta_registro = true
        else
            lb_inserta_registro = false
        end if
    else
        ld_debe_mov_n = 0
        ld_haber_mov_n = ld_jun_h_cta_n * -1
        if ld_haber_mov_n <> 0 then

```

```

        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
CASE 7
if lb_cargos then
    ld_debe_mov_n = ld_jul_d_cta_n * -1
    ld_haber_mov_n = 0
    if ld_debe_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
else
    ld_debe_mov_n = 0
    ld_haber_mov_n = ld_jul_h_cta_n * -1
    if ld_haber_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
CASE 8
if lb_cargos then
    ld_debe_mov_n = ld_ago_d_cta_n * -1
    ld_haber_mov_n = 0
    if ld_debe_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
else
    ld_debe_mov_n = 0
    ld_haber_mov_n = ld_ago_h_cta_n * -1
    if ld_haber_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
CASE 9
if lb_cargos then
    ld_debe_mov_n = ld_sep_d_cta_n * -1
    ld_haber_mov_n = 0
    if ld_debe_mov_n <> 0 then
        lb_inserta_registro = true
    end if
end if

```

```
        else
            lb_inserta_registro = false
        end if
    else
        ld_debe_mov_n = 0
        ld_haber_mov_n = ld_sep_h_cta_n * -1
        if ld_haber_mov_n <> 0 then
            lb_inserta_registro = true
        else
            lb_inserta_registro = false
        end if
    end if
CASE 10
if lb_cargos then
    ld_debe_mov_n = ld_oct_d_cta_n * -1
    ld_haber_mov_n = 0
    if ld_debe_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
else
    ld_debe_mov_n = 0
    ld_haber_mov_n = ld_oct_h_cta_n * -1
    if ld_haber_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
CASE 11
if lb_cargos then
    ld_debe_mov_n = ld_nov_d_cta_n * -1
    ld_haber_mov_n = 0
    if ld_debe_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
else
    ld_debe_mov_n = 0
    ld_haber_mov_n = ld_nov_h_cta_n * -1
    if ld_haber_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
```

```

end if
CASE 12
if lb_cargos then
    ld_debe_mov_n = ld_dic_d_cta_n * -1
    ld_haber_mov_n = 0
    if ld_debe_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
else
    ld_debe_mov_n = 0
    ld_haber_mov_n = ld_dic_h_cta_n * -1
    if ld_haber_mov_n <> 0 then
        lb_inserta_registro = true
    else
        lb_inserta_registro = false
    end if
end if
END CHOOSE

if lb_inserta_registro then
    INSERT INTO dbo.ctb_m_movimiento
    ( ctb_c_cat_contable,
    tipo_pol_str,
    mes_pol_si,
    num_pol_i,
    sec_mov_si,
    cat_mov_si,
    asa_c_area_num,
    area_cta_pol_str,
    ctb_c_cuenta,
    refer_cta_str,
    concepto_mov_str,
    chq_ref_mov_str,
    debe_mov_n,
    haber_mov_n,
    asa_c_aerorig,
    asa_c_aerodest )
VALUES ( :ai_ctb_c_cat_contable,
"D",
:ai_mes_pol_si,
5000,
:li_num_movimientos,
:ai_ctb_c_cat_contable,
:li_asa_c_area_num,
:ls_asa_c_area,

```

```

        :ll_ctb_c_cuenta,
        :ls_refer_cta_str,
        "Cancelacion mes",
        null,
        :ld_debe_mov_n,
        :ld_haber_mov_n,
        null,
        null ) ;
    If SQLCA.sqlcode <> 0 then
        MessageBox("ERROR","Error al insertar movimientor~r" + &
            "Error: " + String(SQLCA.sqlcode) + "~r" + &
            "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
            SQLCA.sqlerrtext)
            Return False
        else
            li_num_movimientos ++
            Id_cifra_control = Id_cifra_control + Double(ls_refer_cta_str)
            ln_tot_debe_pol_n = ln_tot_debe_pol_n +
            Id_debe_mov_n
            ln_tot_haber_pol_n = ln_tot_haber_pol_n + Id_haber_mov_n
        end if
    end if
end if
loop

    close cuentas_con_movimiento;
    MessageBox("AVISO", "cargos " + String(ln_tot_debe_pol_n) + " Abonos " +
        String(ln_tot_haber_pol_n))
next

//una vez que se tienen los movimientos de la póliza se procede a actualizar el
encabezado

ls_cifras_de_control = String(Id_cifra_control)

UPDATE ctb_m_poliza
SET tot_debe_pol_n = :ln_tot_debe_pol_n,
    tot_haber_pol_n = :ln_tot_haber_pol_n,
    cif_con_pol_str = :ls_cifras_de_control
WHERE ( ctb_m_poliza.ctb_c_cat_contable = :ai_ctb_c_cat_contable ) AND
    ( ctb_m_poliza.tipo_pol_str = "D" ) AND
    ( ctb_m_poliza.mes_pol_si = :ai_mes_pol_si ) AND
    ( ctb_m_poliza.num_pol_i = 5000 ) ;

if SQLCA.sqlcode <> 0 then
    MessageBox("ERROR","Error al Actualizar el encabezado de una póliza ~r" + &

```

```
"Error: " + String(SQLCA.sqlcode) + "~r" + &  
"ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &  
SQLCa.sqlerrtext)
```

```
Return false  
end if
```

```
Return true
```

```
end function
```

Código del objeto dropdown que obtiene las contabilidades dependientes de una determinada contabilidad.

```
//Objeto dropdown que obtiene los catálogos dependientes de una contabilidad
```

```
event constructor;  
Integer li_contador_drop,  
li_picture,li_contador,li_cat_a_observar,li_item_a_selec_en_ddplb  
String ls_mascara,ls_area_original,ls_descrip,ls_area_pertenece,ls_areas_hijas  
Boolean lb_primer_catalogo,lb_puede_ver_catalogo_actual  
Integer  
li_catalogo_usuario,li_num_catalogos_del_area,li_catalogo_padre,li_concentra  
Integer li_cat_normativo,li_cat_hijos[]  
String ls_aeropuerto,ls_aeropuerto_padre,ls_area_padre  
Integer  
li_num_cat,li_cat_propio,li_anio_actual,li_concentrador_aeropuerto,li_anio_catalogo  
Any li_catalogos_hijos[]
```

```
//limpiamos el arreglo que nos indica cuál es la relación entre el dropdown que  
muestra  
//los catálogos a los que se puede acceder y el número de catálogo de la Base de  
Datos  
for li_contador = 1 to li_max_catalogos  
  ii_control_catalogos[li_contador] = 0  
next
```

```
//Se obtiene la información del área del usuario que está utilizando el sistema
```

```
ls_area_original = gnv_app.ls_area  
is_area_pertenece = ls_area_original  
li_anio_actual = ctb_f_anio_actual()
```

```
//para los usuarios de las áreas que no tiene catálogo propio se selecciona cual es su
```

```

//área padre que contiene un catálogo

li_cat_propio = ctb_f_obten_catálogo_propio(ls_area_original,li_anio_actual)

select asa_c_area
  into :ls_area_padre
  from ctb_c_cat_contable
  where ctb_c_cat_contable = :li_cat_propio and
        anio_cct_si       = :li_anio_actual ;
  if SQLCA.sqlcode <> 0 and SQLCA.sqlcode <> 100 then
    Messagebox("ERROR","No puedo determinar el área padre ~r" + &
      "Error: " + String(SQLCA.sqlcode) + "~r" + &
      "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
      SQLCA.sqlerrtext)
  end if

//Se determina el catálogo concentrador del área que puede ver el usuario
Select ctb_c_cat_contable
  into :li_catálogo_padre
  from ctb_c_cat_contable
  where asa_c_area   = :ls_area_padre and
        concen_cct_si = 1 and
        anio_cct_si   = :li_anio_actual;

if SQLCA.sqlcode <> 0 then
  if SQLCA.sqlcode = 100 then
    //no hay catálogo concentrador
    Select ctb_c_cat_contable
      into :li_catálogo_padre
      from ctb_c_cat_contable
      where asa_c_area   = :ls_area_padre and
            concen_cct_si = 0 and
            anio_cct_si   = :li_anio_actual;
    if SQLCA.sqlcode <> 0 then
      Messagebox("ERROR","No puedo determinar los catálogos del área del
usuario~r" + &
        "Error: " + String(SQLCA.sqlcode) + "~r" + &
        "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
        SQLCA.sqlerrtext)
      end if
    else
      Messagebox("ERROR","No puedo determinar los catálogos del área del
usuario~r" + &
        "Error: " + String(SQLCA.sqlcode) + "~r" + &
        "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
        SQLCA.sqlerrtext)
    end if
  end if

```

```

end if

//Del catálogo padre se determina el aeropuerto al que pertenece para poder ver
//posteriormente sólo los catálogos del aeropuerto
Select asa_c_aeropuerto
  into :ls_aeropuerto_padre
  from ctb_c_cat_contable
  where ctb_c_cat_contable = :li_catalogo_padre;

if SQLCA.sqlcode <> 0 then
  MessageBox("ERROR","No puedo determinar el aeropuerto en donde me
  encuentro~r" + &
    "Error: " + String(SQLCA.sqlcode) + "~r" + &
    "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
    SQLCA.sqlerrtext)
end if

//Se obtienen los catálogos dependientes que puede observar el usuario
//primero determinamos todos los catálogos hijos del catálogo en cuestión
ctb_f_determina_cat_dependientes(li_catalogo_padre,li_num_cat,li_catalogos_hijos[]
)
li_cat_hijos[] = li_catalogos_hijos[]

is_area_pertenece = ls_area_pertenece
li_cat_a_observar = 0
li_contador_drop = 0
lb_primer_catalogo = false
li_item_a_selec_en_ddplb = 0
ddplb_areas_hijas.Reset()

for li_contador = 1 to li_num_cat
  select
  asa_c_aeropuerto,descrip_cct_str,concen_cct_si,norma_cct_si,concenaecct_si,anio
  _cct_si
  into :ls_aeropuerto,:ls_descrip,:li_concentra,:li_cat_normativo,
  :li_concentrador_aeropuerto,:li_anio_catalogo
  from ctb_c_cat_contable
  where ctb_c_cat_contable = :li_cat_hijos[li_contador];
  if SQLCA.sqlcode <> 0 then
    MessageBox("ERROR","No puedo extraer los datos de un catálogo~r" + &
      "Error: " + String(SQLCA.sqlcode) + "~r" + &
      "ErrorDB: " +
String(SQLCA.sqldbcode) + "~r" + &
      SQLCA.sqlerrtext)
  end if
  if li_cat_hijos[li_contador] = li_catalogo_padre then

```

```

        lb_primer_catalogo = true
    end if

    lb_puede_ver_catalogo_actual = true
    if lb_puede_ver_catalogo_actual then
        ls_areas_hijas = ls_descrip
        li_contador_drop ++
        if li_cat_normativo = 1 then
            li_picture = 1
        elseif li_concentra = 1 then
            li_picture = 2
        else
            li_picture = 3
        end if
        ddplb_areas_hijas.AddItem (ls_areas_hijas,li_picture)
        ii_control_catalogos[li_contador_drop] = li_cat_hijos[li_contador]
        if lb_primer_catalogo then
            li_item_a_selec_en_ddplb = li_contador_drop
            li_cat_a_observar = li_cat_hijos[li_contador]
        end if
    end if
    lb_primer_catalogo = false
next

if li_cat_a_observar = 0 then
    li_cat_a_observar = ii_control_catalogos[1]
    li_item_a_selec_en_ddplb = 1
end if

ddplb_areas_hijas.SelectItem(li_item_a_selec_en_ddplb)
grv_app.is_area = ls_area_original

ii_catalogo_a_observar = li_cat_a_observar
end event

event modified;
call super::modified;ii_catalogo_a_observar =
ddplb_areas_hijas.FindItem(ddplb_areas_hijas.text,0)
ii_catalogo_a_observar = ii_control_catalogos[ii_catalogo_a_observar]

Parent.Event ue_cuando_cambie(ii_catalogo_a_observar)
end event

```

Codigo de la ventana ctb\_w\_cuentas para consulta de los catálogos contables

```

event open;
call super::open;Integer li_contador_drop,
li_picture,li_contador,li_cat_a_observar,li_item_a_selec_en_ddplb
String ls_mascara,ls_area_original,ls_descrip,ls_area_pertenece,ls_areas_hijas
Boolean lb_primer_catalogo,lb_puede_ver_catalogo_actual
Integer
li_catalogo_usuario,li_num_catalogos_del_area,li_catalogo_padre,li_concentra
Integer li_cat_normativo,li_cat_hijos[]
String ls_aeropuerto,ls_aeropuerto_padre,ls_area_padre
Integer li_num_cat,li_cat_propio,li_anio_actual
Any li_catalogos_hijos[]

//limpiamos el arreglo que nos indica cuál es la relación entre el dropdown que
muestra
//los catálogos a los que se puede acceder y el número de catálogo de la Base de
Datos
for li_contador = 1 to li_max_catalogos
    li_control_catalogos[li_contador] = 0
next

//Se obtiene la información del área del usuario que está utilizando el sistema
dw_datos_area_usuario.of_SetTransObject(SQLCA)
dw_datos_area_usuario.Reset()
dw_datos_area_usuario.Retrieve(gnv_app.is_area)

ls_area_original = gnv_app.is_area
is_area_pertenece = ls_area_original
li_anio_actual = ctb_f_anio_actual()

//para los usuarios de las áreas que no tiene catálogo propio se selecciona cual es su
//área padre que contiene un catálogo

li_cat_propio = ctb_f_obten_catalogo_propio(ls_area_original,li_anio_actual)

select asa_c_area
into :ls_area_padre
from ctb_c_cat_contable
where ctb_c_cat_contable = : li_cat_propio and
      anio_cct_si = : li_anio_actual ;
if SQLCA.sqlcode <> 0 and SQLCA.sqlcode <> 100 then
    MessageBox("ERROR", "No puedo determinar el área padre ~r" + &
                "Error: " + String(SQLCA.sqlcode) + "~r" + &
                "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
                SQLCA.sqlerrtext)
end if

//Se determina el catálogo concentrador del área que puede ver el usuario

```

```

Select ctb_c_cat_contable
into :li_catalogo_padre
from ctb_c_cat_contable
  where asa_c_area = :ls_area_padre and
  concen_cct_si = 1 and
  anio_cct_si = :li_anio_actual;

if SQLCA.sqlcode <> 0 then
  if SQLCA.sqlcode = 100 then
    //no hay catálogo concentrador
    Select ctb_c_cat_contable
    into :li_catalogo_padre
    from ctb_c_cat_contable
      where asa_c_area = :ls_area_padre and
      concen_cct_si = 0 and
      anio_cct_si = :li_anio_actual;
    if SQLCA.sqlcode <> 0 then
      MessageBox("ERROR", "No puedo determinar los catálogos del área del
usuario~r" + &
        "Error: " + String(SQLCA.sqlcode) + "~r" + &
        "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
        SQLCA.sqlerrtext)
    end if
  else
    MessageBox("ERROR", "No puedo determinar los catálogos del área del
usuario~r" + &
      "Error: " + String(SQLCA.sqlcode) + "~r" + &
      "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
      SQLCA.sqlerrtext)
  end if
end if

//Del catálogo padre se determina el aeropuerto al que pertenece para poder ver
//posteriormente sólo los catálogos del aeropuerto
Select asa_c_aeropuerto
into :ls_aeropuerto_padre
from ctb_c_cat_contable
  where ctb_c_cat_contable = :li_catalogo_padre;

if SQLCA.sqlcode <> 0 then
  MessageBox("ERROR", "No puedo determinar el aeropuerto en donde me
encuentro~r" + &
    "Error: " + String(SQLCA.sqlcode) + "~r" + &
    "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
    SQLCA.sqlerrtext)
end if

```

```

//Se obtienen los catálogos dependientes que puede observar el usuario
//primero determinamos todos los catálogos hijos del catálogo en cuestión
ctb_f_determina_cat_dependientes(li_catalogo_padre,li_num_cat,li_catalogos_hijos[]
)
li_cat_hijos[] = li_catalogos_hijos[]

is_area_pertenece = ls_area_pertenece
li_cat_a_observar = 0
li_contador_drop = 0
lb_primer_catalogo = false
li_item_a_selec_en_ddplb = 0
ddplb_areas_hijas.Reset()

for li_contador = 1 to li_num_cat
    select asa_c_aeropuerto,descrip_cct_str,concen_cct_si,norma_cct_si
    into :ls_aeropuerto,:ls_descrip,:li_concentra,:li_cat_normativo
    from ctb_c_cat_contable
    where ctb_c_cat_contable = :li_cat_hijos[li_contador];
    if SQLCA.sqlcode <> 0 then
        MessageBox("ERROR","No puedo extraer los datos de un catálogo~r" + &
            "Error: " + String(SQLCA.sqlcode) + "~r" + &
            "ErrorDB: " + String(SQLCA.sqldbcode) + "~r" + &
            SQLCA.sqlerrtext)
    end if
    if li_cat_hijos[li_contador] = li_catalogo_padre then
        lb_primer_catalogo = true
    end if

    if ls_aeropuerto = ls_aeropuerto_padre then
        lb_puede_ver_catalogo_actual = true
    else
        lb_puede_ver_catalogo_actual = false
    end if

    if li_cat_normativo = 1 then
        lb_puede_ver_catalogo_actual = true
    end if

    if lb_puede_ver_catalogo_actual then
        ls_areas_hijas = ls_descrip
        li_contador_drop ++
        if li_cat_normativo = 1 then
            li_picture = 1
        elseif li_concentra = 1 then
            li_picture = 3
        else

```

```

        li_picture = 2
    end if

    ddplb_areas_hijas.AddItem (ls_areas_hijas,li_picture)
    li_control_catalogos[li_contador_drop] = li_cat_hijos[li_contador]
    if lb_primer_catalogo then
        li_item_a_selec_en_ddplb = li_contador_drop
        li_cat_a_observar = li_cat_hijos[li_contador]
    end if
end if
lb_primer_catalogo = false
next

if li_cat_a_observar = 0 then
    li_cat_a_observar = li_control_catalogos[1]
    li_item_a_selec_en_ddplb = 1
end if

ddplb_areas_hijas.SelectItem(li_item_a_selec_en_ddplb)
gmv_app.is_area = ls_area_original

of_SetResize( True )

ls_mascara = wf_mascara_cuenta(li_cat_a_observar)
tv_ctas.of_SetListView( tv_ctas )           // Indica que se usara un listView (Estilo
explorer)
tv_ctas.event
ue_Muestra_Catalogo(li_cat_a_observar,is_area_pertenece,ls_mascara )
tv_ctas.of_SetMenu( m_cuentas )           // Indica el menú que utilizara

of_SetResize( True )

ls_mascara = wf_mascara_cuenta(uo_catalogos_contables.li_catalogo_a_observar)
tv_ctas.of_SetListView( tv_ctas )           // Indica que se usara un listView (Estilo
explorer)
tv_ctas.event
ue_Muestra_Catalogo(uo_catalogos_contables.li_catalogo_a_observar,is_area_perte
nece,ls_mascara )
tv_ctas.of_SetMenu( m_cuentas )           // Indica el menú que utilizara

end event

```

```
event close;
call super::close;ddplb_areas_hijas.Reset()
end event

event ue_muestra_catalogo;
call super::ue_muestra_catalogo;tv_ctas.of_DeleteTreeView()

lv_ctas.DeleteItems()
tv_ctas.of_Muestra_Catalogo( ai_catalogo,as_area_pertenece,as_mascara )
end event

event ue_escontable;call super::ue_escontable;isDWDependientes =
"ctb_dw_cuentas_contables_dep"
isDWPropiedades = "ctb_dw_datos_cuenta"

isKeyDependiente[1] = "cat_padre_i"
isKeyDependiente[2] = "area_padre_str"
isKeyDependiente[3] = "cta_padre_i"

isNivelCat      = "nivel_cat_si"
isIcono         = "sys_iconos"

is_consulta = "NO"
end event

event ue_cuando_cambie;
call super::ue_cuando_cambie;Integer li_catalogo_a_observar
String ls_mascara

li_catalogo_a_observar = ai_catalogo

ls_mascara = wf_mascara_cuenta(li_catalogo_a_observar)

tv_ctas.event
ue_muestra_catalogo(li_catalogo_a_observar,is_area_pertenece,ls_mascara)
end event

event ue_cuando_cambie;call super::ue_cuando_cambie;Integer
li_catalogo_a_observar
String ls_mascara

li_catalogo_a_observar = ai_catalogo

ls_mascara = wf_mascara_cuenta(li_catalogo_a_observar)
```

```

tv_ctas.event
ue_muestra_catalogo(li_catalogo_a_observar,is_area_pertenece,is_mascara)
end event

```

Código del objeto `uo_cuentas` utilizado para la consulta y modificación de las cuentas de los catálogos

```

event ue_escontable ( )
event ue_espresupuestal ( )
event ue_nueva ( long alhandle )
event ue_modificar ( long alhandle )
event ue_eliminar ( long alhandle )
event ue_firstretrieve ( long row, datastore dsdep )
event ue_imprimir ( integer ai_catalogo )
end type
global uo_cuentas uo_cuentas

```

type variables

Integer iiTipo

u\_lv ilv

Boolean ibLv = False

// DataWindows de Apoyo

String isDWDependientes

String isDWPropiedades

Integer ii\_cat\_a\_observar

String is\_area\_pertenece

String is\_mascara

DataStore idsDep

String isColumnKeys[]

Boolean ibAlwaysRefresh = False

Menu imPopup

String isPictures[]

Boolean ibEsAlta

Boolean ibEsBaja

Boolean ibEsModificacion

// Para altas y modificaciones

Long ilHandle

```
TreeViewItem itvi,itvi_padre
integer ii_catalogos[]
integer ii_num_cat
String ii_areas[]
String isKeyDependiente[]
String isNivelCat
String isIcono

Integer iilIndexNew

//para imprimir los reportes contables
String is_balanza
String is_consulta

String is_catalogo

String is_libromayor
String is_librodiario
end variables

forward prototypes
public function integer of_tipodecatalogo (integer aitipo)
public subroutine of_refreshlv (integer alhandle)
public subroutine of_setlistview (listview alv)
public subroutine of_setmenu (menu amMenu)
public function integer of_addpicture (string aspicturename)
public function any of_getitemany (long al_row, string as_column)
public function boolean of_getkeys (long alrow, ref any aaarray[])
public subroutine of_setlistviewcolumns ()
public subroutine of_deletelevel (integer alhandle)
public subroutine of_addrow (long alhandle, long alrow, boolean abreplaceitem)
public subroutine of_refreshdw (long alhandle)
public subroutine of_deletetreeview ()
public function boolean of_puede_dar_alta (string as_usuario, integer ai_id_catalogo)
public subroutine of_muestra_catalogo (integer ai_id_catalogo, string
as_area_pertenece, string as_mascara)
public subroutine of_refresh (long alhandle, boolean ab_refresca)
public subroutine of_refresca_treeview ()
public subroutine of_catalogos_a_actualizar (integer ai_catalogo)
public function boolean of_procede_cambio (string as_usuario, integer
ai_id_catalogo, integer ai_nivel, string as_accion)
public function boolean of_procede_baja (string as_usuario, integer ai_id_catalogo,
integer ai_nivel)
public function boolean of_procede_alta (string as_usuario, integer ai_id_catalogo,
integer ai_nivel)
end prototypes
```

```

event ue_espresupuestal;
isDWDependientes      = ""
isDWPropiedades       = ""

iskeyDependiente[1]  = "area_padre_str"
iskeyDependiente[2]  = "cta_padre_i"

isNivelCat            = "nivel_cat_si"
isIcono               = "sys_iconos"

end event

event ue_nueva;any la_Datos_padre[3],la_llave_padre[3]
integer li_nivel,li_catalogo_padre,li_id_cuenta_padre
String ls_area_padre
treeviewitem Itvi_padre
ctb_s_accion_sobre_catalogo ls_accion_sobre_catalogo

If IsNull( alHandle ) Then
    alhandle = FindItem( CurrentTreetItem!, 0 )
    If alHandle = -1 Then
        MessageBox( "Error", "No puedo determinar la cuenta padre" )
        Return
    End If
End If

//Obtengo los datos del item Padre
GetItem(alhandle,Itvi_padre)
itvi_padre = Itvi_padre
ibesmodificacion = false
ibEsAlta = true
if of_procede_alta(gnv_app.of_getuserid(),ii_cat_a_observar,itvi_padre.level) then
    ibEsAlta = True
    li_nivel = itvi.level
    itvi.Label = "( Cuenta nueva )"
    itvi.PictureIndex = iiIndexNew
    itvi.SelectedPictureIndex = iiIndexNew
    ExpandItem( alHandle )
    iiHandle = InsertItemLast( alHandle, itvi )
    SelectItem( iiHandle )
    OpenWithParm( ctb_w_datos_cuenta, This )
end if

end event

```

```
event ue_modificar;any
la_datos_padre[3],la_llave_padre[3],la_datos_cuenta[3],la_llave_cuenta[]
integer li_nivel,li_catalogo_padre,li_catalogo
String ls_area_padre,ls_nueva_descripcion,ls_area,ls_cuenta
treeviewitem ltv_i_padre
ctb_s_accion_sobre_catalogo ls_accion_sobre_catalogo
double li_id_cuenta_padre,ld_cuenta

ibesmodificacion = true
//Obtengo los datos del item Padre
GetItem(alhandle,ltv_i_padre)
itvi_padre = ltv_i_padre

If IsNull( alHandle ) Then
    alhandle = FindItem( CurrentTreeItem!, 0 )
    If alHandle = -1 Then
        MessageBox( "Error", "No puedo determinar el item" )
        Return
    End If
End If

Event SelectionChanged( alHandle, alHandle )

ibEsAlta = False

ExpandItem( alHandle )

GetItem( alHandle, itvi )
ilHandle = alHandle

if
of_procede_cambio(gnv_app.of_getuserid(),li_cat_a_observar,itvi_padre.level,"alta")
then
    ibesmodificacion = true
    OpenWithParm( ctb_w_datos_cuenta, This )
end if

//Las líneas siguientes son para efecto de presentación y se hace para
//que se actualicen los cambios en la pantalla que se está viendo en este momento

la_datos_cuenta = itvi.Data
la_llave_cuenta = la_datos_cuenta[3]

li_catalogo = la_llave_cuenta[1]
ls_area = la_llave_cuenta[2]
ld_cuenta = la_llave_cuenta[3]
```

## **CONCLUSIONES**

Con el desarrollo el presente trabajo llegamos a varias conclusiones importantes que sin duda alguna registrarán muchas de las acciones que en un futuro podamos tomar dentro del ámbito profesional en el cual nos encontramos.

Hoy en día la construcción de sistemas en ambiente cliente – servidor ha cobrado una gran importancia debido a sus significativas ventajas frente a otros ambientes de desarrollo, entre las más importantes podemos mencionar, el rápido acceso a grandes volúmenes de datos, además de que se pueden desarrollar aplicaciones de manera ágil, atendiendo así, a la gran demanda de sistemas que las empresas y otros organismos tienen en la actualidad. Es necesario recalcar que al utilizar un ambiente Cliente – Servidor el sistema está preparado para atender cualquier modificación que se genere a partir de cambios de políticas dentro de la empresa, o del entorno con el que interactúe, esto es un punto medular, ya que cualquier sistema es susceptible a ser modificado debido a que los requerimientos de los usuarios son dinámicos.

Las metodologías de desarrollo permiten al Ingeniero en Computación durante la fase de diseño prever posibles cambios y adecuaciones posteriores, generando así, una reducción de costos al momento de desarrollar y dar mantenimiento al sistema,

además de acrecentar la rapidez de construcción. Esto se debe a que con el uso de metodologías, el Ingeniero en Computación tiene la posibilidad de observar desde un inicio, el contexto global del problema que se quiere resolver y es la premisa para identificar las diversas alternativas de solución que puedan existir.

El uso de herramientas de desarrollo de vanguardia es algo que se debe de tener muy en cuenta, debido a que las necesidades de las empresas hoy en día son muy cambiantes, por lo cual, se requiere el uso de herramientas que permitan al desarrollador atender con prontitud las necesidades del negocio. Sólo con las herramientas más actuales se tiene la certeza de que la solución generada no se volverá obsoleta en un periodo corto de tiempo, esto es importante debido a que la Ingeniería en Computación es una de las ramas del conocimiento que hoy en día avanza a pasos agigantados, y los sistemas que hoy son novedosos y eficientes, pueden dejar de serlo en unos cuantos meses.

El proponer alternativas de solución, es un factor que da a conocer a los desarrolladores de sistemas, las características de otros productos similares al que se va a construir, con lo cual se pueden tomar en cuenta características importantes que pueden ser implementados en el sistema a construir, además de que el usuario, pueda conocer y evaluar otros productos, genera confianza en este con respecto al sistema que se desarrolla, debido a que se tiene la certeza que es la mejor solución que se le puede dar a determinado problema, ya que el usuario compara el nuevo desarrollo con otros productos que se ofrecen en el mercado o con los que la misma empresa tiene en funcionamiento.

Después de la construcción y puesta en marcha del sistema contable podemos concluir que éste cubre totalmente las expectativas del usuario, ofreciéndole significativas ventajas en el procesamiento y manipulación de su información, las características más importantes que lo califican como una herramienta útil son las siguientes:

- Trabajo en un ambiente multiusuario.
- Utilización de un manejador de bases de datos robusto que permite una manipulación rápida de información, así como un alto grado de confiabilidad en la integridad y seguridad de la misma.
- Pantallas fáciles de manipular.
- Rígidos esquemas de seguridad de datos a fin de evitar que se corrompa o se pierda la información.
- Flexibilidad en la incorporación de nuevos datos.
- Agilización del flujo operativo de los usuarios, evitando pasos innecesarios.
- Manejo de varias contabilidades dentro de un mismo sistema, permitiendo independencia entre ellas a determinados niveles y una rígida dependencia entre las mismas en otros, a fin de poder generar estados financieros consolidados, esto se logra por medio del manejo del concepto de catálogo normativo.
- Generación de reportes financieros en cualquier momento.
- La dependencia de los usuarios con el área de sistemas es mínima, permitiendo al usuario trabajar sin la ayuda del personal de sistemas.
- El procesamiento de la información al realizarse en el servidor reduce de manera considerable los tiempos de respuesta.
- Al ser un sistema que tiene parametrizados los aspectos con mayor movilidad, es un sistema que asegura una rápida respuesta cuando se demande algún cambio por parte de los usuarios.

- Al tener documentados tanto el código del cliente, como la base de datos, el mantenimiento y la operación del sistema se verán reducidos en costos.
- El tráfico de datos en la red disminuye, debido a la realización de los procesos que manipulan grandes cantidades de datos, en el servidor.

El sistema a pesar de ser por sí sólo una herramienta contable completa, permitirá al desarrollador la creación de nuevos módulos a fin de que las tareas contables se vean agilizadas, el sistema está preparado para soportar nuevos módulos importantes dentro de una empresa y que tienen una gran ingerencia en el área contable, como el de inventarios, nómina, conciliaciones bancarias, manejo de cuentas colectivas, etc., esto gracias al uso de herramientas de desarrollo de sistemas modernas, así como el uso de metodologías que estructuran de manera correcta un sistema.

Dentro de las perspectivas del sistema a corto plazo, no sólo está la incorporación de nuevos módulos, sino la integración de todos los sistemas que existen en la empresa, esta tarea es primordial ya que de una u otra forma casi todo el movimiento que se realiza dentro de una institución implica manejo de dinero y por consiguiente un registro en la contabilidad.

Es así como podemos concluir de sin lugar a dudas que se logró construir una herramienta que hoy en día cubre las expectativas del usuario en el área de contabilidad, y que tiene la posibilidad de reducir y mejorar los flujos operativos no sólo dentro de esta área sino dentro de toda la empresa.

## **BIBLIOGRAFÍA**

- 1) Contabilidad Básica 7ª. Edición  
Joaquín Moreno Fernández.  
ANFECA
  
- 2) Formación Contable, Fiscal y Laboral  
Módulo I  
CEFA (Centro de Estudios Fiscales, S.C.)
  
- 3) Distributed Database Systems  
DavidBell, Jane Grimson. Addison Wesley.
  
- 4) Redes con Microsoft TCP/IP 2ª. Edición  
Drew Heywood  
Prentice Hall

- 5) Ingeniería del Software 4ª. Edición  
Roger S. Pressman  
Mc. Graw Hill
  
- 6) Using Sybase System XI  
Peter Hazlehurst  
QUE , USA 1996
  
- 7) Ptx /TCP/IP and Applications Overview  
Sequent Computer System, USA 1993
  
- 8) Client/Server Architecture  
Berson Alex  
McGraw-Hill 1992
  
- 9) Object Oriented Analisis and Desing with Applications  
Booch Grady  
Benjamin/Cummings Publishing, 1994
  
- 10) The Great Leap Forward (SQL Databases)  
Butler Brian  
PC Magazine Vol. 13 No. 17, 1994
  
- 11) Bases de datos, una guía práctica  
Date C.J.  
Addison-Wesly, 1987
  
- 12) Introducción a los sistemas de bases de datos  
Date C.J.  
Addison-Wesly, 1981

- 13) Client/Server Computing  
Dawna Travis  
McGraw-Hill, 1993
  
- 14) Introducción a los sistemas operativos  
Deitel Harvey M.  
Addison-Wesley, 1987
  
- 15) PowerBuilder 5 Unleashed  
Gallagher Simon  
Sams Publishing, 1996
  
- 16) Desarrollo de aplicaciones cliente/servidor con PowerBuilder  
Marler Paul  
Prentice Hall, 1995
  
- 17) Conecting to your Database  
Powersoft  
Powersoft, 1996
  
- 18) Getting Started  
Powersoft  
Powersoft, 1996
  
- 19) Project Primer  
Powersoft  
Powersoft, 1996
  
- 20) Técnicas de bases de datos  
ShaKuntala Atre  
Trillas, 1990

- 21) LAN times Encyclopedia of Networking  
Sheldon Tom  
McGraw-Hill, 1995
  
- 22) Ingeniería de software  
Sommerville Ian  
Addison-Wesley Iberoamericana, 1988
  
- 23) Redes de Ordenadores  
Tanenbaum Andrew S.  
Prentice-Hall, 1991
  
- 24) Sistemas Operativos Modernos  
Tanenbaum Andrew S.  
Prentice-Hall, 1992
  
- 25) Página Para La Búsqueda Del Concepto De Odbc  
<http://www.whatis.com/odbc.htm>
  
- 26) Página Para La Búsqueda Del Concepto De Ntfs (Sistema De Archivos De  
Windows Nt)  
<http://www.whatis.com/ntfs.htm>
  
- 27) Página Para La Búsqueda Del Concepto De Unix  
<http://www.whatis.com/unix.htm>
  
- 28) Página Para La Búsqueda De Conceptos De Power Builder  
<http://www.whatis.com/powerbui.htm>
  
- 29) Página Para La Búsqueda Del Concepto De Cliente/Servidor  
<http://www.whatis.com/clientse.htm>

30) Página Para La Búsqueda Del Concepto De Gui (Graphic User Interface)

<http://www.whatis.com/gui.htm>

31) Página Para La Búsqueda Del Concepto De Lan

<http://www.whatis.com/localare.htm>

32) Página Para La Búsqueda Del Concepto De Wan

<http://www.whatis.com/wan.htm>

33) Página Para La Búsqueda De Conceptos de Componentes de Red

<http://www.lannet.com>



# APÉNDICE A

## DIAGRAMA ENTIDAD-RELACIÓN

# APÉNDICE B

## MANUAL DEL USUARIO



## Contabilidad

**E**l Sistema de Contabilidad contiene los datos relativos al manejo de la Contabilidad del organismo así como la posibilidad de tener e introducir al sistema las pólizas que se requieren para la elaboración de los estados financieros que son necesarios en toda empresa.

## 1. Afectación Contable

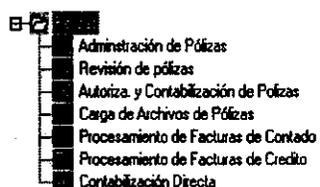
Dentro de la Afectación Contable tenemos la posibilidad de realizar las pólizas pertinentes que requerimos para el registro de nuestra contabilidad. El Sistema de Contabilidad nos permite agregar, eliminar, modificar y consultar una póliza.

El contenido de la opción Afectación Contable del Sistema de Contabilidad es el



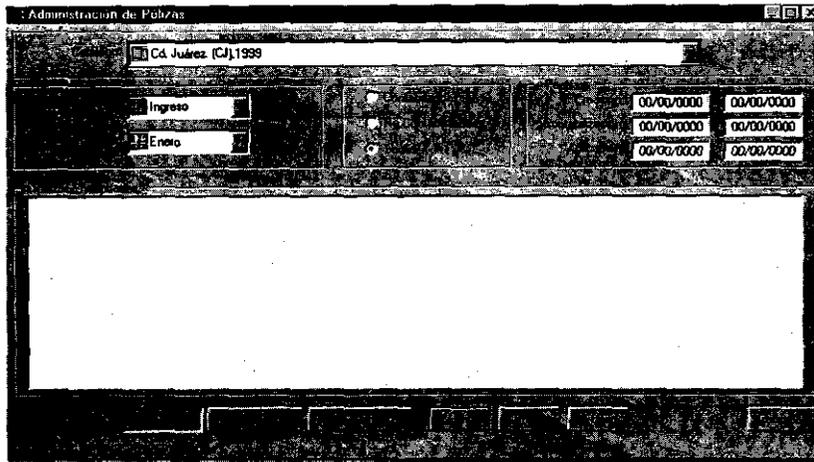
En **Pólizas** encontramos todas las opciones necesarias para el registro de las mismas.

El contenido de la opción **Pólizas** del Sistema de Contabilidad es el siguiente:



### 1.1 Administración de Pólizas

En esta opción el Sistema nos permite administrar todas las pólizas con las que cuenta nuestra base de datos. Vea la siguiente figura:



La ventana Administración de Pólizas contiene los siguientes elementos:

<b>Catálogo</b>	Permite seleccionar de la lista desplegable el catálogo con el que vamos a trabajar para elaborar las pólizas.	Selección
<b>Tipo de Póliza</b>	Lista desplegable con el nombre de la póliza que vamos a dar de alta, las opciones son: Ninguna, Ingreso, Egreso, Diario, Todas.	Selección
<b>Mes</b>	Presenta el mes con en el cual vamos a trabajar.	Selección
<b>Contabilizadas</b>	Permite seleccionar si se requiere visualizar las pólizas que están dentro del catálogo las cuales ya han sido contabilizadas.	Selección
<b>No contabilizadas</b>	Permite seleccionar si se requiere visualizar las pólizas que están dentro del catálogo las cuales no han sido contabilizadas.	Selección
<b>Todas</b>	Permite seleccionar si se requiere visualizar todas las pólizas que están dentro del catálogo de cuentas.	Selección
<b>Creación</b>	Fecha en que se crea la póliza.	Edición
<b>Contabilización</b>	Fecha en que se contabiliza la póliza.	Edición
<b>Aplicación</b>	Fecha en que la póliza se va a aplicar.	Edición

<b>Pólizas Encontradas</b>	Muestra todas las pólizas que han sido encontradas en base a los parámetros introducidos.	Consulta
<b>Pólizas Recuperadas</b>	Permite ver el número de pólizas que han sido recuperadas.	Consulta
<b>Inicializa Fechas</b>	Borra el rango de fechas que ha sido introducido dando la opción de volver a editar el campo.	Selección / Edición
<b>Recuperar Pólizas</b>	Realiza el proceso de recuperar pólizas que están dentro de nuestra contabilidad.	Selección
<b>Nueva</b>	Inicia el proceso de dar de alta una nueva póliza.	Selección
<b>Borrar</b>	Realiza el proceso de borrar una póliza que previamente a sido dada de alta.	Selección
<b>Editar</b>	Permite hacer modificaciones en una póliza que ya ha sido capturada.	Selección
<b>Cancelar</b>	Aborta el proceso de la administración de pólizas, es decir, cierra la ventana actual.	Selección

• **Propiedades de la Administración de Pólizas**

**Pasos a seguir:**

1. Haga doble *click* en **Administración de Pólizas**. El sistema presentará automáticamente la ventana del mismo nombre.
2. Haga *click* con el botón derecho del mouse en el área de **Pólizas Encontradas** de la ventana y seleccione una de las dos opciones que aparecen en ese menú, las opciones son: **Ver** y **Ordenado por**.
3. Haga *click* en **Ver** y seleccione una de las dos opciones de esa ventana **Lista** o **Detalle**.
4. En caso de seleccionar la opción **Lista**, el sistema presentará una lista completa de todas las pólizas encontradas.
5. Si usted selecciona **Detalle**, el sistema presentará los detalles de cada una de las pólizas encontradas. Vea la siguiente figura:

Administración de Pólizas

Cd. Juárez (CJ), 1999

Todos

Todos

00/00/0000 00/00/0000

00/00/0000 00/00/0000

00/00/0000 00/00/0000

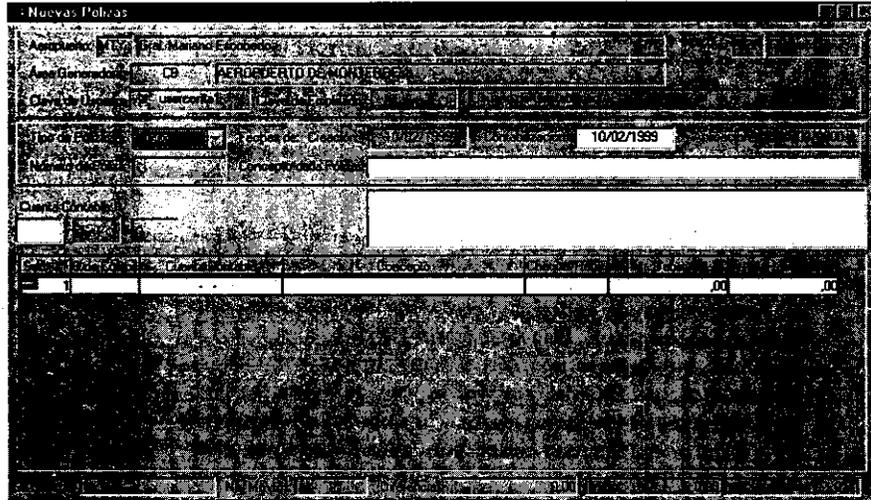
24-CJ	0-1-1	24-CJ	0-1-10	24-CJ	0-2-4	24-CJ	4-2-3	24-CJ	0-3-7
24-CJ	0-1-2	24-CJ	4-1-1	24-CJ	0-2-5	24-CJ	4-2-4	24-CJ	0-3-8
24-CJ	0-1-3	24-CJ	4-1-2	24-CJ	0-2-6	24-CJ	4-2-5	24-CJ	0-3-9
24-CJ	0-1-4	24-CJ	4-1-3	24-CJ	0-2-7	24-CJ	0-3-1	24-CJ	0-3-10
24-CJ	0-1-5	24-CJ	4-1-4	24-CJ	0-2-8	24-CJ	0-3-2	24-CJ	4-3-1
24-CJ	0-1-6	24-CJ	4-1-5	24-CJ	0-2-9	24-CJ	0-3-3	24-CJ	4-3-2
24-CJ	0-1-7	24-CJ	0-2-1	24-CJ	0-2-10	24-CJ	0-3-4	24-CJ	4-3-3
24-CJ	0-1-8	24-CJ	0-2-2	24-CJ	4-2-1	24-CJ	0-3-5	24-CJ	4-3-4
24-CJ	0-1-9	24-CJ	0-2-3	24-CJ	4-2-2	24-CJ	0-3-6	24-CJ	4-3-5

- La ventana le muestra los siguientes campos: Identificación ( la estructura de este campo es la siguiente: Catálogo al que se aplica - Area del catálogo - Tipo de póliza (que puede ser de Diario, Ingresos, Egresos o Unica) - Mes - Número de póliza.), Fecha de creación, Número de Póliza, Fecha de aplicación, Fecha de contabilización, Clave del área generadora, Concepto, Contabilizada, Elaboró, Autorizó, Revisó.
- Haga clic en Ordenar por, el sistema le presenta las siguientes opciones: by Identificación, by Fec. de creación, by No. de Póliza, by Fec. Aplicación, by Fec. de Contabilización, by Cve. Area Generad, by Concepto, by Elaboró, by Autorizó, by Revisó.
- Seleccione la opción que desee para que las pólizas sean ordenadas en base al campo que eligió.

- Registro de una Póliza

Pasos a seguir:

1. Haga doble *clic* en Administración de Pólizas. El sistema presentará automáticamente la ventana del mismo nombre.
2. Haga *clic* en el botón Nueva o bien utilice el método abreviado *Alt + N*. El sistema presentará la siguiente ventana:



La ventana Nueva Póliza contiene los siguientes elementos:

<b>Aeropuerto</b>	Muestra el designador del aeropuerto en donde se esta trabajando. El sistema lo presenta automáticamente.	Consulta
<b>Campo sin nombre</b>	Muestra el nombre completo del aeropuerto donde se esta trabajando. El sistema lo presenta automáticamente.	Consulta
<b>Mes</b>	Número y nombre completo del mes de trabajo. El sistema lo presenta automáticamente.	Consulta
<b>Area Generadora</b>	Clave y nombre del área en donde se genera la póliza. El sistema la presenta automáticamente.	Consulta
<b>Clave del Usuario</b>	Muestra la clave del usuario del sistema. Esta clave se presenta automáticamente.	Consulta
<b>Clave del</b>	Presenta la clave y nombre del empleado.	Consulta

Empleado		
<b>Tipo de Póliza</b>	Nos permite seleccionar el tipo de póliza a dar de alta, las opciones son: Ingresos, Egresos, Diario o Ninguno.	Selección
<b>Fecha de Creación</b>	Nos muestra la fecha en que se crea la cuenta, el sistema registra automáticamente la fecha del día en que se está trabajando.	Consulta
<b>Contabilización</b>	Fecha en que se requiere que la póliza sea contabilizada, el sistema nos presenta automáticamente la fecha actual de trabajo.	Edición
<b>Aplicación</b>	Fecha de aplicación de la póliza.	Consulta
<b>Número de Póliza</b>	Muestra el número de la nueva póliza, el sistema lo presenta automáticamente.	Consulta
<b>Concepto de la Póliza</b>	Permite introducir el concepto de la nueva póliza.	Edición
<b>Sec</b>	Números secuenciales de movimientos que estamos introduciendo, este campo, muestra una flecha que nos permite ubicar en que línea estamos trabajando.	Edición
<b>C. de Costos</b>	Muestra el número del centro de costos con el cual se esta trabajando.	Consulta
<b>Cuenta Contable</b>	Permite introducir el número de cuenta al cual se le va a carga o abonar cierta cantidad dentro de la póliza.	Edición / Selección
<b>Concepto</b>	Concepto por el cual la cuenta se va a afectar.	Edición
<b>Cheque / Refer</b>	Permite introducir si el cargo o abono que se esta haciendo tiene como referencia un cheque y en este campo puede poner el número de cheque o referencia.	Selección.
<b>Debe</b>	Cantidad de cargo para la cuenta introducida.	Edición
<b>Haber</b>	Cantidad de abono para la cuenta introducida.	Edición
<b>Clf. Control</b>	Suma de todas las cuentas contables de la póliza.	Consulta
<b>No. Movtos.</b>	Muestra el número de movimientos que tiene la póliza.	Consulta
<b>Diferencia</b>	Muestra las diferencias ( en caso de que existan ) entre los movimientos del debe y haber de la póliza.	Consulta

Adicionalmente, en la parte superior de la ventana, tenemos un conjunto de botones que

llamamos Barra de Herramientas la cual contiene las siguientes opciones:

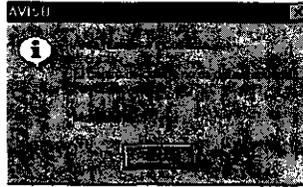


1 2 3

- 1.- Guardar. Permite guardar un registro nuevo o alguna modificación agregada.
- 2.- Agregar Registro. Permite agregar un nuevo registro o póliza.
- 3.- Borrar Registro. Elimina un registro o póliza una vez seleccionado.

Lo anterior, estará presente en el proceso de Agregar, Modificar y Eliminar Pólizas.

3. Seleccione el tipo de póliza a dar de alta en el campo **Tipo de Póliza** las opciones a elegir son: **Ingresos, Egresos, Diario, Unica o Cancelación**.
4. Cabe mencionar, que el tipo de póliza **Cancelación**, es usada únicamente al final del año contable para crear los movimientos de cancelación para las cuentas de resultados.
5. Introduzca la fecha en que desee que la póliza sea contabilizada en el campo **Contabilización**. El sistema presenta automáticamente la fecha del día en que esta trabajando.
6. Introduzca el concepto de la póliza en el campo del mismo nombre.
7. Seleccione el centro de costos en donde va a trabajar en la lista desplegable que aparece al lado derecho del campo **C de Costos**, o bien, si en el aeropuerto donde esta trabajando, sólo hay un centro de costos, el sistema lo pondrá automáticamente.
8. En caso de que introduzca un centro de costos no válido para el sistema, se desplegará un mensaje como el siguiente:



9. Oprima el botón **Aceptar** e introduzca un centro de costos válido para el aeropuerto.
10. Introduzca el número de cuenta en que va a realizar los movimientos necesarios en el campo **Cuenta Contable**, cabe mencionar que en ese campo no es posible modificar nada, sino que en la parte de arriba de la ventana aparecen tres pequeños cuadros en gris los cuales al seleccionar el campo en cuestión toman un color blanco.
11. En los recuadros superiores, introduzca el número de cuenta o bien presione **Ctrl + F1** y el sistema presentará el catálogo de cuentas para que se seleccione la cuenta deseada. Vea la siguiente figura:

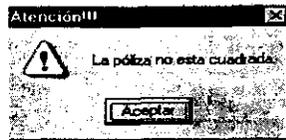
Catálogo de Cuentas Contables

Campo:

Cuenta de Balance:

Cuenta	Descripción de la Cuenta
1000000000	ACTIVO CIRCULANTE
1010000000	DISPONIBLE
1020000000	INVERSIONES EN VALORES
1030000000	CUENTAS POR COBRAR
1040000000	OTRAS CUENTAS POR COBRAR
1050000000	ESTIMACIÓN CTAS DE COBRO DUDOSO
1060000000	ANTICIPOS
1070000000	IVA ACREDITABLE
1080000000	INVENTARIOS

12. Introduzca el concepto del movimiento de la cuenta que esta afectando en el campo **Concepto**, si el concepto es el mismo que el de la póliza presione **CTRL + F3** para repetir el texto.
13. En caso de existir una referencia o un cheque con el cual se soporte el movimiento de la cuenta, introduzca el número correspondiente en el campo **Cheque / Refer.**
14. Introduzca la cantidad de cargo de la cuenta seleccionada en el campo **Debe**.
15. Introduzca la cantidad de abono de la cuenta seleccionada en el campo **Haber**.
16. En el campo **Sec.** podrá observar que conforme va capturando movimientos, el sistema va colocando un número secuencial a la línea, así como también la fecha va pasando de un renglón a otro.
17. Así mismo, si usted desea insertar una línea entre las que ya capturó:
  - 17a. Coloque el cursor en el campo **Concepto** y haga *click* con el botón derecho del mouse y seleccione la opción **Agregar**, o bien, en la barra de herramientas haga *click* en el botón **Agregar**, también puede hacer *click* en **Edición del Panel de control** y seleccionar la opción **Agregar Registro**. El sistema insertará una línea en la parte superior de la línea donde esta colocado actualmente.
  - 17b. Coloque el cursor en el campo **Debe** o en el campo **Haber** y haga *click* con el botón derecho del mouse y seleccione la opción **Agregar**, o bien, en la barra de herramientas haga *click* en el botón **Agregar**, también puede hacer *click* en **Edición del Panel de control** y seleccionar la opción **Agregar Registro**. El sistema insertará una línea en la parte inferior de la línea donde esta colocado actualmente.
18. En caso de que el sistema detecte que la sumatoria de cargos y la de abonos no son iguales, se desplegará el siguiente mensaje:



Haga *click* en el botón *Aceptar* y verifique las cantidades introducidas.

19. Guarde los cambios realizados haciendo *click* en el botón *Guardar* de la barra de herramientas o seleccionando del menú la opción *Guardar* o bien utilice el método abreviado *Ctrl + G*.

• **Modificación de una Póliza**

Pasos a seguir:

1. Haga doble *click* en *Administración de Pólizas*. El sistema presentará automáticamente la ventana del mismo nombre.
2. Haga *click* en el botón *Editar* o utilice el método abreviado *Alt + E* y el sistema le presentará la siguiente ventana.

Ventana de Pólizas					
28/02/1998					
PÓLIZAS A I C M					
<input type="checkbox"/> 201 PASIVO CIRCULANTE <input type="checkbox"/> 012 CORRECCION POR REEXPRESION <input type="checkbox"/> 0003S PATRIMONIO AEROPUERTO					
128	201-012-0003S	DETERM REPOMO INSTIT		54.710.400,00	00
128	405-002-00001	<<ERROR>> DETERM REPOMO INSTIT		,00	54.710.400,00
128	605-001-00001	CANA PARC POL MEX 01031		,00	2.003,68
128	603-001-00001	CANA PARC POL MEX 01031		,00	-3.000,00
128	604-001-00001	CANA PARC POL MEX 01031		,00	909.697,04
128	703-001-00006	CANA PARC POL MEX 01031		,00	11.636.711,73
128	703-001-00005	CANA PARC POL MEX 01031		,00	17.283.581,17
128	302-005-00058	CANA PARC POL MEX 01031		,00	-23.888.003,68
128	132-007-00001	TR COMB A OFNAS GRALES		-1.962.304,00	,00
128	134-006-00001	TR COMB A OFNAS GRALES		-45.426,00	,00

3. Haga *click* sobre el campo que desee modificar e introduzca el nuevo valor para ese campo. Cabe mencionar que una póliza podrá ser modificada, únicamente si esta no ha sido contabilizada, en caso contrario, el sistema no le permitirá cambiar ningún campo.
4. Guarde los cambios realizados haciendo *click* en el botón *Guardar* de la barra de

---

herramientas o seleccionando del menú la opción **Guardar** o bien utilice el método abreviado *Ctrl + G*.

---

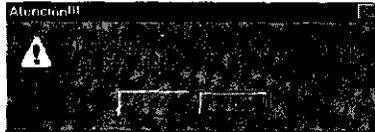
- **Eliminación de una Póliza**

**Pasos a seguir:**

1. Haga doble *click* en **Administración de Pólizas**. El sistema presentará automáticamente la ventana del mismo nombre.
2. Haga *click* sobre la póliza que desee borrar. Cabe mencionar que sólo podrá eliminar una póliza si esta no esta contabilizada, en caso contrario el sistema presentará el siguiente mensaje.



3. Si la póliza no esta contabilizada, haga *click* en el botón **Borrar** o utilice el método abreviado *Alt + B*, o bien haga *click* en el botón **Borrar** de la barra de herramientas. Como confirmación de la operación requerida, el sistema presentará el siguiente mensaje:



Si esta seguro que desea eliminar la póliza, haga *click* en **Aceptar**, de lo contrario oprima el botón **Cancelar**.

---

- **Elaboración de un Reporte de Pólizas**

**Pasos a seguir:**

1. Haga doble *click* en **Administración de Pólizas**. Automáticamente el sistema presentará la ventana del mismo nombre.

2. Seleccione la póliza a imprimir haciendo clic en la columna Identificación de la ventana y presione el botón Editar, el sistema le presentará la póliza que usted eligió.
3. Oprima el botón Presentación Preliminar de la barra de herramientas y el sistema le mostrará la presentación preliminar del reporte. Ve a la siguiente figura:

Área Cuenta	Descripción de la Cuenta	Cheque	Concepto del Movimiento	
21 101-003-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	540.1
21 101-003-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	
21 104-005-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	
21 107-001-00001	IVA ACREDITABLE TASA 16%		BALANZA DE COMPR MZO 98 ZLO	45.2
21 107-001-00001	IVA ACREDITABLE TASA 16%		BALANZA DE COMPR MZO 98 ZLO	
21 111-001-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	81.2
21 111-001-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	
21 201-005-00014	2 AL MILLAR CAM NAL IND CONSTRUCCION		BALANZA DE COMPR MZO 98 ZLO	

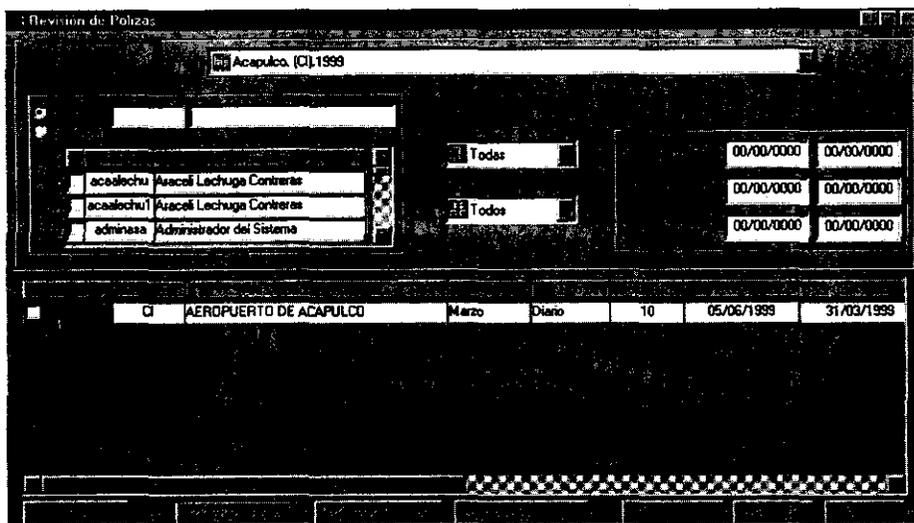
- Impresión del Reporte de Pólizas

Paso a seguir:

1. Haga clic en el botón Imprimir de la barra de herramientas o en el botón Impresión Directa de la misma o bien seleccione del menú principal la opción Imprimir o utilice el método abreviado *Alt + P*.

### 1.1.2 Revisión de Pólizas

En esta opción el Sistema nos permite revisar todas las pólizas con las que cuenta nuestra base de datos, cabe mencionar que el usuario que proceda a la revisión deberá ser un usuario distinto del que capturó la póliza. Vea la siguiente figura:



La ventana Revisión de Pólizas contiene los siguientes elementos:

<b>Catálogo</b>	Permite seleccionar de la lista desplegable el catálogo con el que vamos a trabajar para elaborar las pólizas.	Selección
<b>Uno</b>	Permite introducir el nombre o clave del usuario del que deseamos ver sus pólizas.	Selección
<b>Selección</b>	Permite seleccionar varios usuarios de los que deseamos ver sus pólizas.	Selección
<b>Tipo de Póliza</b>	Lista desplegable con el nombre de la póliza que vamos a dar de alta, las opciones son: Ninguna, Ingreso, Egreso, Diario, Todas.	Selección
<b>Mes</b>	Presenta el mes con el cual vamos a trabajar.	Selección

## Contabilidad

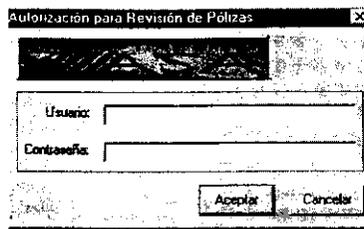
<b>Creación</b>	Fecha en que se crea la póliza.	Edición
<b>Contabilización</b>	Fecha en que se contabiliza la póliza.	Edición
<b>Aplicación</b>	Fecha en que la póliza se va a aplicar.	Edición
<b>Revisa</b>	Permite seleccionar las pólizas a revisar.	Selección
<b>Cve. Area</b>	Muestra la clave del área de ubicación de la póliza.	Consulta
<b>Area Creadora de la Póliza</b>	Muestra el área donde fue creada la póliza.	Consulta
<b>Mes</b>	Muestra el mes en que fue hecha la póliza.	Consulta
<b>Tipo</b>	Muestra el tipo de póliza, puede ser de <b>Ingresos, Egresos, Diario o Unica.</b>	Consulta
<b>Póliza No.</b>	Muestra el número consecutivo asignado a la póliza en base a su tipo.	Consulta
<b>Fecha de Creación</b>	Muestra la fecha de creación de la póliza.	Consulta
<b>Fec. Contabilización</b>	Muestra la fecha de contabilización de la póliza.	Consulta
<b>Concepto</b>	Muestra el concepto de la póliza.	Consulta
<b>Núm. Movtos.</b>	Muestra el número de movimientos que contiene la póliza.	Consulta
<b>Total Debe</b>	Cantidad total de movimientos de cargos dentro de la póliza.	Consulta
<b>Total Haber</b>	Cantidad total de movimientos de abonos dentro de la póliza.	Consulta
<b>Cifras de Control</b>	Cifras de control asignada a la póliza.	Consulta
<b>Elaboró</b>	Usuario que elaboró la póliza.	Consulta
<b>Inicializa Fechas</b>	Borra el rango de fechas que ha sido introducido dando la opción de volver a editar el campo.	Selección / Edición
<b>Recuperar Pólizas</b>	Realiza el proceso de recuperar pólizas que están dentro de nuestra contabilidad.	Selección
<b>Seleccionar Todas</b>	Selecciona todas las pólizas que se encuentran dentro del catálogo actual.	Selección
<b>Deseleccionar Todas</b>	Elimina la selección de todas las pólizas.	Selección
<b>Aplica Revisión</b>	Inicia el proceso de revisión a las pólizas seleccionadas.	Selección
<b>Consultar</b>	Permite ver el contenido de la póliza seleccionada..	Selección
<b>Terminar</b>	Cierra la ventana <b>Revisión de Pólizas</b> y nos regresa al menú principal.	Selección

---

- Revisión de una Póliza

Pasos a seguir:

20. Haga doble *clic* en **Revisión de Pólizas**. El sistema presentará automáticamente la ventana del mismo nombre.
21. Haga *clic* en el campo **Catálogo** y seleccione el mismo de la lista desplegable que aparece del lado derecho.
22. Seleccione el campo **Uno** en caso de que quiera ver las pólizas de un sólo usuario, en caso contrario elija el campo **Selección** para solicitar la lista de usuarios de los cuales desea revisar sus pólizas.
23. Seleccione el tipo de pólizas que desea ver en el campo **Tipo**, estas pueden ser de **Diario, Ingresos, Egresos o Unicas**.
24. Seleccione el mes que desee consultar en el campo **Mes**.
25. Introduzca las fechas de creación, contabilización y aplicación del periodo que desee en los campos del mismo nombre.
26. Haga clic en el botón **Recuperar Pólizas** y el sistema le presentará en pantalla las pólizas encontradas basándose en los criterios señalados.
27. Seleccione en el campo **Revisa la (s) póliza (s)** a revisar.
28. Haga clic en el botón **Aplica Revisión**, el sistema le presentará la siguiente ventana:



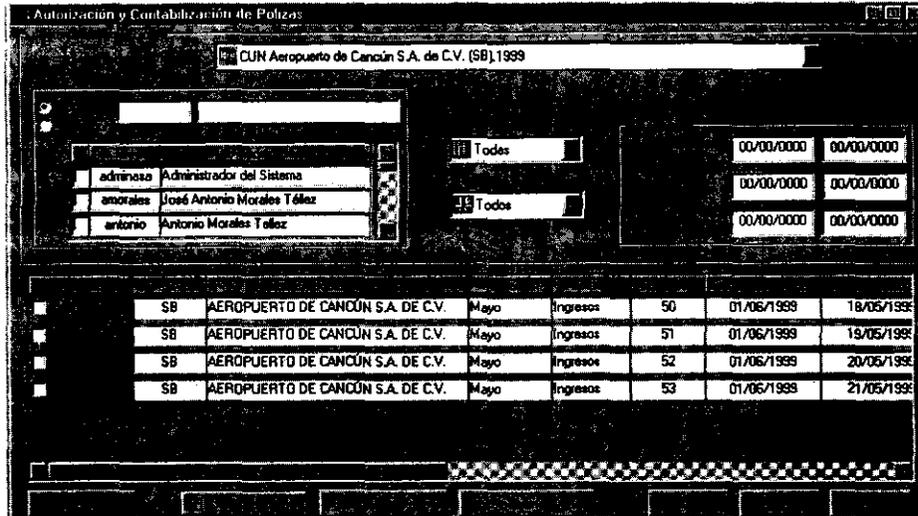
29. Introduzca el nombre del usuario en el campo **Usuario** y la **Contraseña** del mismo y haga clic en el botón **Aceptar**. Lo anterior es una medida de seguridad para proceder a la revisión de las pólizas.
30. Una vez concluido el proceso de revisión, el sistema le presentará el siguiente mensaje:



31. Haga clic en el botón **Aceptar** y el proceso habrá concluido.

### 1.1.3 Autorización y Contabilización de Pólizas

En esta opción el Sistema nos permite autorizar y contabilizar todas las pólizas con las que cuenta nuestra base de datos, cabe mencionar que el usuario que proceda a la autorización y contabilización deberá ser un usuario distinto del que revisó la póliza. Vea la siguiente figura:



La ventana Autorización y Contabilización de Pólizas contiene los siguientes elementos:

Elemento	Descripción	Acción
Catálogo	Permite seleccionar de la lista desplegable el catálogo con el que vamos a trabajar para elaborar las pólizas.	Selección
Uno	Permite introducir el nombre o clave del usuario del que deseamos ver sus pólizas.	Selección
Selección	Permite seleccionar varios usuarios de los que deseamos ver sus pólizas.	Selección
Tipo de Póliza	Lista desplegable con el nombre de la póliza que vamos a dar de alta, las opciones son: Ninguna, Ingreso, Egreso, Diario, Todas.	Selección
Mes	Presenta el mes con el cual vamos a trabajar.	Selección
Creación	Fecha en que se crea la póliza.	Edición

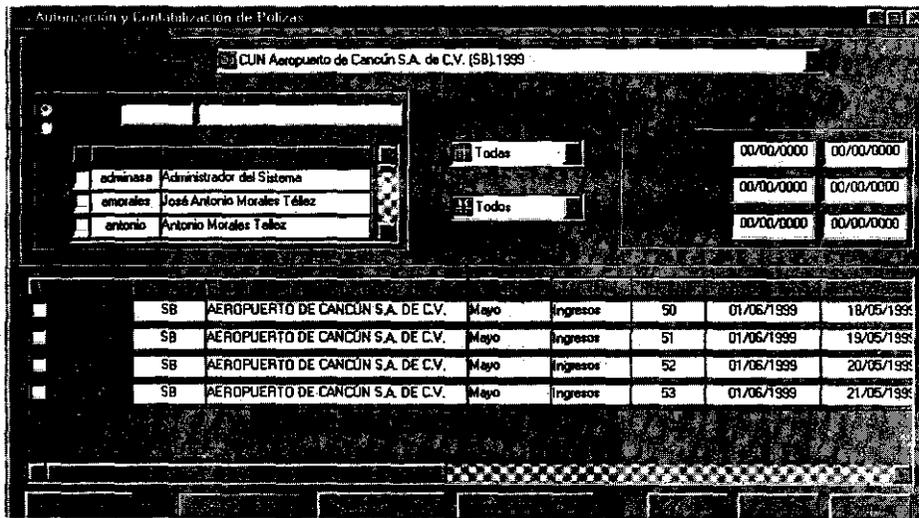
## Contabilidad

Contabilización	Fecha en que se contabiliza la póliza.	Edición
Aplicación	Fecha en que la póliza se va a aplicar.	Edición
<b>Revisa</b>	Permite seleccionar las pólizas a revisar.	Selección
<b>Cve. Area</b>	Muestra la clave del área de ubicación de la póliza.	Consulta
<b>Area Creadora de la Póliza</b>	Muestra el área donde fue creada la póliza.	Consulta
<b>Mes</b>	Muestra el mes en que fue hecha la póliza.	Consulta
<b>Tipo</b>	Muestra el tipo de póliza, puede ser de <b>Ingresos, Egresos, Diario o Unica.</b>	Consulta
<b>Póliza No.</b>	Muestra el número consecutivo asignado a la póliza en base a su tipo.	Consulta
<b>Fecha de Creación</b>	Muestra la fecha de creación de la póliza.	Consulta
<b>Fec. Contabilización</b>	Muestra la fecha de contabilización de la póliza.	Consulta
<b>Concepto</b>	Muestra el concepto de la póliza.	Consulta
<b>Núm. Movtos.</b>	Muestra el número de movimientos que contiene la póliza.	Consulta
<b>Total Debe</b>	Cantidad total de movimientos de cargos dentro de la póliza.	Consulta
<b>Total Haber</b>	Cantidad total de movimientos de abonos dentro de la póliza.	Consulta
<b>Cifras de Control</b>	Cifras de control asignada a la póliza.	Consulta
<b>Elaboró</b>	Usuario que elaboró la póliza.	Consulta
<b>Revisó</b>	Clave del usuario que revisó la póliza.	Consulta
<b>Inicializa Fechas</b>	Borra el rango de fechas que ha sido introducido dando la opción de volver a editar el campo.	Selección / Edición
<b>Recuperar Pólizas</b>	Realiza el proceso de recuperar pólizas que están dentro de nuestra contabilidad.	Selección
<b>Seleccionar Todas</b>	Selecciona todas las pólizas que se encuentran dentro del catálogo actual.	Selección
<b>Deseleccionar Todas</b>	Elimina la selección de todas las pólizas.	Selección
<b>Contabiliza</b>	Inicia el proceso de contabilización a las pólizas seleccionadas.	Selección
<b>Consultar</b>	Permite ver el contenido de la póliza seleccionada.	Selección
<b>Terminar</b>	Cierra la ventana <b>Revisión de Pólizas</b> y nos regresa al menú principal.	Selección

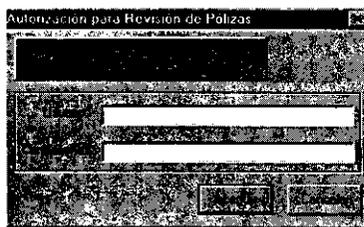
- Autorización y Contabilización de una Póliza

Pasos a seguir:

32. Haga doble *click* en Autorización y Contabilización de Pólizas. El sistema presentará automáticamente la ventana del mismo nombre. Vea la siguiente figura:



33. Haga *click* en el campo **Catálogo** y seleccione el mismo de la lista desplegable que aparece del lado derecho.
34. Haga *click* en el campo **Area Creadora de la Póliza** y seleccione la misma de la lista desplegable que aparece del lado derecho.
35. Seleccione el campo **Uno** en caso de que quiera ver las pólizas de un sólo usuario, en caso contrario elija el campo **Selección** para solicitar la lista de usuarios de los cuales desea revisar sus pólizas.
36. Seleccione el tipo de pólizas que desea ver en el campo **Tipo**, estas pueden ser de **Diario, Ingresos, Egresos o Unicas**.
37. Seleccione el mes que desee consultar en el campo **Mes**.
38. Introduzca las fechas de creación, contabilización y aplicación del periodo que desee en los campos del mismo nombre.
39. Haga *click* en el botón **Recuperar Pólizas** y el sistema le presentará en pantalla las pólizas encontradas basándose en los criterios señalados.
40. Seleccione en el campo **Revisa la (s) póliza (s)** a contabilizar, cabe mencionar que las pólizas que vemos en esta ventana, son únicamente aquellas que ya han sido revisadas.
41. Haga *click* en el botón **Contabiliza**, el sistema le presentará la siguiente ventana:



42. Introduzca el nombre del usuario en el campo **Usuario** y la **Contraseña** del mismo y haga *click* en el botón **Aceptar**. Lo anterior es una medida de seguridad para proceder a la actualización y contabilización de las pólizas.
43. Una vez concluido el proceso de revisión, el sistema le presentará el siguiente mensaje:

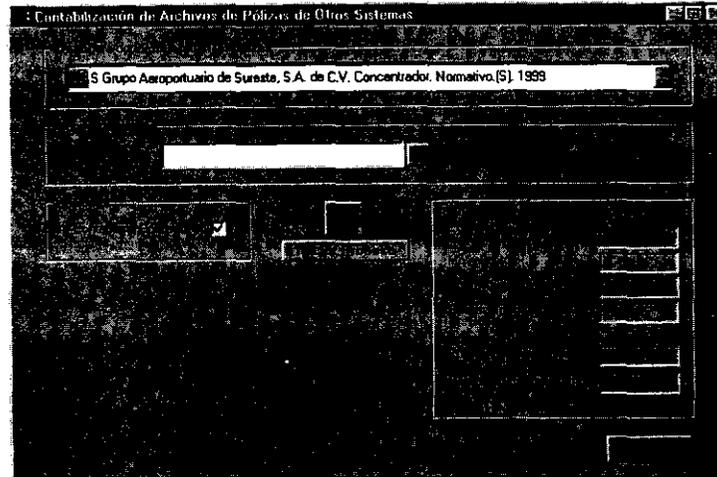


44. Haga *click* en el botón **Aceptar** y el proceso habrá concluido.

### 1.1.4. Carga de Archivos de Pólizas

En Afectación Contable tenemos la posibilidad de cargar archivos de pólizas externos al sistema.

Esta opción del Sistema nos presenta la siguiente figura:



La ventana **Contabilización de Archivos de Pólizas de Otros Sistemas** contiene los siguientes elementos:

<b>Catálogo a donde se aplican las pólizas</b>	Muestra el nombre del Catálogo al que se van a cargar las pólizas que van a ser importadas desde algún archivo de texto.	Selección
<b>Archivo de Pólizas</b>	Muestra la ruta de donde vamos a importar el archivo de pólizas.	Edición / Selección
<b>Contabilizar Pólizas</b>	Permite seleccionar si deseamos contabilizar las pólizas.	Selección
<b>Iniciar Proceso</b>	Procede a iniciar e proceso de importación del catálogo de pólizas desde un archivo externo.	Selección

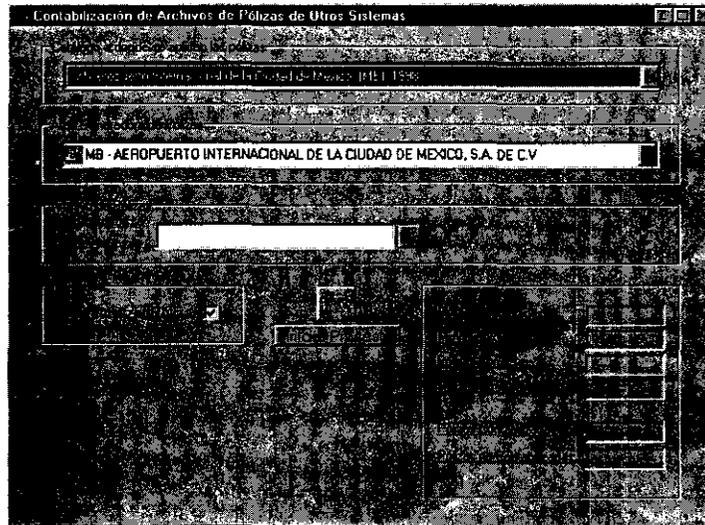
Contabilidad

<b>Pólizas Procesadas</b>	Muestra el total de pólizas procesadas.	Consulta
<b>Movimientos</b>	Muestra el total de movimientos de cada póliza procesados.	Consulta
<b>Pólizas Correctas</b>	Muestra el total de pólizas que han sido procesadas correctamente.	Consulta
<b>Pólizas Erróneas</b>	Muestra el total de pólizas que han sido procesadas de manera incorrecta.	Consulta
<b>Pólizas Contabilizadas</b>	Muestra el número total de pólizas contabilizadas.	Consulta
<b>Pólizas no Contabilizadas</b>	Muestra el número total de pólizas no contabilizadas.	Consulta

- **Carga de un Archivo de Pólizas**

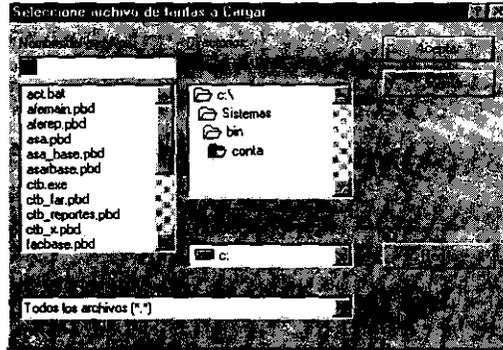
Pasos a seguir:

1. Haga doble *click* en **Carga de Archivos de Pólizas**. El sistema presentará automáticamente la ventana **Carga de Archivos de Pólizas de Otros Sistemas**. Vea la siguiente figura:

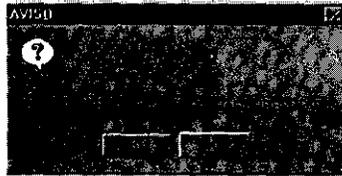


2. Haga *click* en el campo **Catálogo** a donde se aplican las pólizas y seleccione de la lista desplegable el nombre del catálogo a cargar.

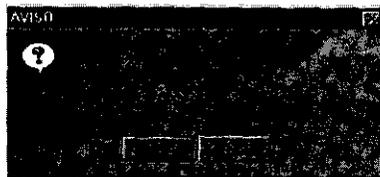
3. Haga *click* en el botón derecho del campo Archivo de pólizas, el Sistema le presentará la siguiente ventana:



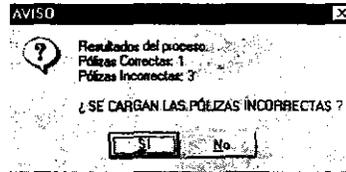
4. Escriba el nombre del archivo de texto de donde se van a importar las pólizas y seleccione la ruta de acceso al mismo, en caso de ser un archivo en un equipo que este dentro de la red haga *click* en el botón Red y seleccione la ruta deseada.
5. Seleccione si desea Contabilizar Pólizas en el campo del mismo nombre.
6. Haga *click* en el botón Iniciar Proceso para que el sistema proceda a la recuperación de las pólizas y espere a que el proceso termine. El sistema presentará un mensaje como el siguiente:



7. Haga *click* en el botón Si en caso de que quiera cargar la cantidad de pólizas que nos reporta el sistema como correctas, en caso contrario oprima el botón No.
8. El sistema presentará también la siguiente ventana:



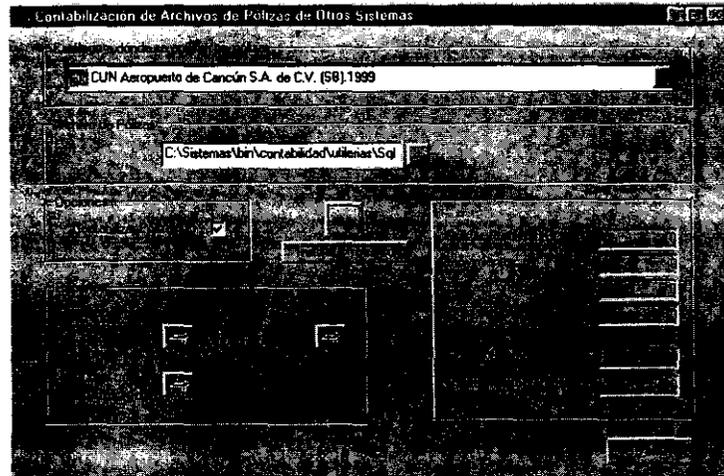
9. Oprima el botón **Si** en caso de que desee que las pólizas que están correctas sean contabilizadas, en caso contrario, presione el botón **No**.
10. Adicionalmente se presentará el siguiente mensaje:



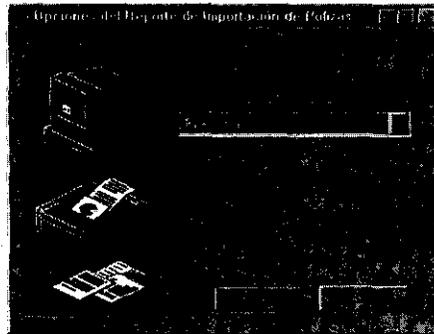
11. Haga clic en **Si** si desea que las pólizas incorrectas encontradas sean cargadas, en caso contrario presione el botón **No**.
12. Si el sistema no localiza ninguna póliza errónea los mensajes anteriores no serán presentados en pantalla y el proceso de contabilización ( en caso de haber sido seleccionado ) seguirá su proceso hasta finalizar.
13. Cuando el proceso quede completamente terminado el sistema presentará la siguiente ventana:



14. Haga clic en el botón **Aceptar** y observe la siguiente figura:



15. Note que la ventana presenta nuevas opciones en la parte inferior izquierda, en esta área es donde podemos imprimir todo lo que requiramos de los archivos de pólizas recién cargados.
16. La opción **Resultados de la Importación (Detalle)** presenta la siguiente ventana:



17. De la lista desplegable que aparece del lado derecho del campo **Pólizas a Visualizar** elija una de las opciones **Ambas**, **Correctas** o **Incorrectas**.
18. Haga clic en **Aceptar** y el sistema le desplegará la **Presentación Preliminar del reporte**. Vea la siguiente figura.

Sistema ASA

**SISTEMA DE CONTABILIDAD**  
DETALLE DEL PROCESAMIENTO DE LOS ARCHIVOS DE PÓLIZAS

Aeropuerto: MEX Internacional De La Ciudad De Mexico

ESTADO DE LAS PÓLIZAS: INCORRECTAS

ESTADO CONTABLE: NO CONTABILIZADAS

Tipo de Póliza: DIARIO

Tipo de Póliza: DIARIO No. de Póliza Archivo: 12001 No. de Póliza ASA: 2  
 Área Generadora: MB AEROPUERTO INTERNACIONAL DE LA CIUDAD DE MEXICO, S.A. DE C.V.  
 Concepto: COBZA 91-12-97 012001  
 Observaciones: Diferencia entre SUMA DE CARGOS y SUMA DE ABONOS

Creación  
10/17/1998

Cuenta	Descripción de la Cuenta	Concepto del Movimiento	Debe	Haber
10100201002	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ER ASA	10,000.00	0.00
10100201001	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ER ASA	0.00	111,038.12
10100301002	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ER ASA	175,848.31	0.00
10200101001	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ER ASA	10,972.29	0.00
10400210787	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ER ASA	0.00	11,850.30
10400211001	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ER ASA	100.00	0.00

19. La opción Resultados de la Importación ( Concentrado ) muestra una *Presentación Preliminar* del reporte. Vea la siguiente ventana:

Sistema ASA

**RESUMEN DEL PROCESO DE IMPORTACIÓN**

Correctas/Incorrectas	Contabilizadas / No contabilizadas	D	Gran Total
Incorrectas	No Contabilizadas	3	3
<b>Total Incorrectas</b>		<b>3</b>	<b>3</b>
Correctas	Contabilizadas	1	1
<b>Total Correctas</b>		<b>1</b>	<b>1</b>
<b>Gran Total</b>		<b>4</b>	<b>4</b>

20. La opción Pólizas no Contabilizadas nos muestra una *Presentación Preliminar* de las mismas.
21. Cabe mencionar que todos estos reportes pueden ser impresos como se describió en la parte de Reportes de Pólizas.

22. Es importante mencionar que todos los reportes que requiera de la importación deberán ser impresos antes de cerrar la ventana Carga de Archivos de Pólizas de Otros Sistemas ya que después no le será posible.

## 2. Cierre Contable

Dentro de la Contabilidad de una empresa, requerimos de hacer Cierres contables para poder observar los resultados de la misma y ver el comportamiento de nuestros recursos.

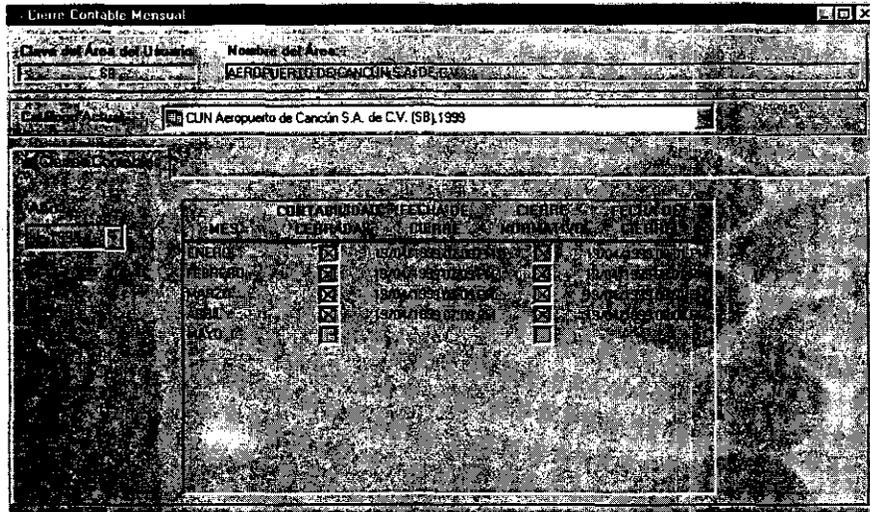
El Sistema de Contabilidad provee de una opción para realizar este proceso.

El contenido de la opción Cierre Contable es el siguiente:



### 2.1 Cierre contable Mensual

La opción **Cierre Contable Mensual** nos presenta la siguiente ventana:



La ventana **Cierre Contable Mensual** contiene los siguientes elementos:

<b>Clave del Área del Usuario</b>	Muestra la clave del área de trabajo del usuario.	Consulta
<b>Nombre del Área</b>	Nombre completo del área de trabajo donde se encuentra el usuario.	Consulta

<b>Catálogo Actual</b>	Permite seleccionar el catálogo con el cual vamos a trabajar.	Selección
<b>Año</b>	Permite seleccionar el año en el cual vamos a trabajar para iniciar nuestro cierre contable.	Selección
<b>Mes</b>	Muestra el mes al cual vamos a aplicar el cierre contable.	Consulta
<b>Contabilidad Cerrada</b>	Permite seleccionar el mes que queremos cerrar contablemente.	Selección
<b>Fecha de Cierre</b>	Muestra la fecha en que el cierre se esta llevando a cabo, el sistema toma la fecha actual para mostrarla.	Consulta
<b>Cierre Normativo</b>	Permite seleccionar el mes que queremos cerrar contablemente en el ámbito normativo.	Selección
<b>Fecha de Cierre</b>	Fecha en que se genera el cierre contable, el sistema toma la fecha actual.	Consulta

#### • Proceso de Cierre Contable Mensual

##### Pasos a seguir:

45. Haga doble *click* en **Cierre Contable Mensual**. El sistema presentará automáticamente la ventana del mismo nombre.
46. Seleccione el **Catálogo Actual** con el que se va a trabajar en el campo del mismo nombre.
47. Seleccione el **Año** en que se va a trabajar el Cierre en el campo del mismo nombre.
48. Seleccione el mes al que va a aplicar el cierre haciendo *click* en el cuadro derecho de la columna **Contabilidad Cerrada**. El sistema presentará la siguiente ventana:



- 5 Oprima el botón **Aceptar** y el sistema pondrá la fecha actual del sistema.
- 6 En caso de haber cometido un error, o bien su desea que el cierre en ese mes no sea llevado a cabo, vuelva a oprimir el recuadro derecho del mes elegido, el sistema

presentará la siguiente ventana:



7Haga *click* en el botón **Aceptar** y el sistema volverá a abrir la contabilidad del mes seleccionado.

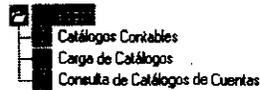
### 3. Catálogos Contables.

Los catálogos contables son aquellos que nos permiten controlar las cuentas que utilizamos para elaborar nuestra contabilidad. El Sistema de Contabilidad nos permite agregar, eliminar, modificar y consultar una cuenta. Cabe mencionar que tenemos tres tipos de catálogos de cuentas:

- 1.- Un Catálogo Normativo, que norma a todos los demás catálogos.
- 2.- Catálogos Concentradores, los cuales no contienen cuentas de detalle.
- 3.- Catálogos Generales, que si contienen cuentas de detalle.

#### 3.1 Catálogos.

El contenido de la opción Catálogos del Sistema de Contabilidad es el siguiente:



La opción **Catálogos Contables** nos permite controlar las cuentas necesarias para el registro de nuestra contabilidad.

En **Carga de Catálogos** tenemos la posibilidad de cargar las cuentas de detalle desde un archivo texto a algún catálogo en particular.

La opción **Consulta de Catálogos de Cuentas** contiene todos nuestros catálogos para que puedan ser consultados.

##### 3.1.1 Catálogos Contables.

La ventana **Catálogos de Cuentas Contables** nos permite agregar, eliminar, modificar y consultar alguna(s) cuenta (s). Vea la siguiente figura:

Catálogos de Cuentas Contables	
Manzanillo (PF) 1998	
130-000-00000	ACTIVO FLUJO
160-000-00000	OTROS ACTIVOS
200-000-00000	PASIVO
300-000-00000	CUENTAS DE ENLACE
400-000-00000	PATRIMONIO
500-000-00000	CUENTAS DE ORDEN
600-000-00000	INGRESOS DE OPERACION
700-000-00000	OTROS INGRESOS Y PRODUCTOS
800-000-00000	GASTOS DE OPERACION
999-999-99999	prueba
101-000-00000	DISPONIBLE
102-000-00000	INVERSIONES EN VALDRES
103-000-00000	CUENTAS POR COBRAR
104-000-00000	OTRAS CUENTAS POR COBRAR
105-000-00000	ESTIMACION CTAS DE COBRO DUDOSO
106-000-00000	ANTICIPOS
107-000-00000	IVA ACREDITABLE
108-000-00000	INVENTARIOS
109-000-00000	ESTIMACION COMBUSTIBLES POCO MOVIMIENTO
110-000-00000	ACTUALIZACION RESERVA COMB POCO MOV
111-000-00000	OTROS INVENTARIOS
113-000-00000	ESTIMACION MATERIALES POCO MOVIMIENTO
114-000-00000	ACTUAL RESERVA MATERIAL POCO MOVIMIENTO

La ventana Catalogo de Cuentas Contables contiene los siguientes elementos:

Campo	Descripción	Tipo de Campo
Clave del Area del Usuario	Identificador único del área de trabajo del usuario.	Consulta
Nombre del Area	Nombre completo del área de trabajo del usuario.	Consulta
Catálogo Actual	Lista desplegable con el nombre del catálogo con el cual estamos trabajando.	Edición / Selección
Columna sin nombre	Muestra las cuentas que contiene el catálogo actual.	Consulta / Selección
Cuenta	Muestra el número de cuenta de detalle del catálogo general.	Consulta / Selección
Descripción	Nombre completo de la cuenta de detalle del catálogo general.	Consulta

Adicionalmente en la parte superior de la ventana, tenemos un conjunto de botones que llamamos Barra de Herramientas la cual contiene las siguientes opciones:

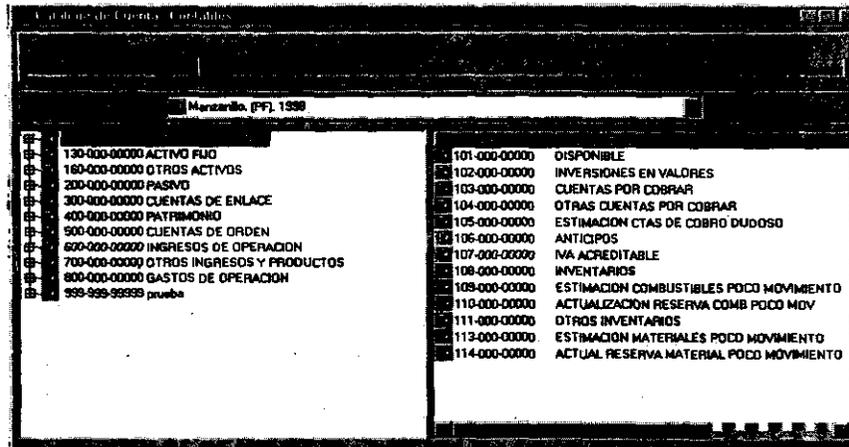
- |                  |   |   |  |
|------------------|---|---|--|
| 1                | 2   | 3 |  |
| ↓                | ↓   | ↓ |  |
| 1.- Nuevo Nivel. | Permite adicionar un nuevo nivel de cuenta. |   |  |
| 2.- Eliminar.    | Permite eliminar una cuenta.                |   |  |
| 3.- Modificar.   | Permite modificar una cuenta.               |   |  |

Lo anterior estará presente siempre que esta ventana sea activada.

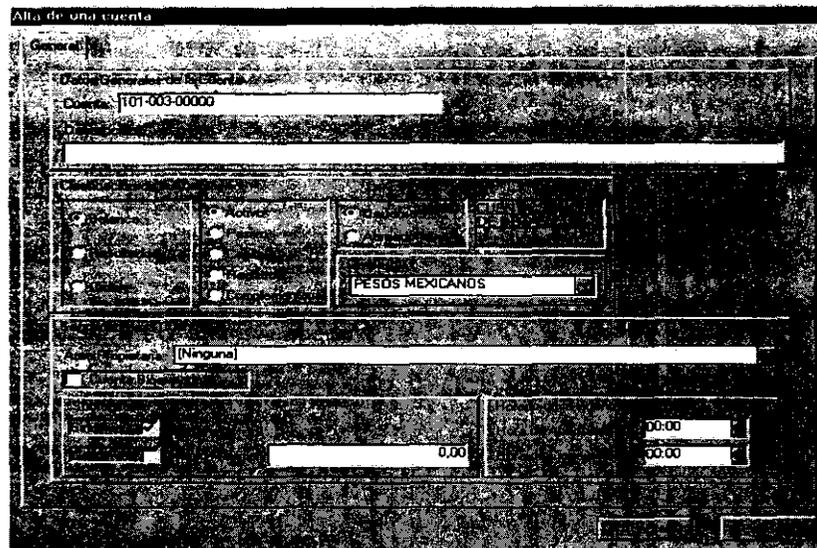
- Registro de una Cuenta

Pasos a seguir:

49. Haga doble *click* en Catálogos Contables. El sistema presentará automáticamente la ventana **Catálogos de Cuentas Contables**. Vea la siguiente figura:



50. Haga *click* sobre la cuenta general en la que desee incluir una nueva cuenta y oprima el botón derecho del Mouse seleccionando la opción Nueva o bien seleccione el botón Nuevo Nivel de la barra de herramientas y haga *click* con el botón izquierdo del Mouse o utilice el método abreviado *Ctrl + N*. El sistema presentará la siguiente ventana:



La ventana **Alta de una Cuenta** contiene los siguientes elementos:

Cuenta	Número de la cuenta que vamos a incluir.	Consulta
Descripción	Nombre completo de la cuenta que vamos a dar de alta.	Consulta
Balance	Nos permite seleccionar si la es propia para el Balance general de nuestra contabilidad.	Selección
Resultados	Nos permite seleccionar si la cuenta es propia para el Estado de Resultados de nuestra contabilidad.	Selección
Orden	Nos permite seleccionar si la cuenta es de Orden dentro de nuestra contabilidad.	Selección

<b>Activo</b>	Nos permite seleccionar si la cuenta pertenece al Activo dentro del Balance General, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de balance.	Selección.
<b>Pasivo</b>	Nos permite seleccionar si la cuenta pertenece al Pasivo dentro del Balance General, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de balance.	Selección.
<b>Capital</b>	Nos permite seleccionar si la cuenta pertenece al Capital dentro del Balance General, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de balance.	Selección.
<b>Registro</b>	Nos permite seleccionar si la cuenta pertenece a Registro, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de Orden.	Selección.
<b>Complementaria</b>	Nos permite seleccionar si la cuenta es complementaria, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de Orden.	Selección.
<b>Deudora</b>	En este campo seleccionamos si la cuenta a incluir en un nuevo nivel es de naturaleza deudora.	Selección.
<b>Acreedora</b>	En este campo seleccionamos si la cuenta a incluir en un nuevo nivel es de naturaleza acreedora.	Selección.
<b>Divisa</b>	Tenemos varios tipos de divisas para operar la cuenta, estos son: Pesos Mexicanos, Dolares, Udis, Libras Esterlinas, Pesetas.	Selección.
<b>Area Propietaria</b>	Muestra la descripción del Area Propietaria donde opera la cuenta.	Selección / Edición.
<b>Cuenta Bloqueada</b>	Este campo nos permite seleccionar si en algún momento queremos bloquear la nueva cuenta creada.	Selección.
<b>Sin Nivel</b>	En este campo seleccionamos si la cuenta nueva que creamos tiene un nivel para sobregiro en la misma, en caso de no tener se selecciona esta opción.	Selección.

**Contabilidad**

<b>Con Nivel</b>	En este campo seleccionamos si la cuenta nueva que creamos tiene un nivel para sobregiro en la misma, en caso de tenerlo se selecciona esta opción.	Selección.
<b>Monto Máximo</b>	En este campo tecleamos el monto máximo de sobregiro de la cuenta recién creada en caso de que se haya seleccionado que cuenta con un nivel de sobregiro.	Edición
<b>Horario de Apertura</b>	Como una medida de seguridad, el sistema provee de la posibilidad de tener un horario de operación para alguna cuenta en este campo habilitamos el horario de apertura de la misma.	Edición.
<b>Horario de Cierre</b>	Hora en que se cierra la operación de una cuenta.	Edición.

Además la ventana presenta los siguientes botones:

<b>Aceptar</b>	Confirma el alta de la nueva cuenta con todos los datos que han sido capturados.
<b>Cancelar</b>	Cancela la operación de dar de alta una nueva cuenta.

51. Introduzca el número de la nueva cuenta en el campo **Cuenta**. El número de cuenta es único por lo que debe de ser diferente a las ya existentes. En caso de introducir una cuenta ya existente el Sistema presentará el siguiente mensaje:



Oprima el botón **Aceptar** y teclee nuevamente el número de cuenta correctamente.

52. Introduzca el nombre completo de la cuenta en el campo **Descripción**.
53. Seleccione la clasificación de la nueva cuenta a incluir en el catálogo: **Balance, Resultados, Orden** si es de **Activo, Pasivo, Capital, Complementaria, Registro** si su naturaleza es: **Deudora o Acreedora**.
54. Seleccione el tipo de divisa con la que la cuenta va a operar en el campo **Divisa**.
55. Seleccione o introduzca el área propietaria de la cuenta en el campo **Area Propietaria**.
56. Haga *click* en el campo **Cuenta Bloqueada** en el caso de que así se requiera por seguridad o bien en caso contrario deje el cuadro en blanco.
57. Haga *click* en el campo **Sin Nivel** si la cuenta no tiene un nivel de sobregiro.
58. Haga *click* en el campo **Con Nivel** si la cuenta tiene un nivel de sobregiro.
59. Si su selección fue con nivel, introduzca el monto hasta donde se permite el sobregiro de la cuenta en el campo **Monto Máximo**.
60. En caso de requerir la operación de la cuenta en un horario establecido, introduzca la fecha de apertura de la cuenta en el campo **Horario de Apertura**, de igual forma introduzca el horario de cierre de la misma en el campo **Horario de Cierre**.
61. Guarde la información capturada.
62. En caso de que usted intente cancelar la operación de alta de la cuenta, el Sistema presentará el siguiente mensaje:



Si desea cancelar la edición de la cuenta haga clic en **Sí**, de lo contrario presione el botón **No** o bien el botón **Cancelar**.

---

- **Modificación de una Cuenta**

Pasos a seguir:

5. Haga doble *clic* en **Catálogos Contables**. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**.

- Haga *click* sobre la cuenta que desee modificar y oprima el botón derecho del mouse seleccionando la opción **Modificar** o bien seleccione el botón **Modificar** de la barra de herramientas y haga *click* con el botón izquierdo del mouse o utilice el método abreviado *Ctrl + M*. El sistema presentará la siguiente ventana:

The screenshot shows a window titled "Modificar una cuenta". It features a text input field at the top with the value "101-003-02001". Below this is a dropdown menu currently displaying "PESOS MEXICANOS". Underneath the dropdown is another dropdown menu showing "[Ninguna]". At the bottom of the window, there are two numeric input fields, both containing the value "0.00".

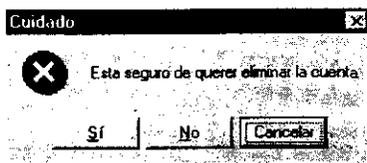
- Seleccione el campo que desee modificar.
- Introduzca el nuevo valor para ese campo.
- Guarde los cambios realizados.

#### • Eliminación de una Cuenta

##### Pasos a seguir:

- Haga doble *click* en **Catálogos Contables**. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**.
- Haga *click* sobre la cuenta que desee eliminar y oprima el botón derecho del mouse seleccionando la opción **Eliminar** o bien seleccione el botón **Eliminar** de

la barra de herramientas y haga *click* con el botón izquierdo del mouse. El sistema presentará la siguiente ventana:

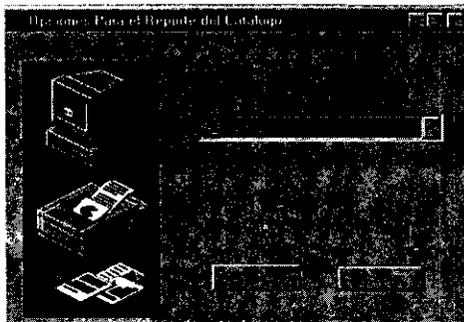


6. Haga *click* en **Sí** si está seguro de querer eliminar la cuenta seleccionada o en caso contrario seleccione la opción **No** o **Cancelar** para abortar el proceso.
7. Guarde los cambios realizados.

### •Elaboración del Reporte del Catálogo de Cuentas

Pasos a seguir:

1. Haga doble *click* en **Catálogos Contables**. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**.
2. Haga *click* con el botón derecho del mouse y seleccione la opción **Imprimir Catálogo**. El Sistema presentará la siguiente ventana:



3. Haga *click* en el campo **Nivel Contable a Reportar** y seleccione el mismo de la lista desplegable que aparece del lado derecho, las opciones a seleccionar son: **Mayor**, **Cuenta**, **Subcuenta**, **SubSubcuenta**.
4. Oprima el botón **Aceptar** y el Sistema le presentará la **Presentación Preliminar** del reporte a Imprimir. Vea la siguiente figura:

Programa ASA

**CATÁLOGO CONTABLE**  
Manzanillo. (PF). 1988

CUENTA	DESCRIPCIÓN	CLASIF
100-000-00000	ACTIVO CIRCULANTE	ACU
101-000-00000	DISPONIBLE	ACU
102-000-00000	INVERSIONES EN VALORES	ACU
103-000-00000	CUENTAS POR COBRAR	ACU
104-000-00000	OTRAS CUENTAS POR COBRAR	ACU
105-000-00000	ESTIMACION CTAS DE COBRO DUDOSO	ACU
106-000-00000	ANTICPOS	ACU
107-000-00000	IVA ACREDITABLE	ACU
108-000-00000	INVENTARIOS	ACU
109-000-00000	ESTIMACION COMBUSTIBLES POCO MOVIMIENTO	ACU
110-000-00000	ACTUALIZACION RESERVA COMB POCO MOV	ACU
111-000-00000	OTROS INVENTARIOS	ACU

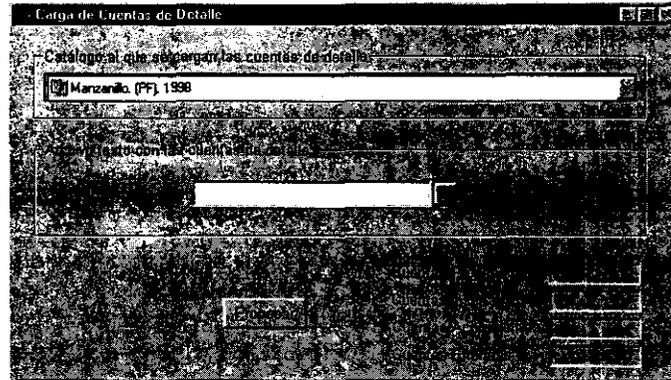
- Impresión del Reporte del Catálogo de Cuentas

Paso a seguir:

1. Seleccione en Archivo el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

### 3.2 Carga de Catálogos

La opción **Carga de Catálogos** nos permite introducir al Sistema cuentas de detalle que se encuentran guardadas en un archivo de texto con un formato previamente establecido. Vea la siguiente figura:



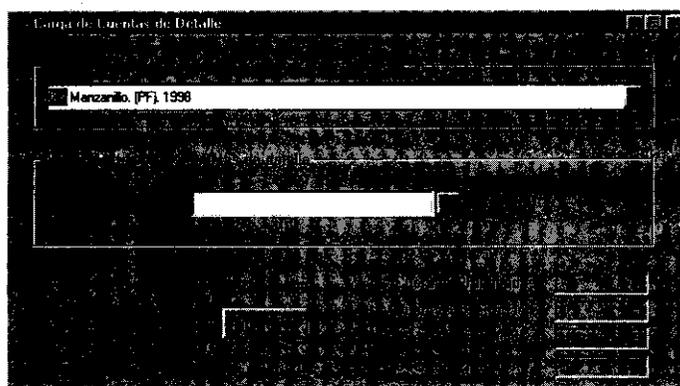
La ventana **Carga de Cuentas de Detalle** contiene los siguientes elementos:

<b>Catálogo al que se Cargan las Cuentas de Detalle</b>	Muestra el nombre del Catálogo al que se van a cargar las cuentas de detalle que van a ser importadas desde algún archivo de texto.	Selección
<b>Archivo Texto con las Cuentas de Detalle</b>	Muestra la ruta de donde vamos a importar el catálogo de cuentas de detalle.	Selección
<b>Procesar</b>	Realiza la operación del proceso del catálogo de cuentas.	Selección
<b>Cuentas Procesadas</b>	Muestra el total de cuentas procesadas.	Consulta
<b>Cuentas Insertadas</b>	Muestra el total de cuentas insertadas al catálogo actual.	Consulta
<b>Cuentas Acumulativas</b>	Muestra el total de cuentas acumulativas incluidas en el catálogo.	Consulta
<b>Cuentas Erróneas</b>	Muestra el número de cuentas erróneas encontradas en el proceso.	Consulta

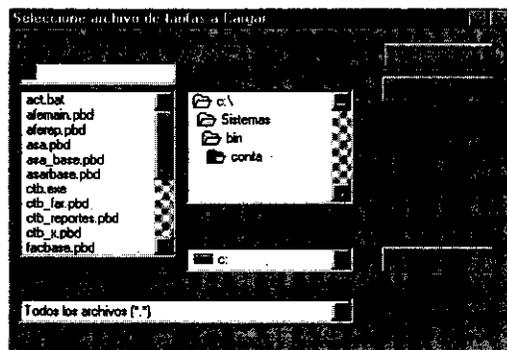
- Carga de un Catálogo

Pasos a seguir:

1. Haga doble *clic* en Carga de Catálogos. El sistema presentará automáticamente la ventana Carga de Cuentas de Detalle. Vea la siguiente figura:



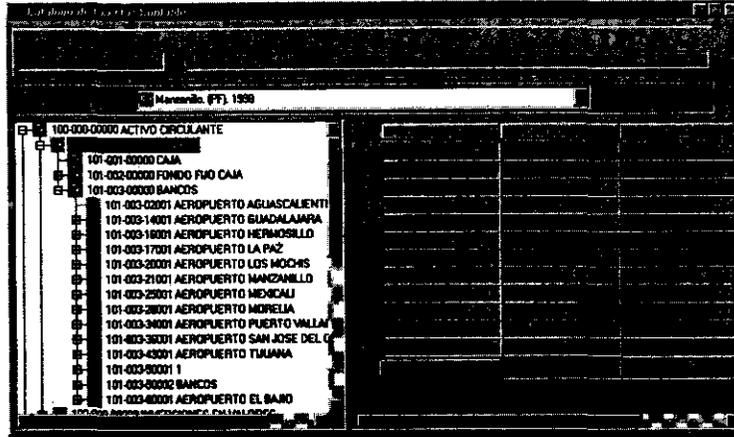
2. Haga *clic* en el campo Catálogo al que se cargan las cuentas de detalle y seleccione de la lista desplegable el nombre del catálogo a cargar.
3. Haga *clic* en el botón derecho del campo Archivo texto con las cuentas de detalle, el Sistema le presentará la siguiente ventana:



4. Escriba el nombre del archivo de texto de donde se van a importar las cuentas de detalle y seleccione la ruta de acceso al mismo, en caso de ser un archivo en un equipo que este dentro de la red haga *click* en el botón **Red** y seleccione la ruta deseada.
5. Haga *click* en el botón **Aceptar** en caso de haber introducido correctamente todos los datos o bien oprima el botón **Cancelar** para abortar el proceso.
6. Haga *click* en el botón **Procesar** y espere a que el proceso termine.

### 3.3 Consulta de Catálogos de Cuentas

Esta opción nos permite consultar los saldos de las cuentas de todos los catálogos que contiene la base de datos. Dentro de esta pantalla también tenemos la opción de generar varios de los reportes que requerimos para documentar nuestra contabilidad, como son: Balanza de Comprobación, Libro Diario, Libros Auxiliares, Pólizas Capturadas, Balance General y Estados de Resultados. Vea la siguiente figura:



La ventana Catálogo de Cuentas Contables contiene los siguientes elementos:

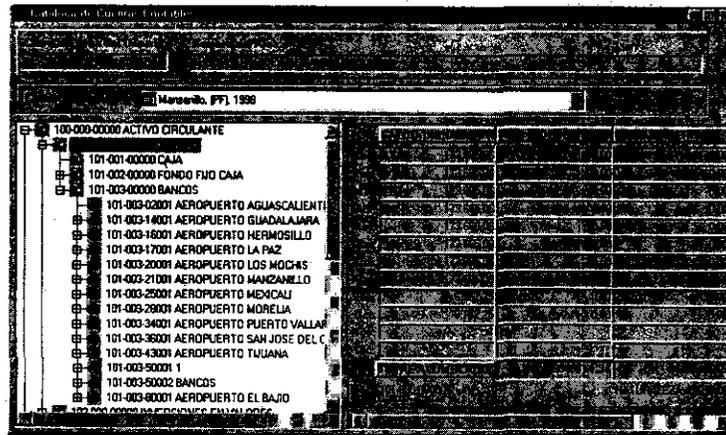
Clave del Area del Usuario	Identificador único del área de trabajo del usuario.	Consulta
Nombre del Area	Nombre completo del área de trabajo del usuario.	Consulta
Catálogo Actual	Lista desplegable con el nombre del catálogo con el cual estamos trabajando.	Edición / Selección
Columna sin nombre	Muestra las cuentas que contiene el catálogo actual.	Consulta / Selección
Saldo Inicial	Muestra el saldo inicial de la cuenta que seleccionamos.	Consulta
Debe	Muestra el total de movimientos de cargo de la cuenta que seleccionamos.	Consulta
Haber	Muestra el total de movimientos de abono de la cuenta que seleccionamos.	Consulta
Saldo Final	Muestra el saldo final de la cuenta que	Consulta

	seleccionamos.	
<b>Total Acumulado</b>	Muestra el total de movimientos de cargos y abonos al final del recuadro de las cuentas.	Consulta

• Consulta del Saldo de una Cuenta

Pasos a seguir:

1. Haga doble *click* en **Consulta de Catálogos de Cuentas**. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**. Vea la siguiente figura:

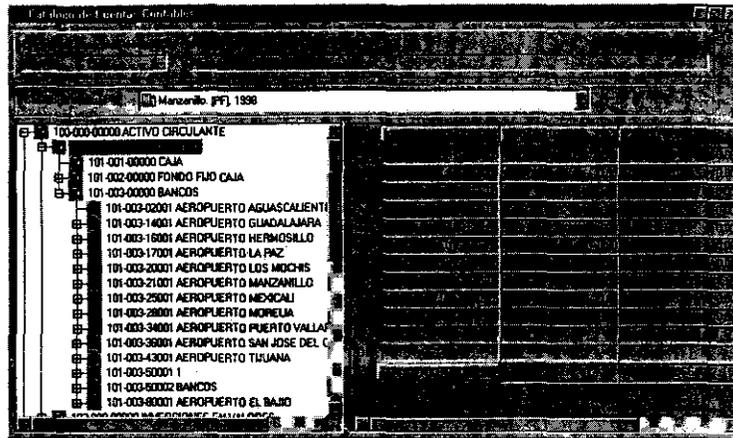


2. Haga *click* en el campo **Catálogo Actual** y seleccione de la lista desplegable que aparece del lado derecho el nombre del catálogo que contiene las cuentas que desee consultar.
3. Seleccione la cuenta que desee consultar y haga *click* sobre ella.
4. El sistema le mostrará el **Saldo Inicial**, los movimientos al **Debe y Haber**, el **Saldo Final** y el **Total Acumulado** anual de la cuenta que seleccionó.
5. Para observar todos los campos puede moverse con las barras de desplazamiento que aparecen en la parte de abajo de la ventana.

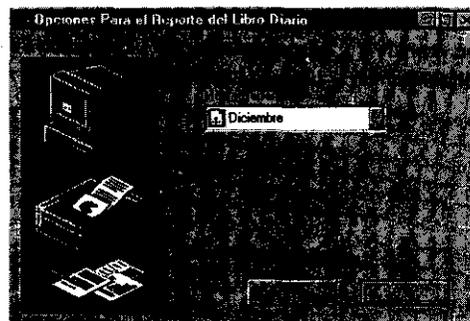
• Elaboración del Reporte del Libro Diario de un Catálogo

Pasos a seguir:

1. Haga doble clic en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana Catálogo de Cuentas Contables. Vea la siguiente figura:



2. Haga clic con el botón derecho del mouse y seleccione la opción Imprimir Libro Diario. El Sistema presentará la siguiente ventana:



3. Seleccione de la lista desplegable que aparece del lado derecho en el campo Mes a Reportar el mes a generar el Libro Diario.
4. Oprima el botón Aceptar y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

Sistema ASA

**SISTEMA DE CONTABILIDAD**  
LIBRO DIARIO

Al: Último día del mes de AGOSTO de 1998

Aeropuerto: HMO General Ignacio Pesqueira

Tipo de Póliza: DIARIO

Tipo de Póliza: DIARIO No. de Póliza: 1

Área Generadora: FI AEROPUERTO DE HERMOSILLO Creación: 11/06/1998

Concepto: NOMINA EMPL FUNC Y HON

Área	Cuenta	Descripción de la Cuenta	Cheque	Concepto del Movimiento	
80	808-001-00101	SUELDO	8000000	NOMINA EMPL FUNC Y HON	283,9
80	808-001-00110	ESTIMULO POR EFICIENCIA	0000000	NOMINA EMPL FUNC Y HON	29,4
80	808-001-00102	COMPENSACION ASA	0000000	NOMINA EMPL FUNC Y HON	2
80	808-001-00103	PRIMA QUINCENAL	0000000	NOMINA EMPL FUNC Y HON	5,5
80	808-001-00105	TIEMPO EXTRA ORDINARIO	0000000	NOMINA EMPL FUNC Y HON	32,3
80	808-001-00109	30% PRIMA DOMINICAL	0000000	NOMINA EMPL FUNC Y HON	2,6
80	808-001-00101	SUELDO	0000000	NOMINA EMPL FUNC Y HON	(4,8)

- Impresión del Libro Diario de un Catálogo

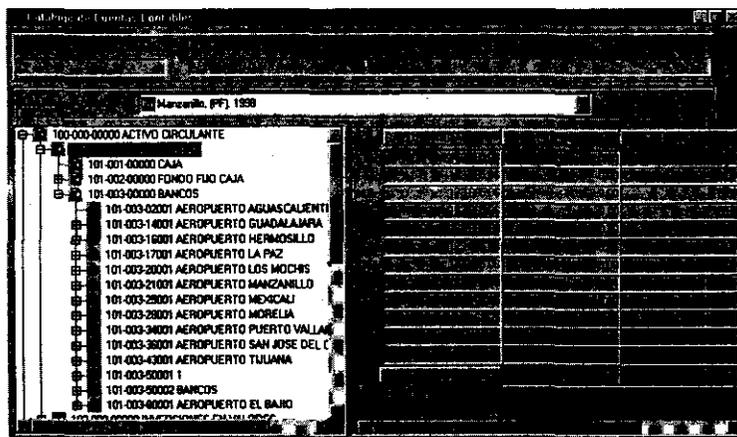
Pasos a seguir:

1. Seleccione en **A**rchivo el comando **I**mprimir... o **I**mpresión **D**irecta u oprima el botón **I**mprimir o **I**mpresión **D**irecta.

- Elaboración del Reporte de Balanza de Comprobación

Pasos a seguir:

1. Haga doble *clic* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana Catálogo de Cuentas Contables. Vea la siguiente figura:



2. Haga *clic* con el botón derecho del Mouse en el área blanca del lado izquierdo de la ventana y seleccione, **Imprimir Balanza**. El Sistema presentará la siguiente ventana:



3. Haga *click* en el campo Nivel Contable a Reportar y seleccione el mismo de la lista desplegable que aparece del lado derecho, las opciones a seleccionar son: **Mayor, Cuenta, Subcuenta, SubSubcuenta.**
4. Seleccione de la lista desplegable que aparece del lado derecho en el siguiente campo el mes a generar la balanza.
5. Oprima el botón Aceptar y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

Área Cuenta	Descripción de la Cuenta	Saldo Anterior		Movimientos		
		Deudor	Acreedor	Debe	Haber	
21 100-000-00000	ACTIVO CIRCULANTE	23.983,21		0,00	0,00	23.983,21
21 101-000-00000	DISPONIBLE	6.182,25		0,00	0,00	6.182,25
21 104-000-00000	OTRAS CUENTAS POR COBRAR	(1.089,14)		0,00	0,00	(1.089,14)
21 106-000-00000	ANTICIPOS	(7.825,86)		0,00	0,00	(7.825,86)
21 107-000-00000	IVA ACREDITABLE	0,00		0,00	0,00	0,00
21 111-000-00000	OTROS INVENTARIOS	28.315,96		0,00	0,00	28.315,96
21 130-000-00000	ACTIVO FIJO	42.350.555,97		0,00	0,00	42.350.555,97
21 131-000-00000	BIENES INMUEBLES HISTORICO	8.898.803,00		0,00	0,00	8.898.803,00
21 132-000-00000	BIENES INMUEBLES REEXPRESION	141.483.220,00		0,00	0,00	141.483.220,00
21 133-000-00000	BIENES MUEBLES HISTORICO	3.480.385,00		0,00	0,00	3.480.385,00
21 134-000-00000	BIENES MUEBLES REEXPRESION	14.708.551,00		0,00	0,00	14.708.551,00

- Impresión de la Balanza de un Catálogo

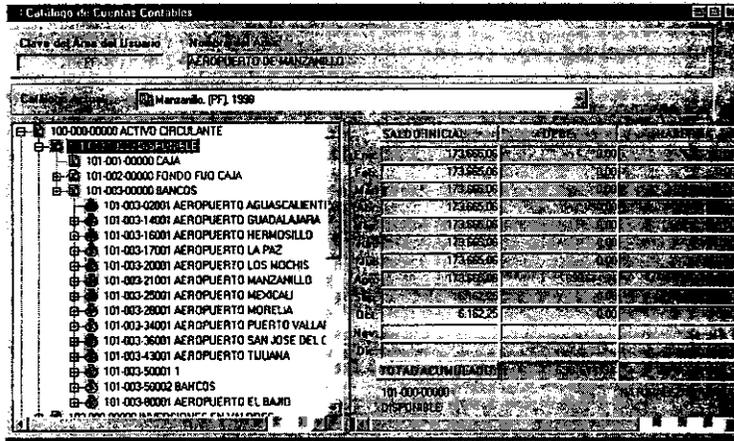
Pasos a seguir:

1. Seleccione en **Archivo** el comando *Imprimir...* o *Impresión Directa* u oprima el botón Imprimir o Impresión Directa.

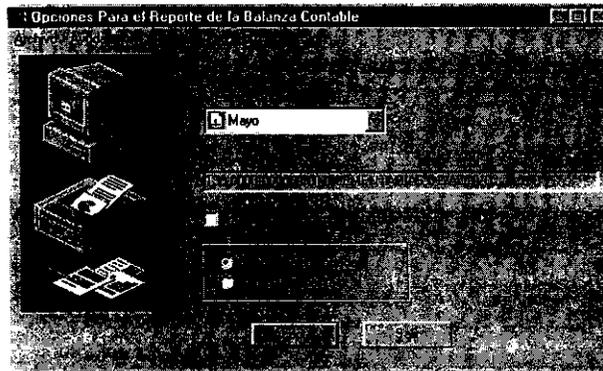
- Elaboración del Reporte de Libros Auxiliares

Pasos a seguir:

1. Haga doble *clic* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana Catálogo de Cuentas Contables. Vea la siguiente figura:



2. Haga *clic* con el botón derecho del mouse y seleccione la opción **Imprimir Libros Auxiliares**. El Sistema presentará la siguiente ventana:



3. Haga *click* en el campo Nivel Contable a Reportar y seleccione el mismo de la lista desplegable que aparece del lado derecho, las opciones a seleccionar son: Mayor, Cuenta, Subcuenta, SubSubcuenta.
4. Seleccione de la lista desplegable que aparece del lado derecho en el siguiente campo el mes a generar el reporte.
5. Oprima el botón *Aceptar* y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

Sistema ASA

**SISTEMA DE CONTABILIDAD**  
LIBROS AUXILIARES  
AL Último día del mes de MAYO de 1999

Aeropuerto: CUN      Internacional De Cancun  
Código: 69              CUN Aeropuerto de Cancún S.A. de C.V. (69) 1000

Cuenta	Descripción de la Cuenta	Saldo Anterior		Movimientos	
		Deudor	Acreedor	Debe	Haber
180-0000-00000	COMPLEMENTARIAS DE ACTIVO		27,018,363.16	0.00	0.00
191-0000-00000	DEPRECIACIÓN ACUMULADA		466,584.70	0.00	0.00
191-0001-00000	MADINARIA Y EQUIPO DE OPERA		67,789.45	0.00	0.00
191-0001-00002	ADMINISTRACION		67,789.45	0.00	0.00
191-0002-00000	MOBILIARIO Y EQUIPO DE OFICINA		29,727.25	0.00	0.00
191-0002-00002	ADMINISTRACION		29,727.25	0.00	0.00
191-0003-00000	EQUIPO DE TRANSPORTE		387,785.25	0.00	0.00

- Impresión del Libros Auxiliares

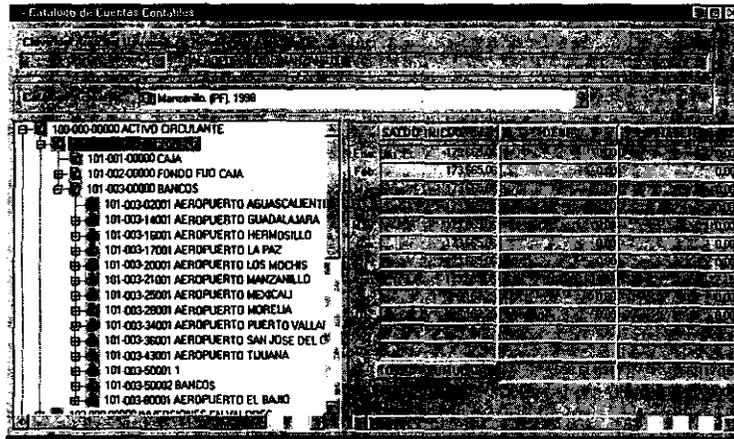
Pasos a seguir:

1. Seleccione en Archivo el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

- Elaboración del Reporte de Pólizas Capturadas

Pasos a seguir:

1. Haga doble *click* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana Catálogo de Cuentas Contables. Vea la siguiente figura:



2. Haga *click* con el botón derecho del mouse y seleccione la opción Imprimir Pólizas Capturadas. El Sistema presentará la siguiente ventana:



3. Seleccione de la lista desplegable que aparece del lado derecho en el campo **Mes a Reportar** el mes a generar el reporte.
4. Oprima el botón **Aceptar** y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

**SISTEMA DE CONTABILIDAD**  
Pólizas Capturadas  
Al: Último día del mes de AGOSTO de 1998

Aeropuerto: NMO      General Ignacio Pasquel

Tipo de Póliza: DIARIO

Tipo de Póliza: DIARIO      No. de Póliza: 1

Área Operadora: PI      AEROPUERTO DE HERMOSILLO

Concepto: NOMINA EMPL FUNC Y NON

Creación: 11/08/1998

Área	Cuenta	Descripción de la Cuenta	Cheque	Concepto del Movimiento	
08	800-001-00101	SUELDO	0000000	NOMINA EMPL FUNC Y NON	203,0
08	800-001-00118	ESTIMULO POR EFICIENCIA	0000000	NOMINA EMPL FUNC Y NON	29,4
08	800-001-00102	COMPENSACION ASA	0000000	NOMINA EMPL FUNC Y NON	2
08	800-001-00103	PRIMA QUINGUENAL	0000000	NOMINA EMPL FUNC Y NON	5,5
08	800-001-00105	TIEMPO EXTRA ORDINARIO	0000000	NOMINA EMPL FUNC Y NON	32,3
08	800-001-00106	30% PRIMA DOMINICAL	0000000	NOMINA EMPL FUNC Y NON	2,8
08	800-001-00101	SUELDO	0000000	NOMINA EMPL FUNC Y NON	(4,0)

- Impresión del Reporte de Pólizas Capturadas

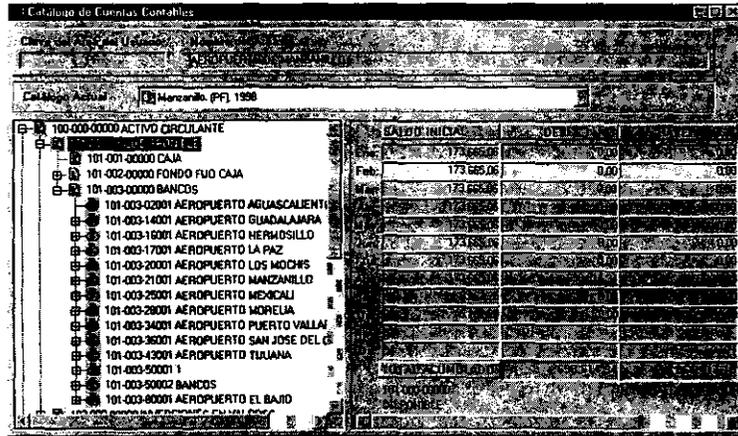
Pasos a seguir:

1. Seleccione en **Archivo** el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

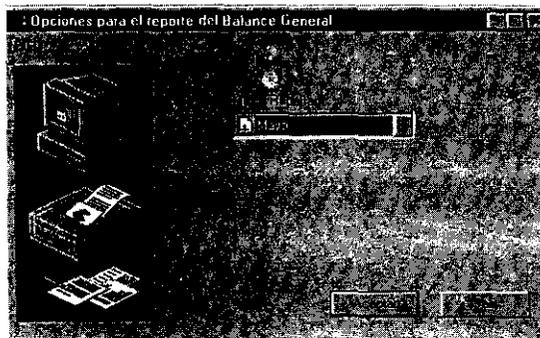
- Elaboración del Reporte de Balance General

Pasos a seguir:

1. Haga doble *click* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**. Vea la siguiente figura:



2. Haga *click* con el botón derecho del mouse y seleccione la opción **Imprimir Balance General**. El Sistema presentará la siguiente ventana:



3. Seleccione de la lista desplegable que aparece del lado derecho en el campo **Mes a Reportar** el mes a generar el reporte.

4. Oprima el botón Aceptar y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

Activo		
<i>Circulante</i>		
Fondos Fijos	10,412.39	0.00
Cuentas Bancarias	41,939,400.09	0.83
Inversiones en Valores	63,550,191.05	1.26
Cuentas	47,358,020.88	0.94
Reserva P/Ctas Incobrables	(525,436.80)	-0.01
Otras Cuentas por Cobrar	4,088,877.22	0.08
Fideicomisos con Saldo Deudor	10,432,935.57	0.21
Impuestos a Favor	8,352,415.36	0.17
Anticipos a Contratistas y Proveedores	9,467,604.66	0.19
	<b>194,676,420.29</b>	<b>3.87</b>
<i>Fijo</i>		

- Impresión del Reporte Balance General

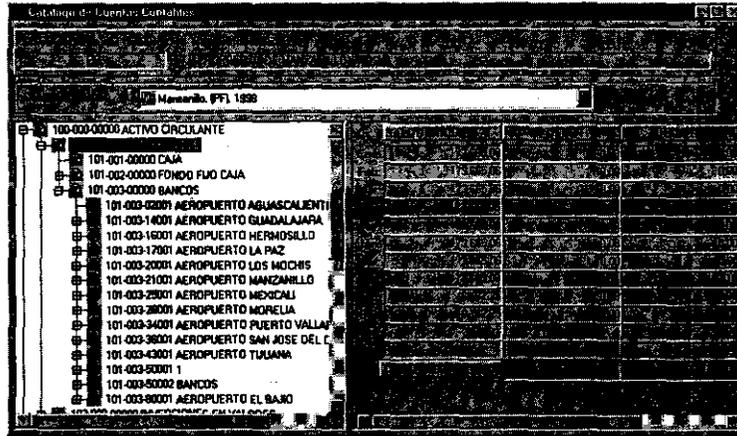
Pasos a seguir:

1. Seleccione en **A**rchivo el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

- Elaboración del Reporte de Estado de Resultados

Pasos a seguir:

1. Haga doble *clic* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**. Vea la siguiente figura:



2. Haga *clic* con el botón derecho del mouse y seleccione la opción **Imprimir Estado de Resultados**. El Sistema presentará la siguiente ventana:



3. Seleccione de la lista desplegable que aparece del lado derecho en el campo Mes a Reportar el mes a generar el reporte.
4. Oprima el botón Aceptar y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

SISTEMA DE CONTABILIDAD		
ESTADO DE RESULTADOS		
AL Último día del mes de MAYO de 1999		
Aeropuerto: Cancun S.A. de C.V.		
<b>Ingresos de Operación</b>		
Servicios Aeroportuarios	39,133,879.84	30.87 %
Tarifa de uso de Aeropuerto Nacional	14,718,037.58	11.54 %
Tarifa de uso de Aeropuerto Internacional	66,180,506.58	51.87 %
Servicios Comerciales	7,554,315.23	5.92 %
	<b>127,586,739.23</b>	<b>100.00 %</b>
<b>Gastos de Operación</b>		
Sueldos	3,209,008.94	2.52 %
Cuotas IMSS	503,829.50	0.39 %
Cuotas INFONAVIT	83,366.10	0.07 %

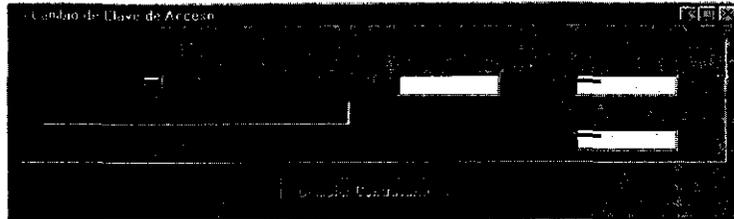
- Impresión del Reporte de Estado de Resultados

Pasos a seguir:

1. Seleccione en **Archivo** el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

## 4.1 Cambio de Clave de Acceso

Una medida de seguridad en todo sistema y que puede ser utilizada por diferentes usuarios, es la de cambiar continuamente la clave de acceso o contraseña al mismo, para esto el sistema cuenta con la ventana **Cambio de Clave de Acceso**. Vea la siguiente figura:



La ventana **Cambio de Clave de Acceso** contiene los siguientes elementos:

<b>Usuario</b>	Muestra la clave de identificación única y el nombre completo del usuario.	Consulta
<b>Contraseña Anterior</b>	En este campo se registra la contraseña de identificación actual.	Edición
<b>Nueva Contraseña</b>	Permite introducir la nueva contraseña de identificación.	Edición
<b>Confirmación</b>	Permite confirmar la nueva contraseña de identificación.	Edición

Además, la ventana anterior presenta el siguiente botón:

<b>Cambiar Contraseña</b>	Activa la nueva contraseña del usuario.
---------------------------	---

### • Cambio de Clave de Acceso

#### Pasos a seguir:

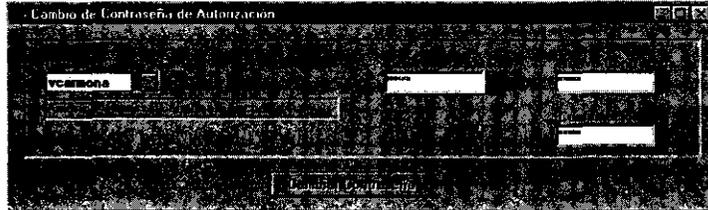
1. Haga doble *click* en **Cambio de Clave de Acceso**. Esta acción le presentará la ventana con el mismo nombre.
2. Escriba en el campo **Contraseña Anterior**, la contraseña de identificación actual y presione *Tab*.

3. Escriba en el campo Nueva Contraseña, la nueva contraseña de identificación y presione *Tab*. La contraseña puede contener hasta 21 caracteres (letras y números) como máximo y 6 caracteres como mínimo, el sistema presenta una mascara de asteriscos como medida de seguridad para que no pueda ser visualizada por otros usuarios.
4. Escriba en el campo Confirmación, nuevamente la contraseña de identificación y presione *Tab*.. Al ejecutar este paso se activa el botón **Cambiar Contraseña**. Nuevamente este campo aparece con una mascara de asteriscos como protección
5. Oprima el botón **Cambiar Contraseña**. Si la nueva contraseña ha sido registrada con éxito en la base de datos del sistema, se presentará el siguiente mensaje:



## 4.2 Cambio de Contraseña de Autorización

Dentro del Sistema de Contabilidad existen ciertos procesos especiales para los que se requiere la autorización de una persona con la responsabilidad sobre las mismas para que estas se lleven a cabo. Para ello, requiere proporcionar al usuario la clave y su contraseña, esta es distinta a la de acceso. Vea la siguiente figura:



La ventana **Cambio de Contraseña de Autorización** contiene los siguientes elementos:

<b>Usuario</b>	Muestra la clave de identificación única y el nombre completo del usuario.	Consulta
<b>Contraseña Anterior</b>	En este campo se registra la contraseña de identificación actual.	Edición
<b>Nueva Contraseña</b>	Permite introducir la nueva contraseña.	Edición
<b>Confirmación</b>	Permite confirmar la nueva contraseña de identificación.	Edición

Además, la ventana anterior presenta el siguiente botón:

<b>Cambiar Contraseña</b>	Activa la nueva contraseña del usuario.
---------------------------	---

- Cambio de Clave de Contraseña de Autorización

Pasos a seguir:

1. Haga doble *clic* en **Cambio de Contraseña de Autorización**. Esta acción le presentará la ventana con el mismo nombre.
2. Escriba en el campo **Contraseña Anterior**, la contraseña de identificación actual y presione *Tab.*, el sistema utiliza una mascara de asteriscos para que la contraseña no pueda ser visualizada.
3. Escriba en el campo **Nueva Contraseña**, la nueva contraseña de identificación y presione *Tab.* La contraseña puede contener hasta 21 caracteres (letras y números) como máximo y 6 caracteres como mínimo, se utiliza tambien una mascara de asterisco como protección.
4. Escriba en el campo **Confirmación**, nuevamente la contraseña de identificación y presione *Tab.*, encontramos nuevamente una mascara de asteriscos como medida de seguridad.
5. Oprima el botón **Cambiar Contraseña**. Si la nueva contraseña ha sido registrada con éxito en la base de datos del sistema, se presentará el siguiente mensaje:



### 4.3 Mensajes de Error en la Ventana Cambio de Clave de Acceso

- a. Se presenta cuando la contraseña introducida no coincide con la registrada en el sistema.



- b. Aparece cuando introduce la contraseña con menos de seis caracteres de longitud.



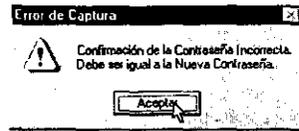
- c. Se presenta cuando la nueva contraseña es igual a la anterior.



- d. Aparece cuando la contraseña tiene menos de seis caracteres de longitud.



e. Se presenta cuando la confirmación no es idéntica a la contraseña introducida.



f. Se muestra cuando la contraseña tiene menos de seis caracteres de longitud.



# APÉNDICE C

## MANUAL TÉCNICO



# 1. GUIA DEL ADMINISTRADOR

El objetivo de la presente guía es conformar un documento en el que el Administrador del Sistema pueda referirse para mantener el Sistema en óptimas condiciones, y en alguna contingencia poder restablecerlo para continuar con la operación del mismo.

## 1.1. REQUERIMIENTOS MINIMOS DEL SISTEMA EN EL SERVIDOR

Es necesario que el Administrador sepa los requerimientos mínimos con los que debe de contar el Sistema para funcionar correctamente. Tales requerimientos pueden ser de 3 tipos: de Software, Hardware y de ambiente. Esto es, todos aquellos aspectos que son requeridos para la puesta en marcha y funcionamiento de un servidor, considerando tanto el aspecto hardware como el aspecto software, que son suficientes y necesarios para soportar la operación del Sistema de Contabilidad.

### 1.1.1. HARDWARE

En cuanto a Hardware se refiere, los requerimientos mínimos fueron establecidos para soportar el funcionamiento de la base de datos y el número de usuarios requeridos por el sistema, los cuales se listan a continuación:

Servidor con procesador dual Intel Pentium Pro a 200 Mhz  
128 Mb de memoria RAM  
Memoria caché de 512 Kb  
Tarjeta de Red Fast Ethernet (de preferencia conectada a un switch)  
2 Disco Duros SCSI, el primero de 2 Gb que alberga al sistema operativo y el área de memoria virtual, y el segundo de 9 Gb que albergará las bases de datos, archivos de

respaldo, etc.

Monitor SVGA y teclado 102 teclas adaptado a WIN 95

Mouse

Unidad de Floppy Disk 3 ½

Unidad de CD ROM

Unidad de Cinta con capacidad para respaldar 5 Gb

### 1.1.2. SOFTWARE

Con lo que respecta al software, se contemplan los siguientes tipos de programa. de Sistema Operativo, para el Sistema Administrador de Bases de Datos ( Database Managment System o DBMS por sus siglas en inglés) y los de Replicación de la Base de Datos, los cuales se listan a continuación:

Sistema Operativo NT v4.0 con Service Pack 3 o mayor (el service pack 3 es indispensable)

SQL Server Professional v11.03 o mayor para Windows NT ó

Adaptive Server Enterprise v11.5 para Windows NT

Replication Server v11.03 para SQL Server o 11.5 para el Adaptive Server.

### 1.1.3. AMBIENTE

Por ambiente se debe entender a los usuarios, tanto de Windows NT como de SQL/Adaptive Server y las condiciones que deben de cumplir dentro del Sistema de Contabilidad para operarlo.

### 1.1.3.1. WINDOWS NT

Para la parte de Windows NT se necesitan las siguientes cuentas:

Admssql	Cuenta para administrar los procesos relacionados con el DBMS, este usuario debe tener privilegios de sa, pero debe ser creada como una cuenta alterna de sa.
Respaldo	Este usuario es el encargado de realizar los respaldos de las Bases de Datos contenidas en el Servidor

### 1.1.3.2. SQL/ADAPTIVE SERVER

Para el DBMS se requieren las siguientes cuentas:

Asa	Cuenta del administrador del Sistema de Contabilidad
Useradmin	Cuenta para la ejecución de procesos especiales entre el Sistema de Contabilidad y el DBMS.
Respaldo	Cuenta para conectarse al DBMS, para realizar los respaldos de la Base de Datos.

### 1.1.3.3. SISTEMA DE CONTABILIDAD

#### CUENTAS

Para poner en marcha el Sistema de Contabilidad, es necesario tener creados las cuentas 'asa' y 'useradmin'. El usuario asa debe tener permiso para administrar el sistema (y todos los permisos del menú de Seguridad), el cual se determina en la tabla de sys\_usuarios en el campo admin\_si con un valor de 1.

Con lo que respecta a la cuenta useradmin este debe tener el privilegio de 'sso' (System Security Officer) como rol principal dentro

de la administración del DBMS para poder realizar su labor de interacción entre el DBMS y el Sistema de Contabilidad; el único inconveniente, si se le da solo el rol anterior, es que en SQL Server es necesario modificar el procedimiento `sp_droplogin` para que se puedan dar de baja usuarios desde el Sistema de Contabilidad, de lo contrario la cuenta `useradmin` debe tener privilegios de `'sa'`.

#### TABLAS

Para poder poner en marcha el Sistema de Contabilidad, es necesario tener una serie de datos en algunas tablas, dichos datos pueden ser copiados de tablas en Bases de Datos existentes o en su defecto, generar los datos manualmente. A continuación se listan las tablas que requieren al menos de un registro:

TABLA	INFORMACION
Asa_c_area	Catálogo de áreas Asa_c_area: *SISASA* Desc_ar_str: Area destinada al Sistema de Contabilidad
Sys_usuarios	Catálogo de usuarios Sys_usuarios: asa Nombre_str Administrador del Sistema de Contabilidad Admin_si: 1
Sys_subsistemas	Catálogo de Subsistemas Sys_subsistema: SEG Nom_subsistema_str Sistema de Contabilidad

## 1.2. REQUERIMIENTOS MINIMOS DEL SISTEMA EN EL CLIENTE

### 1.2.1. HARDWARE

Con lo que respecta al Hardware, los requerimientos mínimos soportan la operación sin tener problemas de espacio y de lentitud. Dichos requerimientos se listan a continuación:

Computadora con procesador Intel Pentium 166 Mhz  
16 Mb de memoria RAM (recomendable 32)  
Memoria caché de 512 Kb  
Tarjeta de Red Ethernet  
Disco Duro de 2 Gb  
Monitor SVGA y teclado 102 teclas adaptado a WIN 95  
Mouse  
Unidad de Floppy Disk 5.25  
Unidad de CD ROM

### 1.2.2. SOFTWARE

El software que necesita el cliente para operar el Sistema de Contabilidad es el que se lista a continuación:

Windows 95 o más reciente  
Powersoft Deployment Kit para PowerBuilder versión 5  
Adaptive/SQL Server Cliente version 10.0.4 o mayor  
Visual Components runtime oxc's versión

### 1.3.INSTALACION DEL SISTEMA

Antes de iniciar la instalación del Sistema de Contabilidad es recomendable que verifique que todos los parámetros de la configuración de su nodo en la red sean correctos, debe usted verificar que la dirección IP que tiene su computadora este asignada correctamente, debe verificar el buen funcionamiento de su tarjeta de red y que las direcciones IP hacia las bases de datos sean correctas. Todos estos puntos los realiza el personal de Soporte Técnico, así que puede solicitar apoyo a alguna persona de ese departamento en caso de ser necesario.

Para iniciar y únicamente en el caso de que su computadora tenga WINDOWS NT ( no es aplicable en caso de WINDOWS 95) como plataforma de trabajo proceda a la creación de una carpeta nueva dentro del directorio raíz C:\ cuyo nombre es **Windows** y dentro de esta carpeta una más cuyo nombre debe ser **temp**.

La instalación del Sistema de Contabilidad requiere del acceso a otra área dentro de la red que contiene el programa pertinente para su configuración. Este equipo lleva por nombre "srvinf01" lo puede buscar de la siguiente manera, observe la figura 1.3.1.

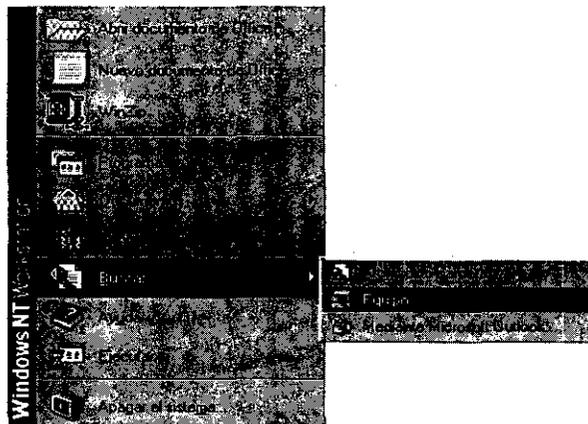


fig. 1.3.1.

Dentro del menú de inicio en Windows tenemos la opción "Buscar" presione el botón izquierdo del Mouse y el sistema presentará la siguiente pantalla:

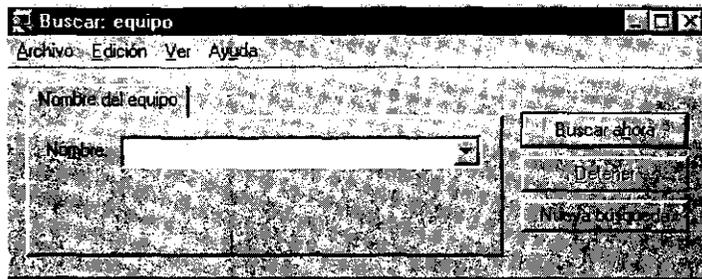


fig. 1.3.2.

Teclee en el campo nombre `srvinf01` presione el botón buscar ahora, el sistema presentará la siguiente pantalla:

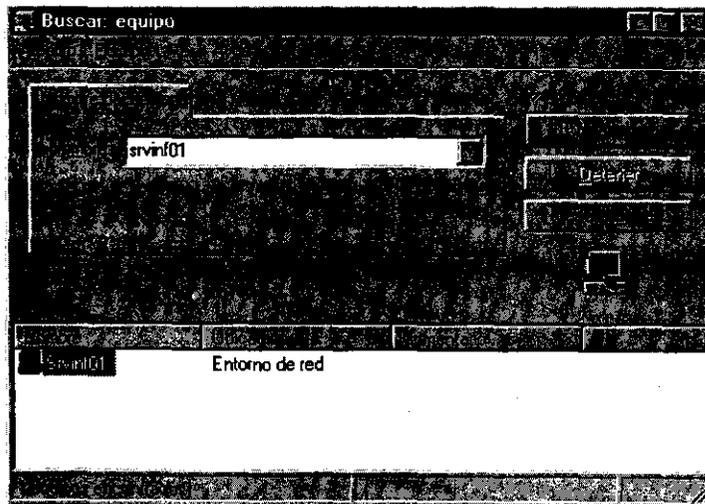
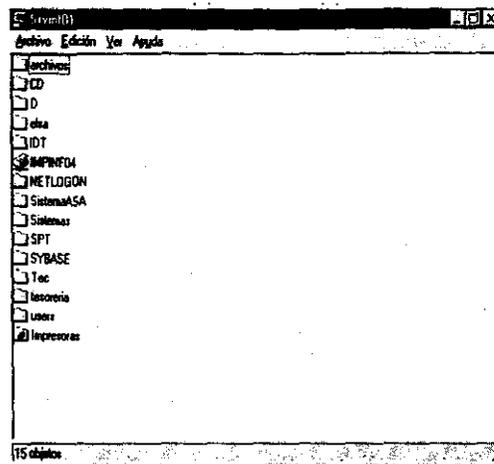


fig. 1.3.3.

Presione dos veces el botón izquierdo del mouse en **srvinf01** el sistema presentará la siguiente ventana:



**fig. 1.3.4.**

Haga *doble clic* en la carpeta **Sistemas** y busque el archivo **Setup** ejecute el programa de instalación y acepte todos los parámetros predeterminados que la instalación proporciona.

El sistema le preguntará si desea reiniciar en ese momento su equipo para empezar a trabajar con el Sistema de Contabilidad presione el botón que indica que **no** ya que es necesario hacer algunos cambios más para iniciar la sesión.

Los cambios que se requieren antes de reiniciar su equipo son:

- 1.- Corregir el archivo **autoexec.bat** que se encuentra en su computadora para que quede de la siguiente manera para el caso de que usted tenga Windows NT

```
SETPATH=%PATH%;c:\Sistemas\archivos;c:\Sistemas\Bin;c:\Sistemas\archivos\i  
conos;c:\Sybase\bin;C:\Archivos de programa\Archivos comunes\Powersoft  
Shared  
SET SYBASE=C:\SYBASE  
SET DSQUERY=SYBASE  
SET PATH=C:\SYBASE\BIN;C:\SYBASE\DLL;%PATH%  
SET LIB=C:\SYBASE\LIB;%LIB%  
SET INCLUDE=C:\SYBASE\INCLUDE;%INCLUDE%
```

En caso de que su máquina tenga Windows 95, el archivo deberá tener la siguiente configuración:

```
SETPATH=%PATH%;c:\Sistemas\archivos;c:\Sistemas\Bin;c:\Sistemas\archivos\i  
conos;c:\Sybase\bin;"C:\Archivos de programa\Archivos comunes\Powersoft  
Shared"  
CALL WSYBSET
```

2.- Modificar el archivo config.sys de la siguiente manera:

```
BUFFERS=25  
FILES=30
```

3.- Dentro de la carpeta **Sistemas** que está ubicado en el directorio raíz C:\ existe un archivo a cambiar:

Archivo ASA.INI el cual deberá de contener la dirección a las bases de datos a las que usted tendrá acceso, en esta parte, usted tendrá apoyo de personal especializado el cual deberá saber, hacia donde deberá conectarlo.

4.- Dentro de la carpeta **Sybase** que está ubicado en el directorio raíz C:\ se encuentra el subdirectorio **Ini** en donde existe otro a cambiar:

Archivo **SQL.INI** en el subdirectorio **Sybase/ini**, el cual deberá contener la dirección **IP** a los servidores que se requiera accesar, en este parte, usted tendrá apoyo de personal especializado el cual deberá saber, hacia donde deberá conectarlo.

Después de realizar estos cambios, reinicie su computadora y el Sistema de Contabilidad procederá a cargar los archivos desde la base de datos a la que está dirigido, espere de 15 a 20 segundos y después el sistema iniciará su operación.

Cabe mencionar, que el nombre de usuario y la clave de acceso al Sistema son proporcionados por el administrador del mismo, por lo que requiere solicitar su registro y accesos.

## 2.PROCEDIMIENTO PARA LA INSTALACIÓN DE LA BASE DE DATOS

### 2.1.INSTALACIÓN DE SERVIDORES DE BASES DE DATOS ADAPTIVE/SQL SERVER

El presente documento considera la instalación tanto del Adaptive Server como del SQL Server, por lo cual nos referiremos a cualquiera de los 2 como el Servidor de Bases de Datos. De tal manera que es necesario realizar las siguientes acciones:

NOTA 1: La versión del Adaptive Server y del SQL Server que se manejan en el presente documento son las descritas en el punto 1.1.2 Software (del punto 1.1 Requerimientos Mínimos en el Servidor).

NOTA 2: Para instalar los productos Sybase (Adaptive/SQL Server), se requiere conectarse con un usuario que tenga privilegios de Administrador.

- Inserte el CD-ROM e inicie el *programa instalador* de la siguiente forma:

Desde la opción **ejecutar** del menú inicio de Windows NT, ejecute:

*n:\setup.exe*

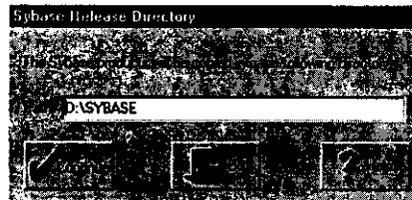
donde *n*: es el nombre del dispositivo lector de CD-ROM

Cuando el instalador se inicie, podrá tener acceso a la ayuda en línea en cada pantalla.

Una vez que aparece la primera pantalla interactiva de instalación, es necesario especificar la ruta del Servidor. Debe tomarse en cuenta que la ruta debe ser en una unidad de discos en

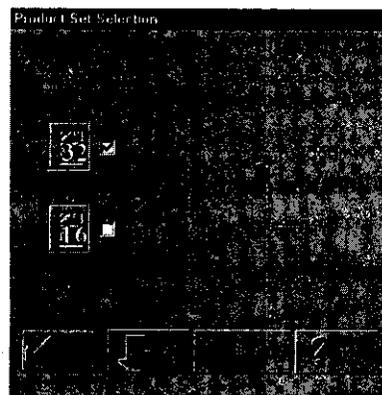
la que haya suficiente espacio (por lo menos 300 MB), que como estándar es el disco *D*.

Especifique un directorio local en el cual instalar los productos Sybase:



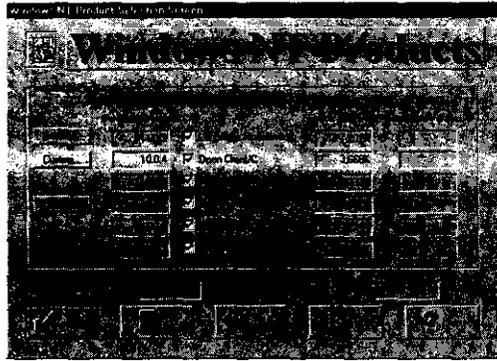
*fig. 2.1.1*

Instale SQL Server en un sistema de archivos NTFS en lugar de un sistema FAT a fin de mantener la integridad de la Base de Datos y la coherencia del caché en caso de una caída de la Base de Datos.



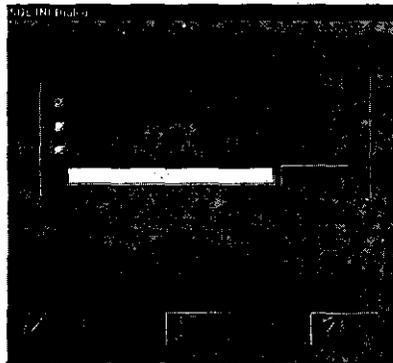
*fig. 2.1.2*

Seleccione el tipo de producto que quiere instalar (*fig. 2.1.3*). En algunas opciones aparece un botón **Custom...** del lado izquierdo; esto indica que dicha opción tiene forma de personalizarse.



**fig. 2.1.3**

Cuando el instalador solicite especificar la fuente del archivo sql.ini, seleccione **Ignore for Now** o asegúrese de que esta opción esté elegida (**fig. 2.1.4**), posteriormente se configurará este archivo. Dé clic en **Continue**.



**fig. 2.1.4**

En seguida, en la opción **Languages Modules** se deben seleccionar dos idiomas: **English** y **Spanish**. (**fig. 2.1.5**).

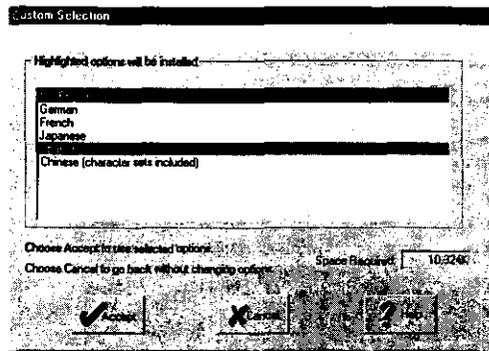


fig. 2.1.5

Después, en la opción **Open Client** solamente se seleccionan 2 opciones, de acuerdo a la **fig. 2.1.6**:

Network library for name pipes

Network library for Window Sockets TCP/IP

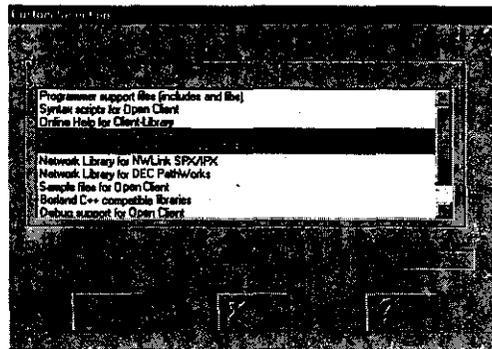


fig. 2.1.6

A continuación, en la ventana de Selección de producto de Windows NT que aparece (**fig. 2.1.3**), se selecciona **Historical Server**, haciendo *clic* en la casilla de selección. Esta opción habilitada permitirá que se guarde información de las transacciones hechas a la Base de Datos, para fines de auditoría.

A continuación se seleccionan **Monitor Client** (sin abrir el botón **Custom**), y **Monitor Client Library**.

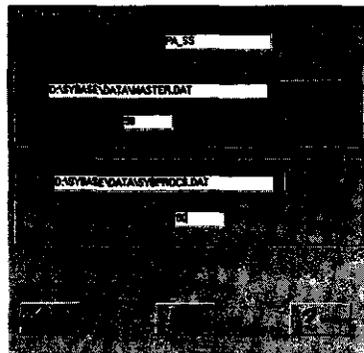
Después se selecciona **SQL Server**, y se oprime el botón **Continue** (ver *fig. 2.1.3*).

A continuación pregunta por el nombre del servidor, el cual dependerá del aeropuerto o de la unidad de Negocios al que pertenece el Servidor (*fig. 2.1.7*).

El formato del nombre del servidor es: **XXX\_SS**. Donde **XX** indica las siglas del aeropuerto o de la unidad de negocios, y **SS** para SQL Server ó **AS** para Adaptive Server.

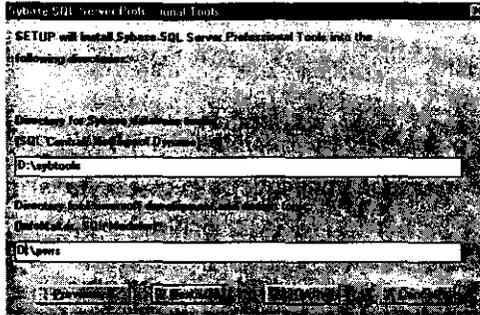
Por ej. si es el servidor de la unidad de Negocios del Pacífico, entonces el servidor se llamará **PA\_SS**.

A continuación, pregunta las rutas y tamaños de los dispositivos master y subsystemprocs, las cuales por omisión corresponden a la ruta en la que se instaló el SQL Server; en cuanto al tamaño de las Bases de Datos, serán de 50 MB, tal como se muestra en la *fig. 2.1.7*.



*fig. 2.1.7*

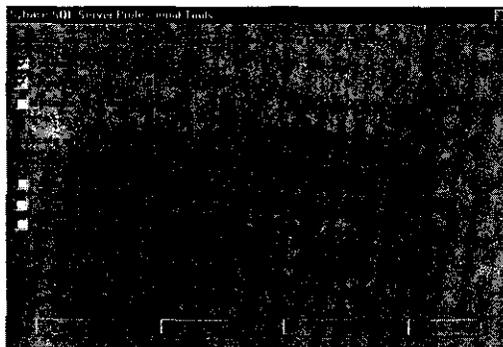
Al presentarse el cuadro de diálogo siguiente (*fig. 2.1.8*) se revisará solamente que se tengan las opciones *D:\sybtools*, y *D:\pwrs* en los respectivos campos.



*fig. 2.1.8*

Oprima el botón Next.

En la siguiente ventana de diálogo (*fig. 2.1.9*) el *instalador* le cuestionará acerca de las herramientas que se quieren instalar.



*fig. 2.1.9*

Seleccione **Sybase SQL Central**. Presione el botón Next.

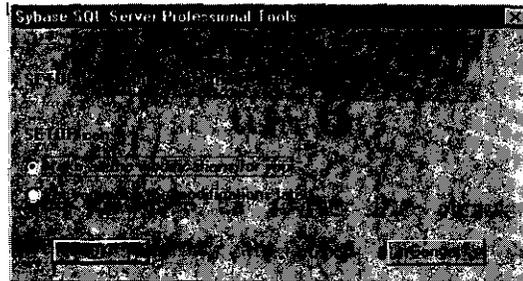
Ahora presentará la última ventana de diálogo (**fig. 2.1.10**) antes de proceder a copiar los archivos necesarios.



**fig. 2.1.10**

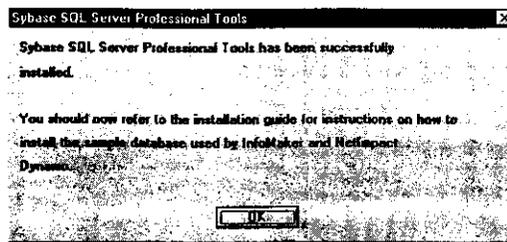
Opima el botón Ok.

Para permitir que el instalador haga las modificaciones automáticamente, oprima el botón Ok para iniciar la instalación, como indica la siguiente figura.



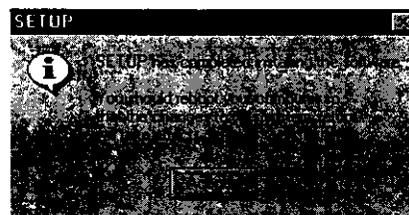
**fig. 2.1.11**

Al terminar la instalación, se presentará el mensaje ilustrado en la **fig. 2.1.12**. Informando que la instalación ha finalizado de manera satisfactoria. Presione el botón Ok para aceptar:



*fig. 2.1.12*

Presione el botón **OK** cuando aparezca la siguiente ventana, indicando que el sistema necesita reiniciarse para reconocer los cambios:



*fig. 2.1.13*

Una vez instalado SQL Server, es necesario que **REINICIE LA COMPUTADORA** para que todos los cambios realizados durante la instalación surtan efecto.



*fig. 2.1.14*

Una vez terminada la instalación de SQL Server, el siguiente paso es configurar el servidor para estandarizarlo con los demás servidores. Este proceso se describe en la siguiente sección CONFIGURACION DE LOS SERVIDORES ADAPTIVE/SQL SERVER.

Cabe señalar que este punto es aprovechado para realizar algunos pasos de optimización que se efectúan con todos los Servidores para así tener un buen funcionamiento de éstos.

## **2.2. CONFIGURACION DEL SERVIDOR**

Antes de realizar este procedimiento Ud. debió haber instalado Sybase Adaptive/SQL Server, en caso contrario favor de referirse a la sección INSTALACION DE SYBASE ADAPTIVE/SQL SERVER.

### **NOTA IMPORTANTE :**

En las siguientes instrucciones se indican cambios. ES IMPORTANTE QUE GUARDE CADA OPCION (SALVAR) EN CADA UNA DE ELLAS, DE LO CONTRARIO el proceso puede tardar demasiado ó trabar el servidor.

El programa en el que se configura el Servidor se llama: SYCONFIG:EXE y se encuentra con la siguiente rutina que a continuación se muestra en la *Fig. 2.2.1*.

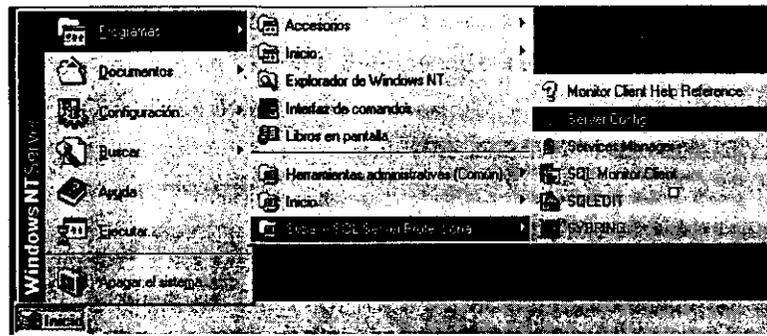


fig. 2.2.1

Se mostrará una pantalla en la que se presentan las siguientes opciones:

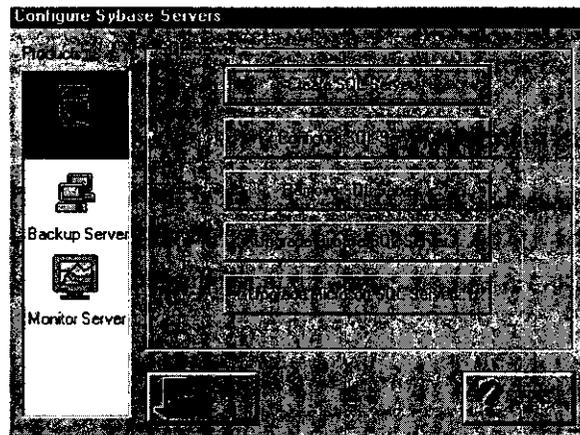


fig. 2.2.2

Se debe seleccionar Configure Server

Seleccione el Servidor a configurar:

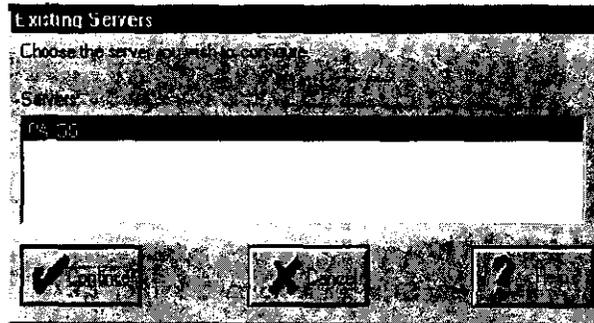


fig. 2.2.3

Introduzca el nombre del usuario administrador (sa) y la contraseña en la siguiente pantalla:

NOTA: Si acaba de instalar el servidor, el usuario sa no tiene contraseña

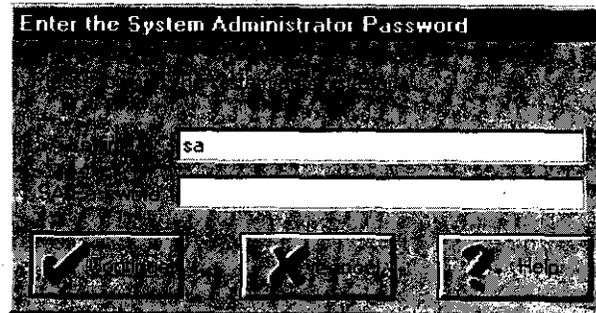
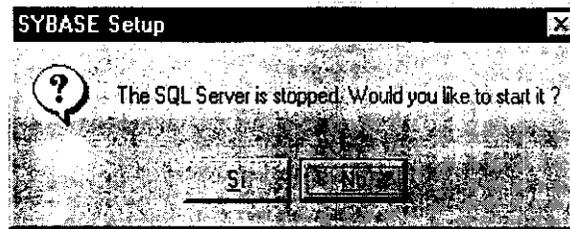


fig. 2.2.4

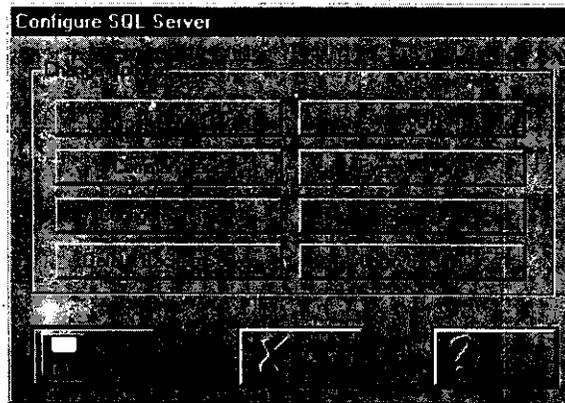
El programa le preguntará si quiere iniciar el Servidor.



*fig. 2.2.5*

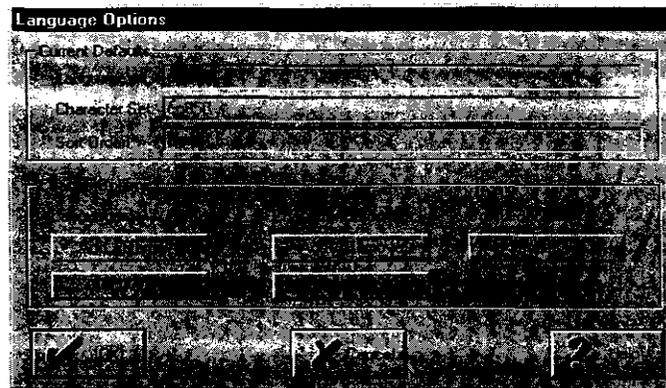
Conteste **Si** y continúe.

Inmediatamente aparece la siguiente pantalla (*fig. 2.2.6*).



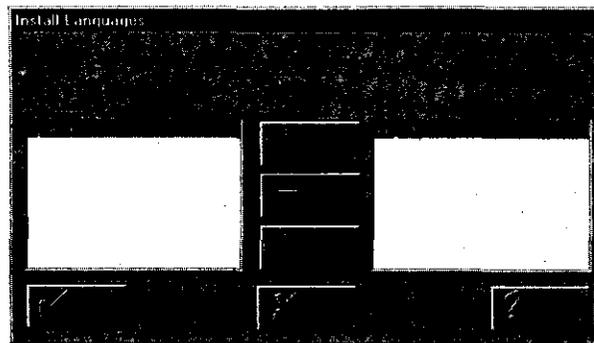
*fig. 2.2.6*

Se selecciona **Language**. Entonces aparece la pantalla de la **fig. 2.2.7**:



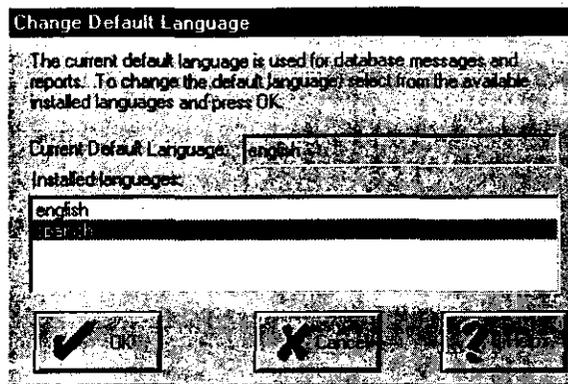
**fig. 2.2.7.**

Al seleccionar la opción **Add/Remove** se agrega el idioma **SPANISH**.



**fig. 2.2.8**

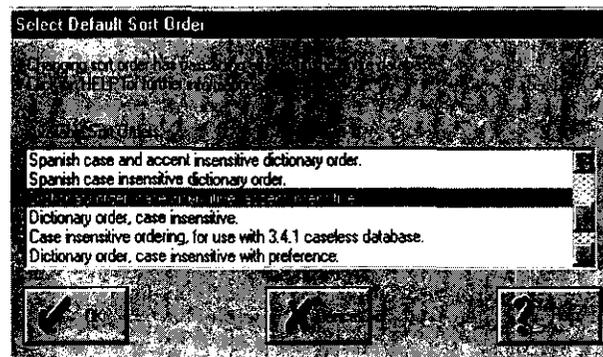
En la opción de **Set Default** de la pantalla (**fig. 2.2.9**) se selecciona el idioma español como default y se guarda la modificación.



**fig. 2.2.9**

Al presionar OK, volverá a la pantalla de la fig. 2.2.6. Haga clic en Save.

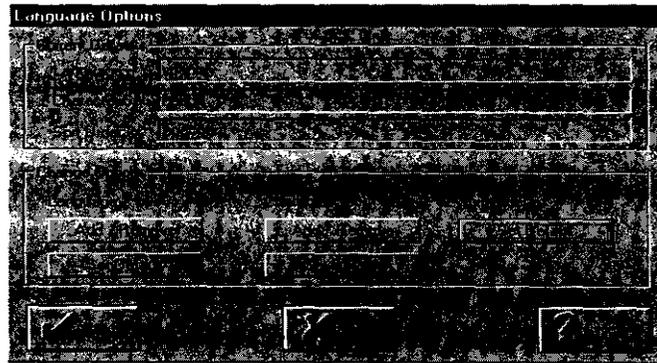
En la opción **Select Default Sort Order** (**fig. 2.2.10**) Se selecciona **Dictionary order case insensitive accent insensitive**. Se guarda la información:



**fig 2.2.10**

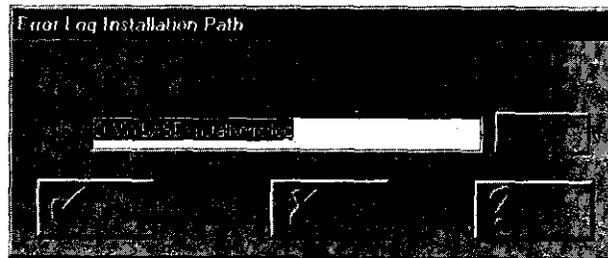
Al presionar **OK**, volverá a la pantalla de la **fig. 2.2.6**. Haga clic en **Save**.

En la opción de **CHARACTER SET** se selecciona el **CP850** -- probablemente ya este seleccionada -- (Si no lo está, se selecciona y se salva):



**fig. 2.2.11**

Al presentarse nuevamente la pantalla de la figura 3.6. y seleccionar la opción de **Error Log Path** (**fig. 2.2.12**) se cambia sólo el nombre (pero no la ruta) a **Nombre Servidor.log**, donde el nombre del servidor es el nombre del Servidor de SQL Server:



**fig. 2.2.12**

Ejemplo:

D:\sybase\Install\err.log

se cambia a:

D:\sybase\Install\PA\_SS.log (donde el nombre del servidor de  
SQL Server es PA\_SS).

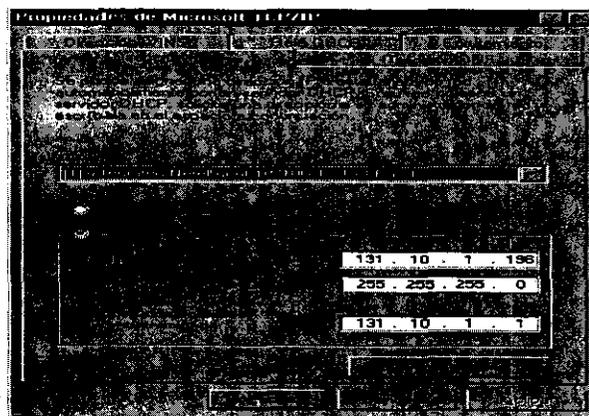
Al presionar **OK**, volverá a la pantalla de la *fig. 2.2.6*. Haga clic en **Save**.

En este punto termina de utilizarse el programa **SYCONFIG.EXE**.

Recuerde que debe verificar la dirección IP a la que apunta el Servidor SQL dentro del archivo **SQL.INI** (ver *fig. 2.2.13*)

A continuación se deben especificar las características bajo las que funcionará la comunicación en RED, y la localización del servidor. Lo anterior deberá realizarlo en las propiedades de entorno de red, o través del icono red, del menú panel de control de Windows NT.

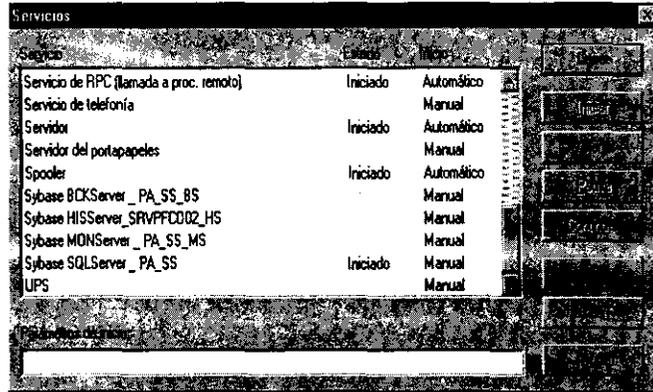
A continuación se muestra una pantalla similar a la que le aparecerá:



*fig. 2.2.13*

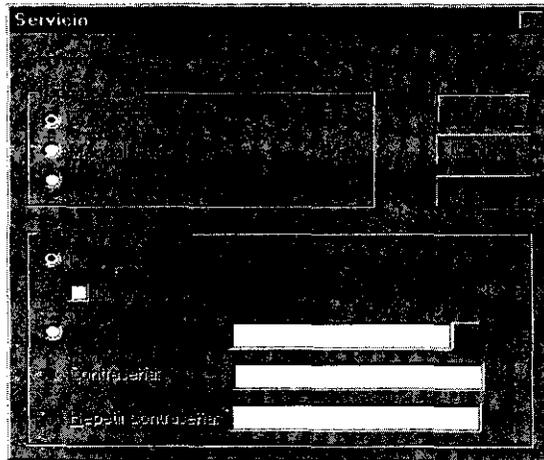
**NOTA :** La dirección IP que aparece en pantalla anterior, deberá ser la misma que éste declarada en el archivo SQL.INI, tal como se comento con anterioridad.

Dentro de la carpeta **panel de control**, en el menú **servicios**:



**fig. 2.2.15**

Asegúrese de cambiar el inicio de los servicios de los Servidores, al modo **automático**.



**Fig. 2.2.15**

Termine la configuración de los servicios para **Backup Server**, **Historical Server**, **Monitor Server** y **SQL Server**.

En el archivo `D:\Sybase\ini\SQL.INI` (siempre y cuando el subdirectorio de SQL Server sea `D:\Sybase`). Se deben tener 2 ítems por cada sección correspondiente a cada servidor.

[Servidor]

master={Protocolo},{Dirección IP},{Puerto}

query={Protocolo},{Dirección IP},{Puerto}

Donde:

Servidor = Nombre del Servidor que se le dio al instalar el SQL Server.

Protocolo = El protocolo se establece con Windows Sockets.

Dirección IP = Dirección IP del Servidor.

Puerto = Puerto lógico por medio del cual se recibe y envía información.

Como ejemplo tenemos la configuración del archivo `SQL.INI` para el servidor [PA\_SS]:

```
master=NLWNSCK,131.10.1.29,5000
```

```
query= NLWNSCK,131.10.1.29,5000
```

**NOTA** : La dirección IP varía dependiendo del Servidor.

Para cada Servidor se deben tener 5 secciones que son:

[Backup Server]	puerto: 5001
[Monitor Server]	puerto: 5002
[Historical Server]	puerto: 5003

Para el Historical Server por default pone le nombre del Servidor (CPU), no se debe de cambiar este nombre.

[Executable Server].- puerto: 5004 (solo en el Adaptive Server)

Una vez realizada la verificación del archivo SQL.INI, reinicie la computadora.

### **2.3. CREACION Y DEFINICION DE LA BASE DE DATOS DEL SISTEMA DE CONTABILIDAD**

#### **DEFINICION DE DISPOSITIVOS DE LA BASE DE DATOS**

Para crear una base de datos, el primer paso que debemos realizar es la creación de los dispositivos que la van a conformar. Entonces para ello es necesario definirlo en el programa SQL Central que se encuentra en la ruta D:\sybtools\win32\scview.exe

Posteriormente es necesario conectarse al servidor con un usuario que tenga privilegios de "sa" (System Administrator). A continuación, se presentara un despliegue de carpetas en donde la carpeta Databases contendrá las bases de datos que se encuentran activas en el servidor. Para agregar una nueva Base de Datos, es necesario dar doble click sobre la opción "Add Database Device" como se muestra en la **figura 2.3.1:**

Name	Size (MB)	Unused (MB)	Physical Name
ASAbt	100	100	D:\SYBASE\data\ASAbt.dat
ASAdat	1024	1024	D:\SYBASE\data\ASAdat.dat
ASAlag	400	400	D:\SYBASE\data\ASAlag.dat
master	50	41	master.dat
PAbt	100	100	D:\SYBASE\data\PAbt.dat
PAdat	1024	1024	D:\SYBASE\data\PAdat.dat
PAidx	1024	1024	D:\SYBASE\data\PAidx.dat
PAlog	400	400	D:\SYBASE\data\PAlog.dat
sysprocsdev	50	34	D:\SYBASE\DATA\SYBPROCS.DAT

fig. 2.3.1

Esta acción comenzará una serie de pasos que le guiarán paso a paso a la definición de cada uno de los dispositivos que conforman la Base de Datos. A continuación se presentará la pantalla de la **figura 2.3.2**:

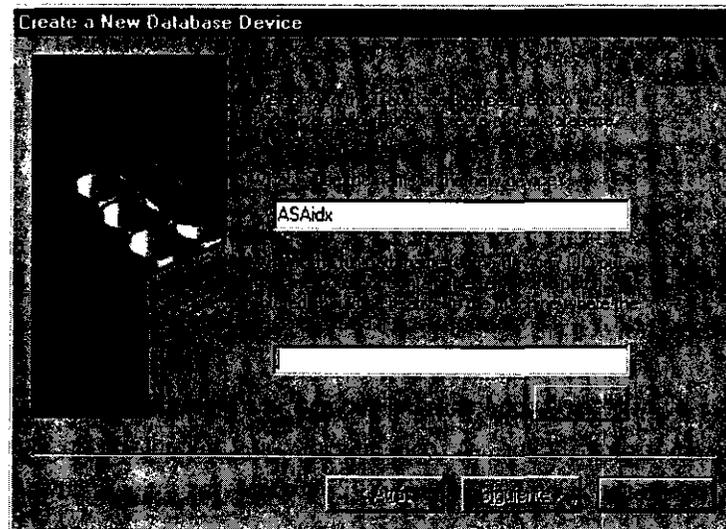


fig. 2.3.2

En esta pantalla lo que se define es el nombre del dispositivo y la ruta en donde se alojará. Por normatividad de la Gerencia de Informática, los nombres de los dispositivos tendrán la siguiente nomenclatura:

XXX99xxx.dat

Donde:

XXX Indica el nombre de la Base de Datos

99 Indica el consecutivo del dispositivo

xxx Indica el propósito para el que será utilizado el dispositivo:

Dat para contener las tablas del sistema

Bit para contener las tablas de Bitácora

Idx para contener los índices de las tablas

Log para contener la tabla del Log de Transacciones

NOTA: El consecutivo se activará si es necesario incrementar la Base de Datos.

Si la Base de Datos se llama ASA, entonces los dispositivos que se crearán serán los siguientes:

ASADat.dat

ASABit.dat

ASAIIdx.dat

ASALog.dat

Todos estos dispositivos serán creados en la carpeta D:\Sybase\Data como se muestra a continuación en la **figura 2.3.3**:

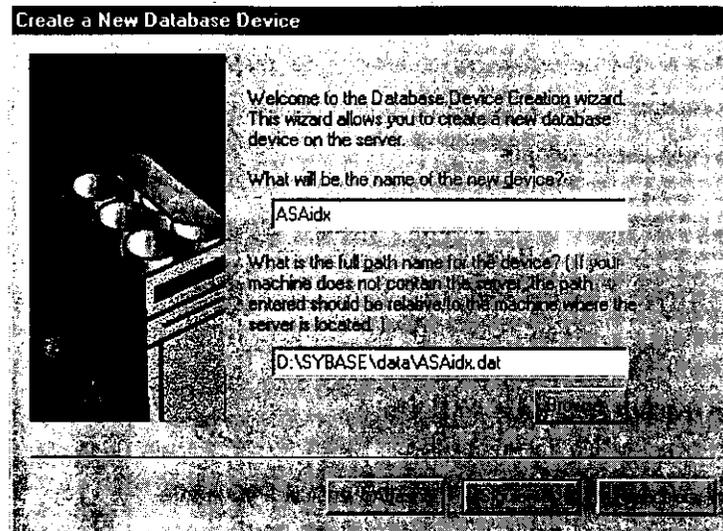


fig. 2.3.3

A continuación se presentará la pantalla que se muestra en la **figura 2.3.4**:

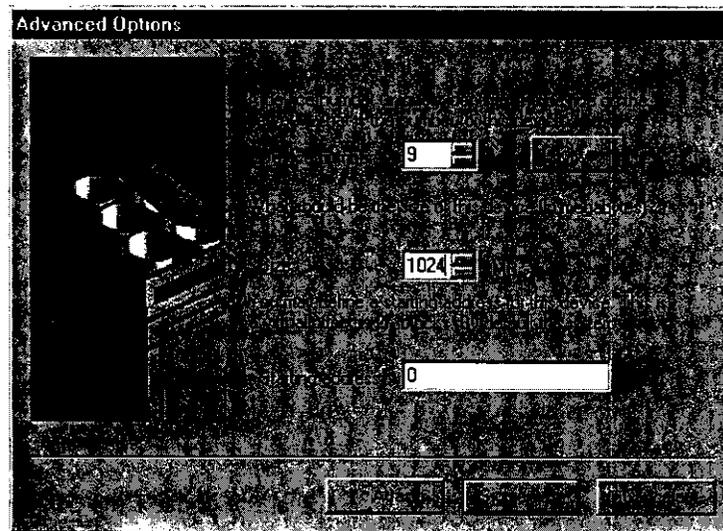


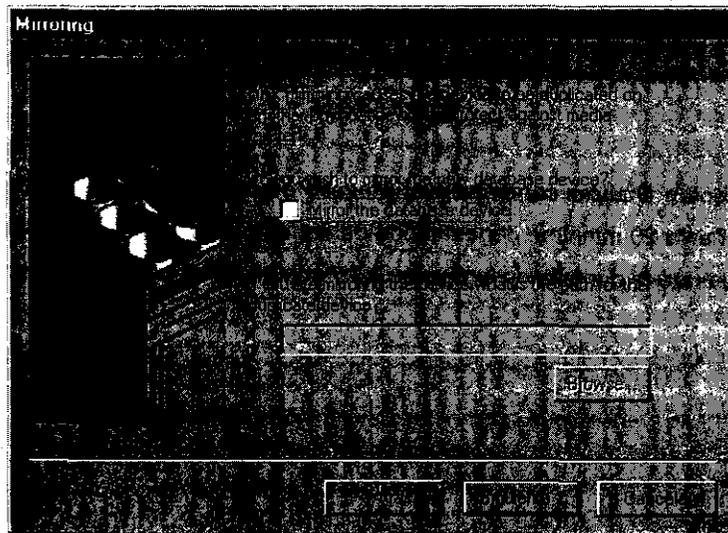
fig. 2.3.4

En esta pantalla lo que se requiere es definir la identificación, el tamaño y la dirección en memoria a partir de la cual el dispositivo será definido. La identificación es un

número que se designa automáticamente por lo que no es necesario cambiarla, la dirección inicial tampoco es necesaria definirla ya que no es necesaria manejarla. Lo único que es necesario definir es el tamaño del dispositivo. El tamaño esta en función del propósito al que será dedicado teniendo como estándar las siguientes tamaños iniciales para cada uno de los que es necesario crear:

ASAdat	1024
ASABit	100
ASAI dx	1024
ASALog	400

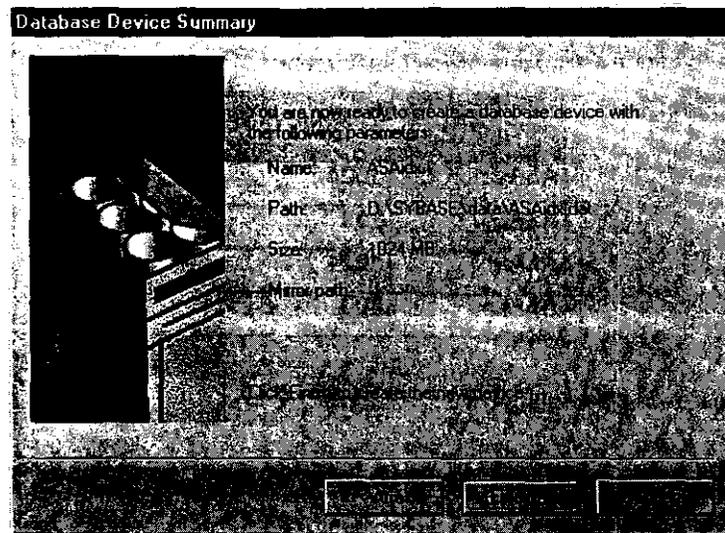
A continuación se presenta la pantalla de la **figura 2.3.5**:



**fig. 2.3.5**

En esta pantalla no se define absolutamente nada, puesto que no se tienen políticas para el manejo de bases de datos en espejo.

Y por último, muestra la pantalla de la **figura 2.3.6** en la que se despliega un resumen de todo lo que se definió anteriormente:



**figura 2.3.6**

Al oprimir el botón finalizar comenzará el proceso de creación del dispositivo, el cual puede tardar hasta 20 minutos, dependiendo del tamaño del mismo. De tal manera que al finalizar mostrará la pantalla de la **figura 2.3.7** con el dispositivo que acaba de crear:

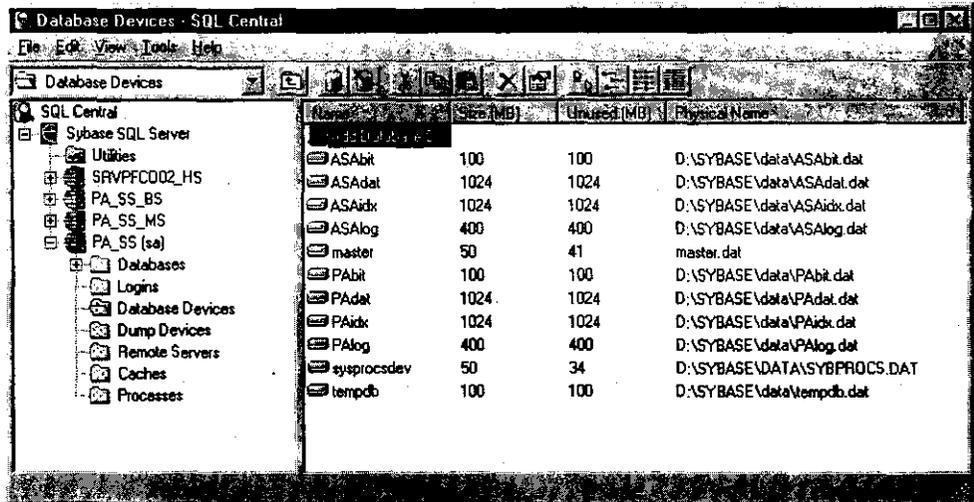


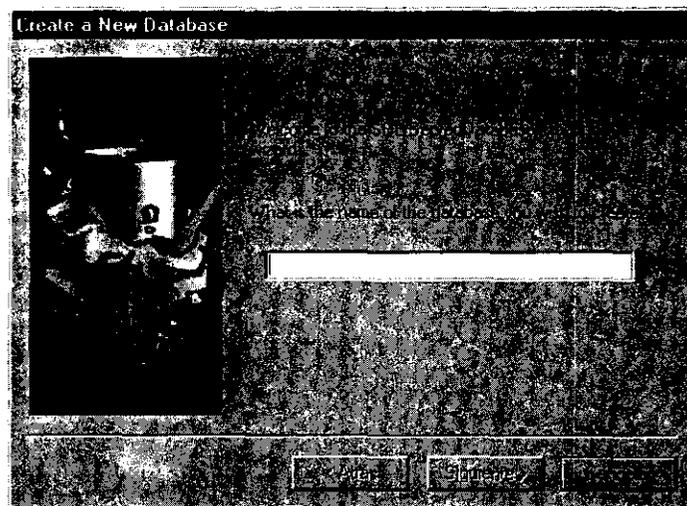
fig. 2.3.7

Este proceso será ejecutado para cada uno de los dispositivos que conformarán la Base de Datos.

### **CREACION DE LA BASE DE DATOS**

Una vez definidos los dispositivos, es necesario crear la base de datos, para ello se debe conectar con un usuario que tenga privilegios de administrador, y en la carpeta Database se seleccionará la opción Add New Database como se muestra en la siguiente figura:

Una vez ejecutada la acción se comenzarán a presentar una serie de pantallas que guiarán al usuario paso a paso a la creación de la Base. En primer lugar se pregunta el nombre de la Base de Datos como se muestra en la figura 2.3.8.:



**fig. 2.3.8**

A continuación se debe determinar la segmentación de la Base de Datos, esto es, qué dispositivos conforman la Base, y que tamaño de cada uno de los dispositivos se asigna a la Base. Primeramente se muestra la figura 2.3.9. para que posteriormente se oprima el botón Add:

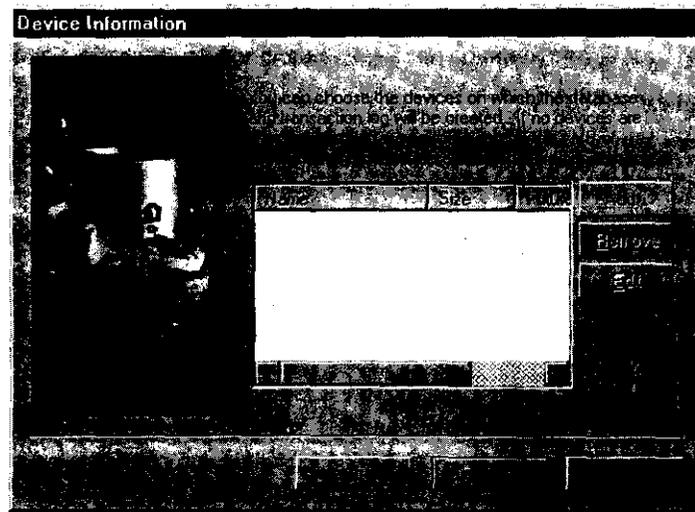


fig. 2.3.9

El siguiente paso es seleccionar para que será dedicado ( Data o Transaction Log – Datos o Log de Transacciones respectivamente), el dispositivo que formará parte de la Base de Datos y asignar el espacio del dispositivo que se dedicará a la Base ver figura 2.3.10.

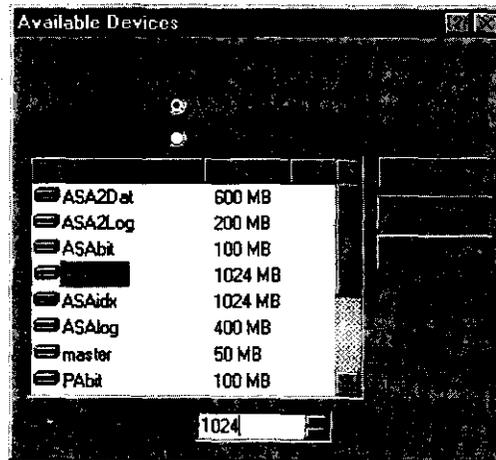


fig. 2.3.10

Este procedimiento se efectúa para cada uno de los dispositivos que definimos en la sección anterior, quedando definida la segmentación de la Base de Datos como se muestra en la figura 2.3.11:

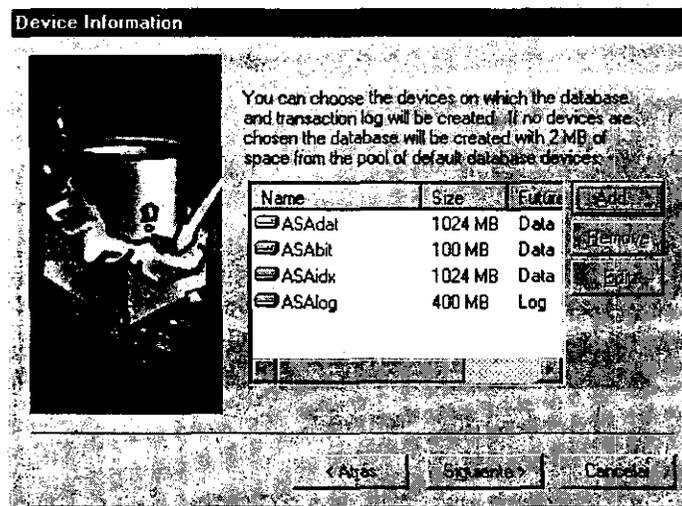


fig. 2.3.11

NOTA: Es importante que cuando se asigne el dispositivo del Log de Transacciones seleccione la opción "Transaction Log" como se muestra en la figura 2.3.12

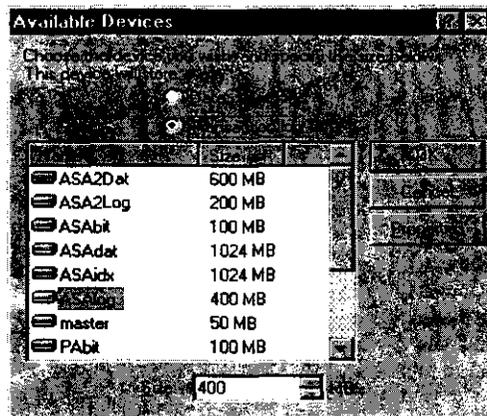
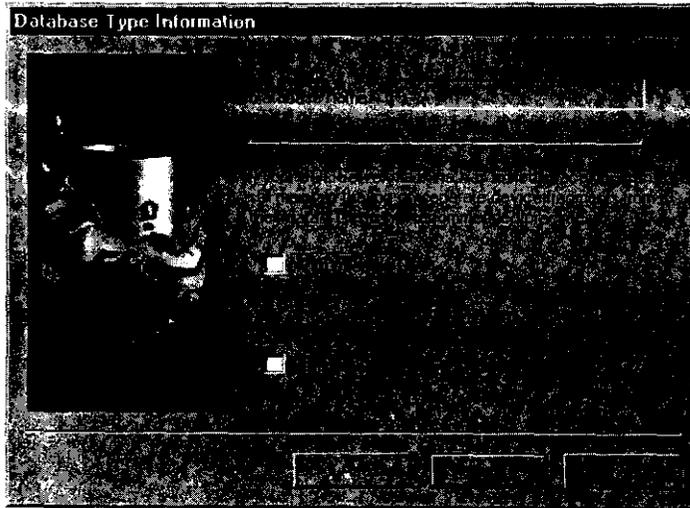


fig. 2.3.12

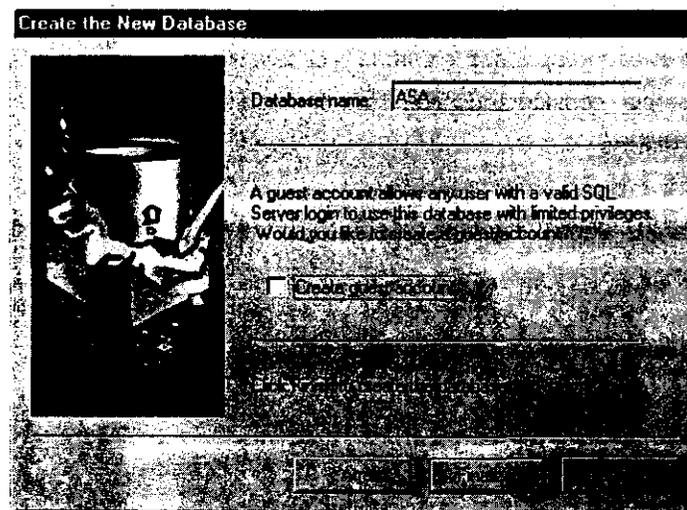
**NOTA IMPORTANTE:** La definición del espacio asignado a los dispositivos y el orden en que se definen es **indispensable** para la recuperación de la base de datos en caso de una contingencia, considerando que será cargada en **otra base de datos** con las mismas características.

Después mostrará la siguiente pantalla (figura 2.3.13) a la que solamente se oprimirá el botón "Siguiente":



**fig. 2.3.13**

En la pantalla que se muestra a continuación en la figura 2.3.14 solo se oprime el botón de "Finalizar"



**fig. 2.3.15**

A continuación comenzará el proceso para crear la Base de Datos, el cual puede durar hasta 1 hora dependiendo del tamaño de la misma.

#### **2.4. DEFINICIÓN DE AREAS DE MEMORIA DE ACCESO RAPIDO (CACHES)**

Para optimizar el performance del Servidor Adaptive/SQL Server se utilizaron las áreas de acceso rápido (que en adelante se conocerán como '**cachés**'), que se definen sobre la memoria designada al servidor y, para cada una de los cachés, almacenarán datos de sólo ciertas tablas. Su función consiste en mantener disponible en la memoria los datos más comúnmente consultados, dando como resultado una respuesta más rápida a las peticiones de los clientes.

**NOTA:** Para mayor información sobre "**cachés**", referirse al manual de Adaptive/SQL Server de Performance and Tuning, Capítulo 15 - Memory Use and Performance, página 14.

---

### 3.MANTENIMIENTO Y MONITOREO DEL SISTEMA DE CONTABILIDAD

El Sistema de Contabilidad requiere de un constante mantenimiento y monitoreo debido a que las necesidades de espacio, performance y optimización de procesos sufren un cambio constante y dinámico. La información almacenada en la Base de Datos del Sistema de Contabilidad crece y requiere de nuevas adaptaciones; además de que es necesario verificar que la respuesta y el espacio del Log de Transacciones, por ejemplo, cuenten con las condiciones óptimas para que los usuarios obtengan una respuesta adecuada.

De tal manera que es necesario realizar la verificación constante de todos aquellos parámetros que pudieran ser causales de algún tipo de desperfecto o anomalía en el Sistema. Tales causales pueden ser miles y de distinta naturaleza, pero nos concentraremos en aquellas que son las más críticas y que por lo tanto requieren de mayor cuidado, además de que pudieran detener la operación del sistema.

En primer lugar tenemos el Log de Transacciones; este tiene instrumentado una serie de eventos que se disparan automáticamente cada que se cumplan ciertas circunstancias llamados "Thresholds" (ver el documento: *Procedimiento para el respaldo y recuperación de Bases de Datos*), vaciando así el Log de Transacciones; pero puede suceder que ninguno de esos eventos se dispare, provocando el llenado del Log de Transacciones y la consecuente paralización del Sistema. Por lo tanto, la labor de monitoreo se debe de efectuar lo mas frecuentemente posible, para que esto no suceda. Si esto llegase a suceder y decida vaciar el Log de Transacciones y tiene problemas, entonces consulte el apartado *SOLUCION DE PROBLEMAS* que le indicará que es lo que debe de hacer.

## **4.PROCEDIMIENTO PARA EL RESPALDO Y RECUPERACION DE LA BASE DE DATOS**

### **4.1.OBJETIVO**

Asegurar el resguardo y la recuperación de la información que se genera en una Base de Datos. De igual modo, se determinará el procedimiento ó plan de recuperación de la información a partir de los resguardos realizados a una Base de Datos.

### **4.2.RESPALDO DE LA INFORMACION**

Como medida de seguridad, es de crucial importancia el respaldo de la Base de Datos a fin de mantener la integridad de la misma, ya que en cualquier momento pudiera existir el riesgo de presentarse alguna falla ó anomalía en el disco duro ó en el mismo Servidor, el cual contiene la Base de Datos.

Para ello, se requiere definir mecanismos que ayuden a resguardar la información y en caso de presentarse alguna falla ó problema recuperarla inmediatamente. Dichos mecanismos se llevarán a cabo a través del RESPALDO y la RECUPERACIÓN de Bases de datos implementadas en el DBMS (Data Base Management System) Adaptive/SQL SERVER. (A partir de éste momento se referirá indistintamente al DBMS como Adaptive/SQL Server).

Los respaldos realizados se generarán en archivos con un formato y manejo específico que brinda el DBMS.

---

### **4.3. MEDIOS DE RESPALDO**

#### **4.3.1. Disco Duro**

Uno de los medios en los que se pueden realizar los respaldos de la B.D. en forma inmediata, es el disco duro del propio servidor donde reside la B.D. ó en el de algún otro servidor ó cliente (lo cual no es nada recomendable), sin embargo, esta práctica no debe realizarse a menos que se tenga una emergencia, y si así fuera, inmediatamente se debe realizar el respaldo del archivo generado en cinta.

Por otra parte, debido a que los archivos generados por respaldos de la Base de Datos pueden ser muy grandes, es necesario verificar que exista espacio disponible en el Disco Duro, de igual manera este criterio se aplica para el respaldo de los archivos del Log's de Transacciones.

#### **4.3.2. Cartucho de Datos**

El otro medio en el que se realizan los respaldos es en una cinta o cartucho, el cual es el dispositivo de almacenamiento más recomendable para realizar los respaldos. El servidor debe contar con una unidad de cinta para poder ejecutar el respaldo en este tipo de medio.

#### **NOTA:**

En servidores NT la longitud de la cinta es indistinta, mas sin embargo, en el servidor Hewlett-Packard Serie 9000 Modelo T-500 sólo permite el respaldo en cintas de longitud de 90 m.

### **4.3.3. TIPOS DE RESPALDO**

#### **4.3.3.1. Base de Datos**

El RESPALDO de la Base de Datos se hace para todo el contenido de ésta, lo cual significa que tendrá un tamaño equivalente al de toda la información almacenada en todos los objetos que conforman la Base de Datos.

El respaldo de la Base de Datos se realiza conectándose directamente al DBMS, ejecutando el procedimiento `sp_dump`, cuyo uso se describirá posteriormente.

#### **4.3.3.2. LOG DE TRANSACCIONES**

El RESPALDO del Log de transacciones consiste en el resguardo de la información contenida en una tabla de la base de datos llamada `syslogs`, dicha tabla reside en el dispositivo creado expresamente para contenerla llamado así `{Nombre de la Base de Datos}.log.dat`; el cual se encuentra localizado en `D:\Sybase\data`

La Tabla `syslogs` tiene importancia crítica en la vida del sistema ya que en dicha tabla se registran todas las transacciones realizadas en la Base de Datos, es decir, todos los inserts, updates y deletes efectuados en ésta, teniendo así, el registro de todos los movimientos realizados en dicha Base de Datos. De tal manera que el resguardo de dicha información constituye un recurso imprescindible en la recuperación de la Base de Datos.

Por otra parte, el espacio que ocupa el Log de Transacciones (así es como será llamado de ahora en adelante) es limitado, por lo que es necesario implementar algún mecanismo para el mantenimiento del Log de Transacciones.

El mecanismo consiste en hacer el RESPALDO del Log de Transacciones, esto es se generarán archivos que contengan las Transacciones hechas en la Base de Datos.

Cada vez que se respalda el Log de Transacciones automáticamente se vacía, para así comenzar de nuevo a registrar transacciones. Bajo este esquema es necesario plantear algunas definiciones para el manejo de los respaldo en general.

#### **4.3.4. Niveles de Respaldo**

Los respaldos se efectúan de manera incremental, esto es, en un inicio se parte de un respaldo global de la Base de Datos y después de éste, van incrementándose los respaldos de acuerdo a las transacciones ó movimientos que se generaron posteriormente al respaldo global inicial

De acuerdo a lo anterior, llamaremos respaldo de Nivel 0 al respaldo global de la Base de Datos, es decir, al respaldo de todos los objetos de dicha Base de Datos. Este evento constituye el punto de partida. Una vez hecho el respaldo de Nivel 0 se seguirán generando transacciones en la Base de datos, para entonces tener dos alternativas:

- 1) Realizar un nuevo respaldo de Nivel 0 ó,

- 2) Respaldo sólo aquellas transacciones que se realizarán inmediatamente después del respaldo de Nivel 0.

Por una parte, el primero tarda más tiempo ya que se guarda toda la información de la Base de Datos y el segundo es más rápido, debido a que solo se guardan las últimas transacciones, sin embargo, ambos son 100% confiables.

Si se respalda el Log de Transacciones entonces lo llamaremos respaldo de Nivel uno. Ahora bien, si decidimos posteriormente respaldar el Log de Transacciones nuevamente, entonces le llamaremos respaldo de Nivel dos y así sucesivamente.

Por último, añadiremos que en una emergencia la recuperación de la base de datos debe realizarse bajo ciertos criterios los cuales serán descritos en el capítulo de Recuperación de la Base de Datos.

#### 4.3.5. Modo de Respaldo

Los respaldos pueden ser efectuados de 3 formas:

- 1) Manuales
- 2) Store Procedure
- 3) Automáticos

A continuación se explica en qué consiste cada uno de los modos de respaldo:

- 1) El respaldo Manual se realiza con instrucciones de **Transact SQL**, directamente en el DBMS. Este modo de respaldo sólo puede realizarlo un DBA (**Data Base Administrator / Administrador de la Base de Datos**) capacitado para ejecutar las instrucciones necesarias.

- 2) El respaldo mediante un Store Procedure es un mecanismo más simple y puede ser realizado por casi cualquier persona, lo cual permite delegar la función y así disminuir la carga de trabajo del DBA. Este consiste en conectarse al DBMS, y ejecutar el procedimiento "sp\_dumplog". Este procedimiento genera el archivo correspondiente que debe de ser respaldado en cinta posteriormente.
- 3) Por último, el respaldo automático es implementado por medio de eventos programados en el DBMS que reciben el nombre de " **Threshold** ". El **Threshold** consiste en el disparo automático del procedimiento para respaldo " sp\_dumplog " cuando se rebasa el " umbral " (**Threshold**) del espacio libre que resta en el Log de Transacciones.

NOTA: El momento en que se disparan los Thresholds es determinado por el DBA y se definen a través del programa SQL Central ó por medio de comandos del Transact SQL.

El DBA generalmente define en el Log de Transacciones un conjunto de Thresholds para distintas cantidades de espacio libre, ya que no siempre se disparan, debido a que se rebasa el límite de espacio libre especificado en una sola transacción.

Cada uno de los respaldos del Log de Transacciones se genera en archivos, y representan las transacciones realizadas entre el momento del último respaldo de la Base de Datos ó el último respaldo del Log de Transacciones y el momento del actual respaldo del Log de Transacciones.

Por ejemplo, si se tiene un tamaño del Log de Transacciones de 400 MB, se puede determinar que el disparo del Threshold (umbral) sea al tener un espacio libre de 350 MB, es decir, 50 MB de espacio ocupado en transacciones esto me generará un archivo de respaldo del Log de Transacciones de 50 mb aproximadamente.

Es conveniente recordar que, los respaldos automáticos del Log de Transacciones, se realizan directamente al disco, por lo que es necesario transferirlos oportunamente a cartuchos dedicados para el proceso de Backup y así evitar la saturación del disco, además de asegurar la recuperación de la Base de Datos en caso de falla del disco duro. Al final de este documento se propone una sugerencia para el manejo de los respaldos.

#### **4.4. Nomenclatura de los archivos de respaldo**

Como se ha mencionado anteriormente, los respaldos de las Bases de Datos y del Log de Transacciones se generan en archivos, y para poder identificarlos se les debe dar un nombre.

Dicho nombre esta formado por el nombre de la Base de Datos y la fecha en la que fueron realizados, para facilitar su identificación y el restablecimiento en dado caso. La Nomenclatura es la siguiente:

**AAAYYYMMDDHHMMSS** donde;

<b>AAA</b>	indica el nombre de la Base de Datos,
<b>YYYY</b>	indica el año,
<b>MM</b>	indica el mes,
<b>DD</b>	indica el día,
<b>HH</b>	indica las horas,
<b>MM</b>	indica los minutos

**SS** indica los segundos

En que se realizó el respaldo.

Para el respaldo de la Base de Datos el sistema no pone extensión a los archivos, en el caso del Log de Transacciones la extensión es ".log".

Nota: Cabe señalar que los procedimientos `sp_dump` y `sp_dumplog` generan automáticamente los nombres de los archivos con la misma nomenclatura detallada anteriormente.

#### **4.5. Instrucciones y consideraciones básicas para llevar a cabo un respaldo**

Antes de realizar un respaldo de la información se deben seguir los siguientes pasos:

- Asegúrese primeramente que la cinta se encuentre debidamente insertada dentro del "drive" que le corresponde.
- Una vez insertado el cartucho automáticamente la unidad reembobinará la cinta dejándola lista para iniciar el respaldo de la información. **(1)**
- Deben existir además los dispositivos de almacenamiento (Dump Devices, ASADisk, ASATape) **(2)**
- Deben existir los Store Procedures (`sp_dump`, y `sp_dumplog`) **(2)**

**Nota:**

**(1)** Es recomendable no lanzar inmediatamente ninguna instrucción hasta que el cartucho este listo, es decir que la luz que indica lectura de cinta (luz naranja) no este encendida.

- (2) En caso de no existir dichos comandos ó que se requiera de verificar su existencia favor de consultar con el DBA del Sistema.

#### 4.6. Procedimiento para respaldar una Base de Datos

A continuación se describe el procedimiento para respaldar una Base de Datos. Primeramente, es necesario que se encuentre instalado el software cliente de Sybase esto en el caso de que el Backup vaya a correrse desde una PC cliente ó directamente desde el Servidor de la Base de Datos. (Este software se encuentra instalado en la carpeta con la siguiente ruta: C ó D:>\Sybase\bin\)

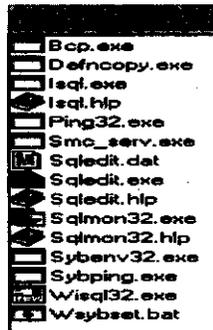
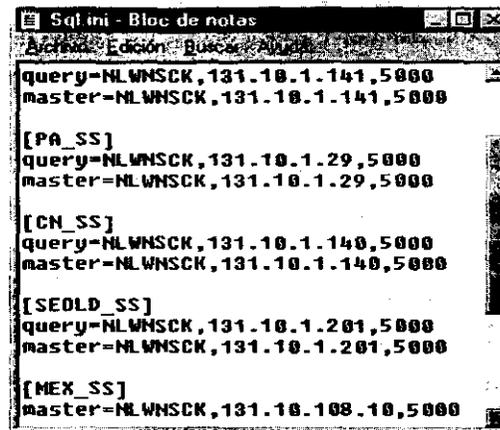


fig. 4.6.1.

Además, el archivo "sql.ini" (C:\> Sybase\ini\sql.ini) deberá tener claramente especificados los nombres de los servidores con su correspondiente Dirección IP. Observe como en la figura 2.6.2 esta definido un ejemplo para el archivo "sql.ini". (Para más información consulte al 'DBA').



```
Sql ini - Bloc de notas
Edición: Buscar

query=NLWNSCK,131.10.1.141,5000
master=NLWNSCK,131.10.1.141,5000

[PA_SS]
query=NLWNSCK,131.10.1.29,5000
master=NLWNSCK,131.10.1.29,5000

[CN_SS]
query=NLWNSCK,131.10.1.140,5000
master=NLWNSCK,131.10.1.140,5000

[SEOLD_SS]
query=NLWNSCK,131.10.1.201,5000
master=NLWNSCK,131.10.1.201,5000

[MEX_SS]
master=NLWNSCK,131.10.108.10,5000
```

fig. 4.6.2

- a) Del menú inicio (figura 4.6.3) de un clic en la opción Ejecutar

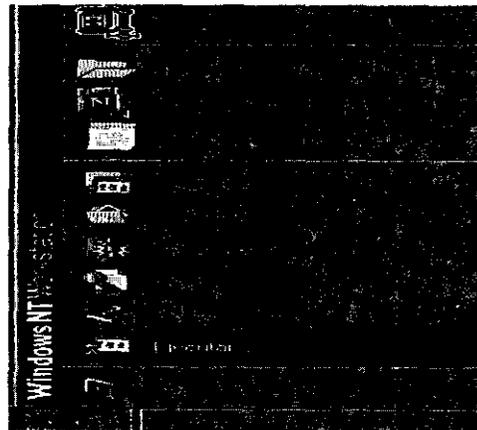


fig. 4.6.3

- b) Escriba la siguiente instrucción:

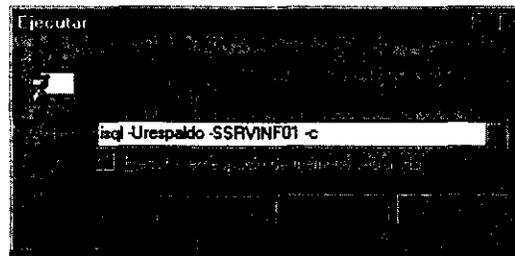
**isql -U<cuenta> -S<nombre\_servidor> -c** donde;

**isql** es el nombre del programa que sirve como interfase entre el *cliente* y el manejador de la Base de Datos

**-U(usuario)** debe indicar una cuenta de acceso al sistema (ejem.: respaldo, csanchez, etc.)

**-S(servidor)** debe indicar el nombre del Servidor en el cual está alojada la Base de datos(ASA\_SS, MEX\_SS, SE\_SS, etc.) y

**-c** este parámetro indica la terminación de la instrucción; si dicho comando no es declarado en ésta instrucción entonces deberá teclear un "go" seguido de un **Enter** después de cada instrucción. Observe el siguiente ejemplo:



**fig. 4.6.4**

**Nota:**

Observe que los parámetros **-U** y **-S** deben ir declarados con mayúsculas. Por **Normatividad** de la Gerencia de Informática el **login** ó la clave del usuario que se empleará para este tipo de procedimientos será el de **respaldo**.

c) Introduzca su password (Recuerde que para el acceso a los distintos servidores requiere de una cuenta y password, consulte al DBA). A continuación iniciará una sesión de "isql".

d) La sintaxis de la instrucción para realizar el respaldo es la siguiente:  
**1> sp\_dump <nombre de BD>, {'disk'|'tape'},{'init'|'noinit'}}**

2> ...

donde :

**sp\_dump** es el procedimiento de respaldo,

**<nombre de BD>** especifica el nombre de la Base de Datos a respaldar.

**{'disk'|'tape'}** indica en que medio ó dispositivo va a realizarse dicho respaldo (en este caso todos los respaldos se harán en cintas ó cartuchos (**tape**). En el último de los casos puede realizarse también en el disco duro (**disk**).

**{'init'|'noinit'}** el parámetro **init** se empleará cuando vaya a reiniciarse la cinta (cartucho) ó en caso contrario se utilizará el parámetro **noinit** para agregar la información a la ya existente. El parámetro está determinado por omisión, por lo que sino se inicializa la cinta no es necesario declararlo.

e) Teclee doble **Enter**. Observe el siguiente ejemplo;



```
C:\ASYBASI\ADMIN> sqlcmd -i
sqlcmd:
1> sp_dump 0001, 'tape', 'noinit'
```

*fig. 4.6.5*

f) El programa comenzará por hacer un respaldo del Log para asegurar que la totalidad de Transacciones realizadas hasta ese momento se respalden. Así mismo desplegará los detalles de dicho respaldo e irá desplegando el

porcentaje de información guardada (dumped) de la Base de Datos en cuestión. Observe la siguiente ventana :

```

C:\SYBASE\BIN\sql.exe
C> sp_dump backup 'Cape', 'normal'
13
p_dumplog: PROCEDIMIENTO LLAMADO POR sp_dumplog
p_dumplog: RESPALDO DE LOG DE TRANSACCIONES DE LA BASE DE DATOS 'OS63'
Máq 4231, Level 01, State 13
Proceeding 'sp_dumplog', Line 34:
Se va a permitir DUMP TRANSACTIONS en un dispositivo de copia donde se ha especificado
una copia de transacción de solo transacción (solo el último DUMP OBTENDRÁ una
COPIA TRANSACTIONS) en su lugar.
p_dumplog: ERROR AL RESPALDAR EL LOG (p_dumplog): 4231
p_dumplog: LOG DE TRANSACCIONES RESPALDADO EN
'D:\SYBASE\backup\OS6319961214185321.log'
p_dumplog: TIEMPO DE RESPALDO DEL LOG: Antez: '37', Despues: '39'
p_dumplog: TIEMPO DE RESPALDO DEL LOG: día 14 1996 4:53PM, día 14 1996 4:55PM

El id de sesión de Backup Server es: 39. Use este valor cuando ejecute el
procedimiento almacenado de sistema 'sp_verifybackup' después de completar
cualquier actividad de respaldo de un nivel de Backup Server.
Backup Server: 6.28.1.1: Nombre de archivo de copia '06329744301002'
Número
de sesión BSMI guardado en unidad de cinta 'SSS\BTRP01'
Backup Server: 4.56.1.1: Base de datos OS63: 25196 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 62464 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 62416 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 119302 Kilobytes DUMPed.

```

fig. 4.6.6

g) Al término de este proceso el sistema le indicará que extraiga el cartucho de la unidad, le indicará igualmente el nombre del dispositivo donde se hizo el respaldo, la fecha y la hora de inicio y final del Backup. Observe la siguiente pantalla.

```

C:\SYBASE\BIN\sql.exe
procedimiento almacenado de sistema 'sp_verifybackup' de que se completaron
cualquier actividad de respaldo de un nivel de Backup Server.
Backup Server: 6.28.1.1: Nombre de archivo de copia '06329744301002'
Número
de sesión BSMI guardado en unidad de cinta 'SSS\BTRP01'
Backup Server: 4.56.1.1: Base de datos OS63: 25196 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 62464 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 62416 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 119302 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 25196 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 62464 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 62416 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Base de datos OS63: 119302 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Nombre de base de datos 1 respaldada.
Backup Server: 7.44.1.1: Nombre de base de datos 2 respaldada.
Backup Server: 7.44.1.1: Nombre de base de datos 3 respaldada.
Backup Server: 4.58.1.1: Base de datos OS63: 25196 Kilobytes DUMPed.
Backup Server: 4.58.1.1: Nombre de archivo de respaldo de una unidad de la unidad
'SS\SM110' en posición 1 de un cartucho 3333.
p_dumplog: 6.28.1.1: RESPALDO DE TRANSACCIONES COMPLETADO.
p_dumplog: 6.28.1.1: RESPALDO DE TRANSACCIONES COMPLETADO.
p_dumplog: TIEMPO DE RESPALDO DEL LOG: día 14 1996 4:55PM, día 14 1996
4:57PM
return status: 0

```

fig. 4.6.7

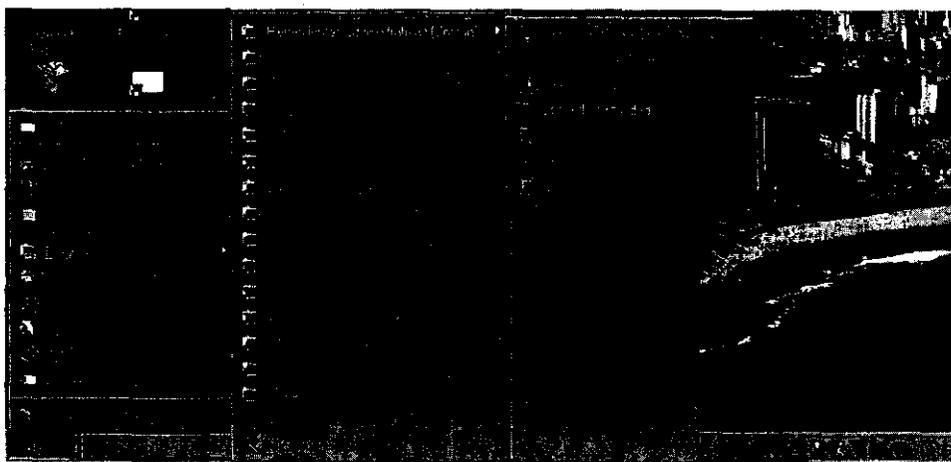
### 1. Procedimiento para respaldar archivos de Log's

A continuación se describe el procedimiento para respaldar los archivos generados del respaldo del Log's de Transacciones. Cabe mencionar que el backup de estos Log's es a través de utilerías de Windows NT. Es necesario que cuente con la cuenta (**Login**) correspondiente para llevar a cabo dicho proceso. (Para más información consulte al administrador del Servidor de Windows NT).

No olvide las instrucciones y consideraciones para llevar a cabo un respaldo descritas en la página 3.

#### **Pasos a seguir:**

- 1) Del menú **Inicio** seleccione primeramente la opción **Programas** enseguida la opción **Herramientas administrativas (Común)** y finalmente la opción **copia de seguridad**. Observe la siguiente gráfica :



**fig. 4.6.8**

- 2) El sistema presentará la ventana **Copia de seguridad** como se muestra en la siguiente figura:

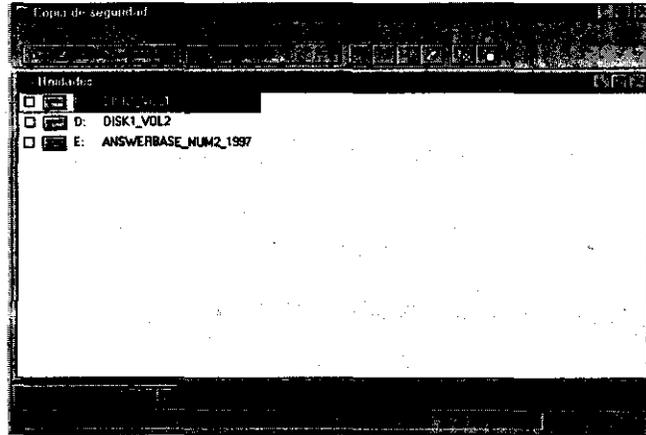


fig. 4.6.9

- 3) Haga *clíc* sobre el menú **Ventana** y seleccione la opción **Unidades**. Como resultado de ésta acción el sistema desplegará todas las unidades de almacenamiento que estén activos ó disponibles en ese momento. Vea la siguiente figura:

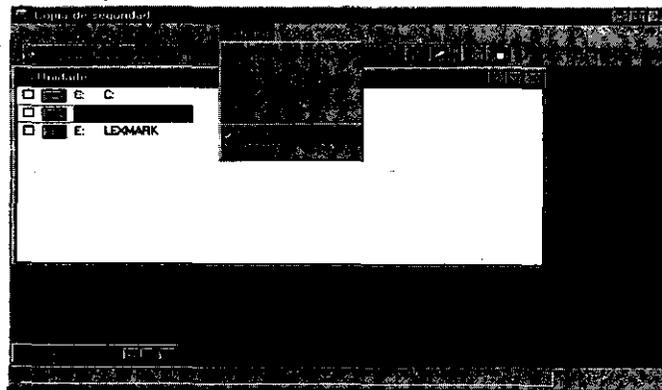


fig. 4.6.10

- 4) Haga doble *clic* sobre el icono de la Unidad D. Por **Normatividad** de la Gerencia de Informática los Log's de Transacciones son almacenados en el Disco Duro de la Unidad D, éstos Log's estarán siempre ubicados en la Carpeta **D:\SYBASE\Backup\\*.\*** , por lo tanto seleccione dicha Ruta. Vea la siguiente figura:

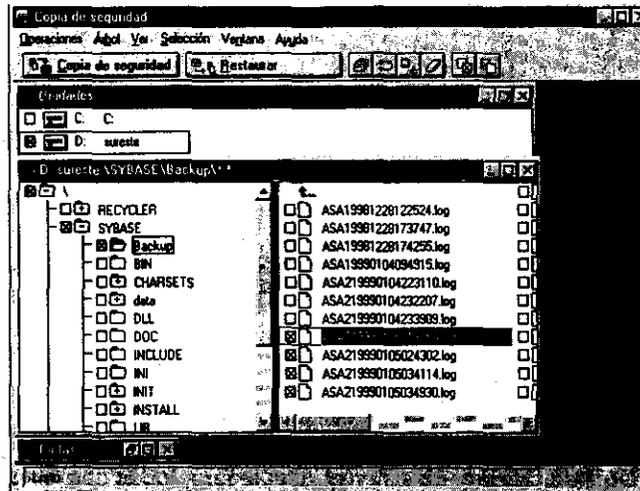


fig. 4.6.11

- 5) Observe la Nomenclatura del archivo de Log de Transacciones de la ventana anterior. Los primeros 3 ó 4 caracteres representan el nombre de la Base de Datos (Ejem.; ASA, ASA2, MEX, etc.). Los siguientes caracteres representan el año, el mes, el día, las horas, minutos y los segundos (Ejem.; ASA219990106150551.log). La extensión asignada a este tipo de archivos es ".log".
- 6) Seleccione dando un *clic* al pequeño recuadro inicial del Log's a respaldar de acuerdo a su último o recién Backup. Note que el sistema los marca

7) Una vez seleccionado(s) el (los) archivos de Log(s), puede proceder a realizar el Backup. Así que entonces oprima el botón:



Aparecerá desplegada la Ventana Información sobre la copia de seguridad como se muestra a continuación:

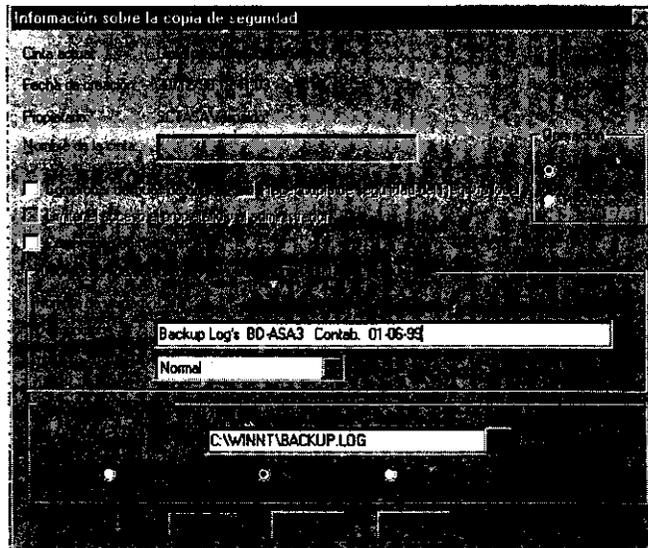


fig. 4.6.12

8) Colóquese en la operación que desee, **Anexar** ó **Reemplazar**.

- Si desea hacer el respaldo por primera vez al cartucho ó que requiera de sobrescribir el cartucho ya utilizado entonces haga *clic* en el círculo vacío para seleccionar **Reemplazar**. Escriba en el campo **Descripción** una descripción de la información que va a respaldar, preferentemente relacionado al contenido de la misma. Ejemplo: Backup de Log's de la BD-ASA3 Contabilidad 01-06-99, para los demás campos respete los

parámetros por default que el sistema propone. Finalmente presione el botón **Aceptar** e inmediatamente el sistema desplegará la siguiente ventana, en el caso de que se vaya a sobrescribir la información:

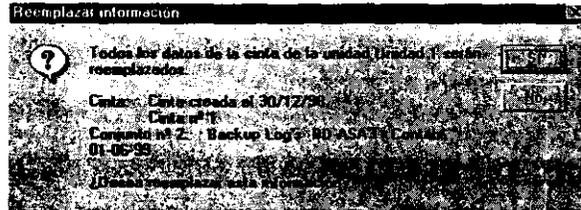


fig. 4.6.13

- Sin embargo, si no va a inicializar su cartucho, elija la operación **Anexar** e introduzca en el campo **Descripción** la descripción general sobre la información a respaldar y acepte los parámetros por default que el sistema propone. Finalmente presione el botón **Aceptar**.  
Para cualquiera de las 2 operaciones que lleve a cabo el sistema desplegará la siguiente pantalla:

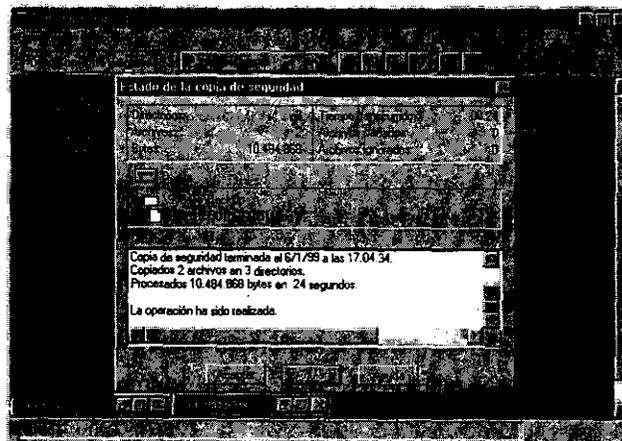


fig. 4.6.14

- 9) El sistema le indicará si el proceso fue realizado satisfactoriamente o en que en caso contrario deba reinicializar dicho proceso, si es éste el caso repita a partir del paso 7. Si no oprima **Aceptar** para cerrar la ventana **Estado de la copia de seguridad**. Cierre también la ventana **Copia de seguridad** y todos los programas. Finalmente bloquee su sesión de Windows.

## 2. Solución de Problemas

Si durante el proceso de respaldo del Nivel 0 de la Base de Datos aparece un truncamiento ó interrupción del mismo, entonces deberá realizar las siguientes tareas:

Quando se terminó el espacio disponible en el cartucho y por lo tanto el respaldo no ha sido concluido se pueden seguir las siguientes alternativas:

1. Una, es abrir otra sesión de "isql" para Abortar ó interrumpir totalmente el procedimiento.

  - Teclee la siguiente instrucción para *abortar* o *deshabilitar* el proceso de respaldo no concluido ...

```
1> execute sp_volchanged @session_id = 39, @devname =  
      '\\.\TAPE0',@action = 'ABORT'      (*)
```

2> y teclee doble Enter.

El sistema indicará que ha liberado dicho proceso, enviando el siguiente mensaje "return status = 0". Observe la siguiente figura:

```

C:\MSYBASE\BIN\isql.exe
Password:
1> execute sp_volchanged @session_id = 39, @devname = '\\.\TAPE0', @action = 'ABORT'
2>
(return status = 0)
1>

```

fig. 4.6.15

(\*) Observe que el "@session\_id" es el mismo que se propone en la pantalla anterior. Note además, que para volver a realizar este procedimiento es necesario reinicializar ó reciclar su cartucho.

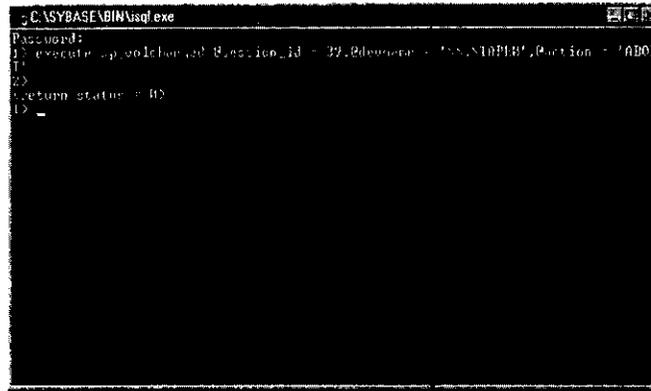
2. La segunda alternativa es continuar el proceso interrumpido de Backup en otro cartucho (cartridge) nuevo y proseguir con el respaldo. Veamos ahora que en vez de utilizar **ABORT**, emplearemos el comando **PROCEED**. Cabe señalar que debe hacer el intercambio de cintas en la Unidad de almacenamiento antes de ejecutar dicha instrucción.

- Teclee la siguiente instrucción para *continuar con el Backup* interrumpido

1>

2>execute sp\_volchanged @session\_id = 39, @devname = '\\.\TAPE0', @action = 'PROCEED' y teclee doble Enter.

El sistema indicará que ha liberado dicho proceso, con el siguiente mensaje "return status = 0". Observe la siguiente figura.



```
C:\SYBASE\BIN\sql.exe
Password:
1> execute 'spoolcharact' @action_id = 37,@device = 'SS:ST0101',@action = 'ABOL
1>
1>
1> return status = 0)
1>
1> -
```

fig. 4.6.16

### 3. Sugerencias para un respaldo óptimo

#### Frecuencia de Respaldo

La frecuencia con la que se realicen los respaldos dependerá de la carga de trabajo ó de la frecuencia en que se afecte la Base de Datos; como consecuencia de esto, va a determinar también el número de archivos de Log's de Transacciones generados por dichas actividades.

Por ejemplo, supongamos que se realizan respaldos de Nivel 0 todos los días por las mañanas, opcionalmente se tendrían que hacer respaldos de los archivos **Log** generados en las últimas horas, evitando así alguna pérdida de información generada durante el día en el caso de que llegase a presentarse alguna eventualidad.

Si por el contrario sólo se realizan respaldos una sola vez a la semana, se deberán entonces respaldar todos los Logs de una semana.

**Respaldo Incremental**

El uso del respaldo incremental minimiza la cantidad de tiempo que se necesita para el respaldo de todo el sistema. El respaldo incremental proporciona los siguientes 3 niveles de respaldo.

NIVEL	DESCRIPCION
0	Respaldo de toda la Base de Datos
1	Respaldo de los Logs que fueron generados después del último respaldo de Nivel 0
2	Respaldo de los Logs que fueron generados después del último respaldo de Nivel 1

El usuario podrá restablecer su Base de Datos usando el respaldo de Nivel 0, a lo más un respaldo de Nivel 1, y a lo más un respaldo de Nivel 2.

Por ejemplo, suponga que tiene el siguiente cuadro de respaldos, donde los números representan días consecutivos de una semana completa, es decir de Lunes a Domingo.

		<u>D I A S</u>										
		L	M	M	J	V	S	D	L	M	M	J
<b>RESP</b>	<b>Nivel 0</b>	1			4					8		
	<b>Nivel 1</b>		2				5				9	
	<b>Nivel 2</b>				3			6				10
	<b>Nivel 3</b>								7			11

Para los días 6 y 7 (sábado y domingo) hacer respaldos de Nivel 1 y 2 si así se requiriera. Finalmente el día 8, a primera hora volver a realizar el respaldo de Nivel 0 para así comenzar con otro ciclo.

Como podríamos recuperar una Base de Datos ?

Si por algún motivo desea restaurar o "bajar" la información que anteriormente fue respaldada, siga la secuencia de pasos que a continuación se muestran como sugerencia para el restablecimiento de una Base de Datos.

Veamos, por ejemplo, como podemos recuperar una Base de Datos, incluyendo los archivos de **Log's**:

En el **primer día**, hacer el Respaldo de Nivel 0 en el **cartucho #1** y Nivel 1, en el **cartucho #2**; para el **segundo día**, el Respaldo de Nivel 2 se hace también en el **cartucho #2**; y para el **tercer día**, se realiza nuevamente el Respaldo de Nivel 0, en el **cartucho #3**.

Cabe recalcar que existen varias razones por las cuales la Base de Datos se puede dañar ó corromperse. De igual manera el disco duro puede sufrir algún tipo de falla, por lo que la recuperación de la Base de Datos a partir de los cartuchos será la única alternativa para recuperarla.

Ahora bien, lo más práctico es recuperar la Base de datos a partir del respaldo de Nivel 0 contenido en el cartucho #3.

Sin embargo, sí el cartucho #3 llegará a perderse o dañarse entonces reconstruiremos la Base de Datos con el cartucho #1 el cual contiene el último respaldo de Nivel 0 y el cartucho #2 que contiene los respaldos de nivel 1 y 2. Observemos a continuación una gráfica de ésta situación:

Días	Respaldos	Información
1	Nivel 0	(**) ASADIA1HORA1 (w) <b>Cartucho #1</b>
	Nivel 1	ASADIA1HORA9 (x) <b>Cartucho #2</b>
2	Nivel 2	ASADIA2HORA1 (y) <b>Cartucho #2</b>
3	Nivel 0	ASADIA3HORA1 (z) <b>Respaldo dañado ó perdido</b>  <b>Cartucho #3</b>

(\*\*) información de la Base de Datos

Por lo tanto, la restauración sería "bajar" del día uno, el respaldo de nivel 0 y 1 y del día 2 el respaldo de nivel 2, esto nos daría la nuevamente la Base de Datos integra. ( $w + x + y = \text{Base de Datos}$ )

**Nota:** Al momento de efectuar un respaldo de la Base de Datos mediante el procedimiento `sp_dump`, se genera automáticamente un respaldo de Log de Transacciones. Estos archivos también deben ser respaldados para así poder recuperar la Base de Datos íntegramente.

## 5. SOLUCION DE PROBLEMAS

En este apartado se tomarán problemas tanto de SQL – Adaptive Server como de Replication Server. Dependiendo de la naturaleza del problema se indicará para que programa se aplica y la versión.

### 1.- Programa: SQL /Adaptive Server v11.03/11.5

#### **Problema:**

- a) *Se respalda del Log de Transacciones de una Base de Datos que se esta replicando y éste no se respalda y/o se cae el servicio de Base de Datos.*
- b) *En caso de que no se respalde el Log de Transacciones, que otra alternativa puede existir.*

#### **Solución:**

a) Probablemente el Log de Transacciones de la Base de Datos RSSD está lleno y es por eso que no se puede vaciar el Log de Transacciones de la BD que replica. Trunque el Log de la BD RSSD y si se cayó el servicio de la BD vuelva a levantarlo.

NOTA: Verifique que el servicio de Replication Server se mantenga arriba para que se no tenga problemas posteriormente.

b) En un caso de emergencia, considerando que es necesario que siga trabajando la replicación, se debe de ignorar la parte de la BS que no se ha liberado, obviamente teniendo como consecuencia que los datos se perderán. Para realizar esta tarea se efectúan los siguientes pasos:

1.- Se conecta con isql

2.- Se Introducen las siguientes instrucciones

detrane LTM o REPAGENT

dbcc settrunc( 'lrm', 'ignore' )

dbcc settrunc( 'lrm', 'valid' )

use RSSD  
rs\_zerolrm MEX\_SS, ASA  
use ASA  
dump transaction ASA with truncateonly  
levantar LTM o REPAGENT  
sp\_dump

# APÉNDICE B

## MANUAL DEL USUARIO



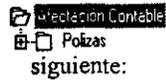
## Contabilidad

**E**l Sistema de Contabilidad contiene los datos relativos al manejo de la Contabilidad del organismo así como la posibilidad de tener e introducir al sistema las pólizas que se requieren para la elaboración de los estados financieros que son necesarios en toda empresa.

## 1. Afectación Contable

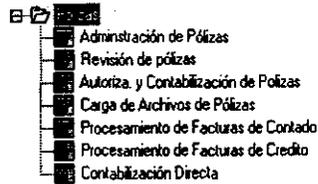
Dentro de la Afectación Contable tenemos la posibilidad de realizar las pólizas pertinentes que requerimos para el registro de nuestra contabilidad. El Sistema de Contabilidad nos permite agregar, eliminar, modificar y consultar una póliza.

El contenido de la opción Afectación Contable del Sistema de Contabilidad es el



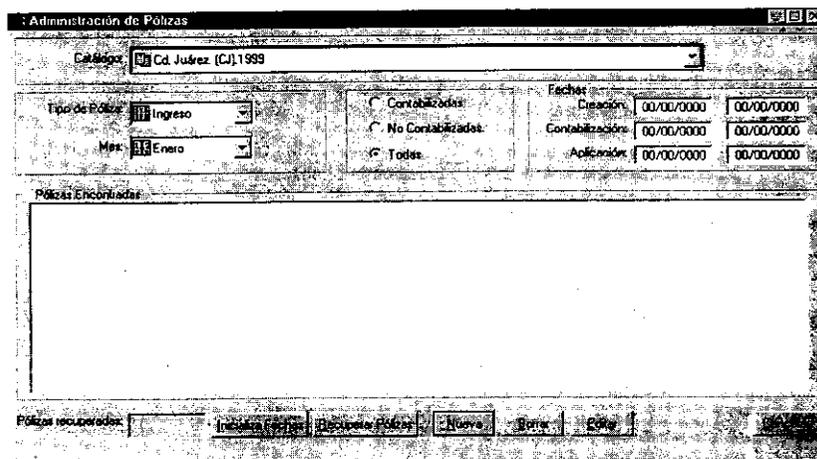
En Pólizas encontramos todas las opciones necesarias para el registro de las mismas.

El contenido de la opción Pólizas del Sistema de Contabilidad es el siguiente:



### 1.1.1 Administración de Pólizas

En esta opción el Sistema nos permite administrar todas las pólizas con las que cuenta nuestra base de datos. Vea la siguiente figura:



La ventana Administración de Pólizas contiene los siguientes elementos:

<b>Catálogo</b>	Permite seleccionar de la lista desplegable el catálogo con el que vamos a trabajar para elaborar las pólizas.	Selección
<b>Tipo de Póliza</b>	Lista desplegable con el nombre de la póliza que vamos a dar de alta, las opciones son: Ninguna, Ingreso, Egreso, Diario, Todas.	Selección
<b>Mes</b>	Presenta el mes con el cual vamos a trabajar.	Selección
<b>Contabilizadas</b>	Permite seleccionar si se requiere visualizar las pólizas que están dentro del catálogo las cuales ya han sido contabilizadas.	Selección
<b>No contabilizadas</b>	Permite seleccionar si se requiere visualizar las pólizas que están dentro del catálogo las cuales no han sido contabilizadas.	Selección
<b>Todas</b>	Permite seleccionar si se requiere visualizar todas las pólizas que están dentro del catálogo de cuentas.	Selección
<b>Creación</b>	Fecha en que se crea la póliza.	Edición
<b>Contabilización</b>	Fecha en que se contabiliza la póliza.	Edición
<b>Aplicación</b>	Fecha en que la póliza se va a aplicar.	Edición

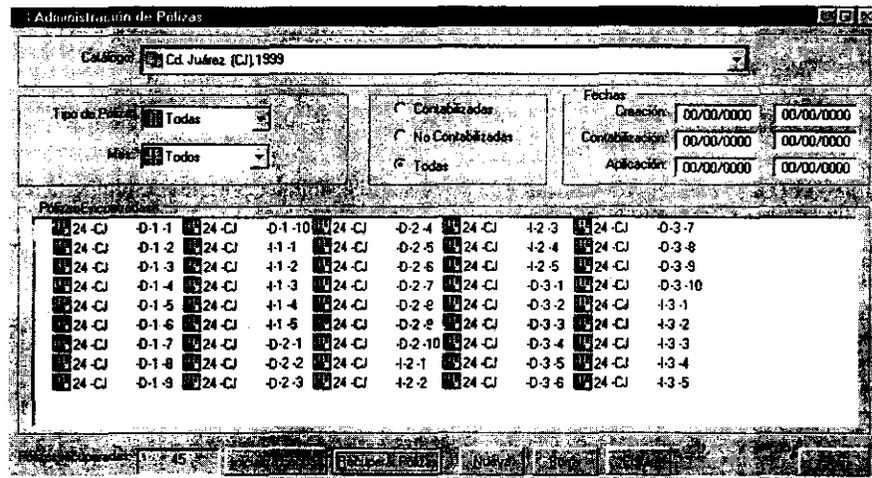
Contabilidad

<b>Pólizas Encontradas</b>	Muestra todas las pólizas que han sido encontradas en base a los parámetros introducidos.	Consulta
<b>Pólizas Recuperadas</b>	Permite ver el número de pólizas que han sido recuperadas.	Consulta
<b>Inicializa Fechas</b>	Borra el rango de fechas que ha sido introducido dando la opción de volver a editar el campo.	Selección / Edición
<b>Recuperar Pólizas</b>	Realiza el proceso de recuperar pólizas que están dentro de nuestra contabilidad.	Selección
<b>Nueva</b>	Inicia el proceso de dar de alta una nueva póliza.	Selección
<b>Borrar</b>	Realiza el proceso de borrar una póliza que previamente a sido dada de alta.	Selección
<b>Editar</b>	Permite hacer modificaciones en una póliza que ya ha sido capturada.	Selección
<b>Cancelar</b>	Aborta el proceso de la administración de pólizas, es decir, cierra la ventana actual.	Selección

• Propiedades de la Administración de Pólizas

Pasos a seguir:

1. Haga doble *click* en Administración de Pólizas. El sistema presentará automáticamente la ventana del mismo nombre.
2. Haga *click* con el botón derecho del mouse en el área de Pólizas Encontradas de la ventana y seleccione una de las dos opciones que aparecen en ese menú, las opciones son: Ver y Ordenado por.
3. Haga *click* en Ver y seleccione una de las dos opciones de esa ventana Lista o Detalle.
4. En caso de seleccionar la opción Lista, el sistema presentará una lista completa de todas las pólizas encontradas.
5. Si usted selecciona Detalle, el sistema presentará los detalles de cada una de las pólizas encontradas. Vea la siguiente figura:



6. La ventana le muestra los siguientes campos: Identificación ( la estructura de este campo es la siguiente: Catálogo al que se aplica – Area del catálogo – Tipo de póliza (que puede ser de Diario, Ingresos, Egresos o Unica) – Mes – Número de póliza.), Fecha de creación, Número de Póliza, Fecha de aplicación, Fecha de contabilización, Clave del área generadora, Concepto, Contabilizada, Elaboró, Autorizó, Revisó.
7. Haga clic en Ordenar por, el sistema le presenta las siguientes opciones: by Identificación, by Fec. de creación, by No. de Póliza, by Fec. Aplicación, by Fec. de Contabilización, by Cve. Area Generad, by Concepto, by Elaboró, by Autorizó, by Revisó.
8. Seleccione la opción que desee para que las pólizas sean ordenadas en base al campo que eligió.

• Registro de una Póliza

Pasos a seguir:

1. Haga doble *click* en Administración de Pólizas. El sistema presentará automáticamente la ventana del mismo nombre.
2. Haga *click* en el botón Nueva o bien utilice el método abreviado *Alt + N*. El sistema presentará la siguiente ventana:

La ventana Nueva Póliza contiene los siguientes elementos:

Elemento	Descripción	Modo de Consulta
Aeropuerto	Muestra el designador del aeropuerto en donde se esta trabajando. El sistema lo presenta automáticamente.	Consulta
Campo sin nombre	Muestra el nombre completo del aeropuerto donde se esta trabajando. El sistema lo presenta automáticamente.	Consulta
Mes	Número y nombre completo del mes de trabajo. El sistema lo presenta automáticamente.	Consulta
Area Generadora	Clave y nombre del área en donde se genera la póliza. El sistema la presenta automáticamente.	Consulta
Clave del Usuario	Muestra la clave del usuario del sistema. Esta clave se presenta automáticamente.	Consulta
Clave del	Presenta la clave y nombre del empleado.	Consulta

<b>Empleado</b>		
<b>Tipo de Póliza</b>	Nos permite seleccionar el tipo de póliza a dar de alta, las opciones son: <b>Ingresos, Egresos, Diario</b> o <b>Ninguno</b> .	Selección
<b>Fecha de Creación</b>	Nos muestra la fecha en que se crea la cuenta, el sistema registra automáticamente la fecha del día en que se está trabajando.	Consulta
<b>Contabilización</b>	Fecha en que se requiere que la póliza sea contabilizada, el sistema nos presenta automáticamente la fecha actual de trabajo.	Edición
<b>Aplicación</b>	Fecha de aplicación de la póliza.	Consulta
<b>Número de Póliza</b>	Muestra el número de la nueva póliza, el sistema lo presenta automáticamente.	Consulta
<b>Concepto de la Póliza</b>	Permite introducir el concepto de la nueva póliza.	Edición
<b>Sec</b>	Números secuenciales de movimientos que estamos introduciendo, este campo, muestra una flecha que nos permite ubicar en que línea estamos trabajando.	Edición
<b>C. de Costos</b>	Muestra el número del centro de costos con el cual se esta trabajando.	Consulta
<b>Cuenta Contable</b>	Permite introducir el número de cuenta al cual se le va a carga o abonar cierta cantidad dentro de la póliza.	Edición / Selección
<b>Concepto</b>	Concepto por el cual la cuenta se va a afectar.	Edición
<b>Cheque / Refer</b>	Permite introducir si el cargo o abono que se esta haciendo tiene como referencia un cheque y en este campo puede poner el número de cheque o referencia.	Selección.
<b>Debe</b>	Cantidad de cargo para la cuenta introducida.	Edición
<b>Haber</b>	Cantidad de abono para la cuenta introducida.	Edición
<b>Cif. Control</b>	Suma de todas las cuentas contables de la póliza.	Consulta
<b>No. Movtos.</b>	Muestra el número de movimientos que tiene la póliza.	Consulta
<b>Diferencia</b>	Muestra las diferencias ( en caso de que existan ) entre los movimientos del debe y haber de la póliza.	Consulta

Adicionalmente, en la parte superior de la ventana, tenemos un conjunto de botones que

llamamos Barra de Herramientas la cual contiene las siguientes opciones:

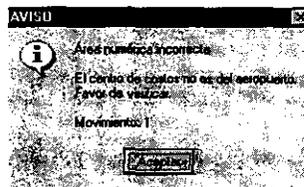


1 2 3

- 1.- Guardar. Permite guardar un registro nuevo o alguna modificación agregada.
- 2.- Agregar Registro. Permite agregar un nuevo registro o póliza.
- 3.- Borrar Registro. Elimina un registro o póliza una vez seleccionado.

Lo anterior, estará presente en el proceso de Agregar, Modificar y Eliminar Pólizas.

3. Seleccione el tipo de póliza a dar de alta en el campo Tipo de Póliza las opciones a elegir son: Ingresos, Egresos, Diario, Unica o Cancelación.
4. Cabe mencionar, que el tipo de póliza Cancelación, es usada únicamente al final del año contable para crear los movimientos de cancelación para las cuentas de resultados.
5. Introduzca la fecha en que desee que la póliza sea contabilizada en el campo Contabilización. El sistema presenta automáticamente la fecha del día en que esta trabajando.
6. Introduzca el concepto de la póliza en el campo del mismo nombre.
7. Seleccione el centro de costos en donde va a trabajar en la lista desplegable que aparece al lado derecho del campo C de Costos, o bien, si en el aeropuerto donde esta trabajando, sólo hay un centro de costos, el sistema lo pondrá automáticamente.
8. En caso de que introduzca un centro de costos no válido para el sistema, se desplegará un mensaje como el siguiente:



9. Oprima el botón Aceptar e introduzca un centro de costos válido para el aeropuerto.
10. Introduzca el número de cuenta en que va a realizar los movimientos necesarios en el campo Cuenta Contable, cabe mencionar que en ese campo no es posible modificar nada, sino que en la parte de arriba de la ventana aparecen tres pequeños cuadros en gris los cuales al seleccionar el campo en cuestión toman un color blanco.
11. En los recuadros superiores, introduzca el número de cuenta o bien presione *Ctrl + F1* y el sistema presentará el catálogo de cuentas para que se seleccione la cuenta deseada. Vea la siguiente figura:

Catálogo de Cuentas Contables

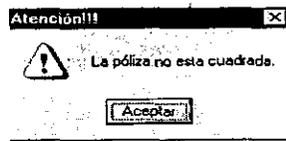
Campo:

Criterio de Selección:

Cuenta	Descripción de la Cuenta
1000000000	ACTIVO CIRCULANTE
1010000000	DISPONIBLE
1020000000	INVERSIONES EN VALORES
1030000000	CUENTAS POR COBRAR
1040000000	OTRAS CUENTAS POR COBRAR
1050000000	ESTIMACION CTAS DE COBRO DUDOSO
1060000000	ANTICIPOS
1070000000	IVA ACREDITABLE
1080000000	INVENTARIOS

Aceptar

12. Introduzca el concepto del movimiento de la cuenta que esta afectando en el campo **Concepto**, si el concepto es el mismo que el de la póliza presione **CTRL + F3** para repetir el texto.
13. En caso de existir una referencia o un cheque con el cual se soporte el movimiento de la cuenta, introduzca el número correspondiente en el campo **Cheque / Refer.**
14. Introduzca la cantidad de cargo de la cuenta seleccionada en el campo **Debe**.
15. Introduzca la cantidad de abono de la cuenta seleccionada en el campo **Haber**.
16. En el campo **Sec.** podrá observar que conforme va capturando movimientos, el sistema va colocando un número secuencial a la línea, así como también la flecha va pasando de un renglón a otro.
17. Así mismo, si usted desea insertar una línea entre las que ya capturó:
  - 17a. Coloque el cursor en el campo **Concepto** y haga *click* con el botón derecho del mouse y seleccione la opción **Agregar**, o bien, en la barra de herramientas haga *click* en el botón **Agregar**, también puede hacer *click* en **Edición del Panel de control** y seleccionar la opción **Agregar Registro**. El sistema insertará una línea en la parte superior de la línea donde esta colocado actualmente.
  - 17b. Coloque el cursor en el campo **Debe** o en el campo **Haber** y haga *click* con el botón derecho del mouse y seleccione la opción **Agregar**, o bien, en la barra de herramientas haga *click* en el botón **Agregar**, también puede hacer *click* en **Edición del Panel de control** y seleccionar la opción **Agregar Registro**. El sistema insertará una línea en la parte inferior de la línea donde esta colocado actualmente.
18. En caso de que el sistema detecte que la sumatoria de cargos y la de abonos no son iguales, se desplegará el siguiente mensaje:



Haga clic en el botón Aceptar y verifique las cantidades introducidas.

19. Guarde los cambios realizados haciendo clic en el botón Guardar de la barra de herramientas o seleccionando del menú la opción Guardar o bien utilice el método abreviado Ctrl + G.

- Modificación de una Póliza

Pasos a seguir:

1. Haga doble clic en Administración de Pólizas. El sistema presentará automáticamente la ventana del mismo nombre.
2. Haga clic en el botón Editar o utilice el método abreviado Alt + E y el sistema le presentará la siguiente ventana.

Edición de Pólizas

Aeropuerto: MEX Internacional De La Ciudad De Mexico Mes: 2 Febrero

Alia Generada: A MEX AEROPUERTO INTERNACIONAL DE LA CIUDAD DE MEXICO, S.A. DE C.V.

Nombre de Usuario: Roberto Morán

Tipo de Póliza: OTRAS Fecha de Emisión: 07/10/1998 Contabilización: 28/02/1998 Aplicación: 00/00/0000

Número de Póliza: 23 Concepto de Póliza: POLIZAS A I C M

201 PASIVO CIRCULANTE  
 012 CORRECCION POR REEXPRISION  
 0003S PATRIMONIO AEROPUERTO

Año	Csa	Cuenta	Concepto	Debe	Haber
128		201-012-00039	DETERM REPOMO INSTIT	54 710 400.00	
128		405-002-00001	<<ERROR>> DETERM REPOMO INSTIT		54 710 400.00
128		605-001-00001	CANA PARC POL MEX 01031		2 003.68
128		603-001-00001	CANA PARC POL MEX 01031		-3 000.00
128		604-001-00001	CANA PARC POL MEX 01031		909 697.04
128		703-001-00006	CANA PARC POL MEX 01031		11 696 711.79
128		703-001-00005	CANA PARC POL MEX 01031		17 283 591.17
128		302-005-00058	CANA PARC POL MEX 01031		-23 898 003.68
128		132-007-00001	TR COMB A OFNAS GRALES	1 962 304.00	
128		134-006-00001	TR COMB A OFNAS GRALES	45 426.00	

El Contol: 672109705397 No. Movtos: 0 Diferencia: 0.00 52,702,670.00 52,702,670.00

3. Haga clic sobre el campo que desee modificar e introduzca el nuevo valor para ese campo. Cabe mencionar que una póliza podrá ser modificada, únicamente si esta no ha sido contabilizada, en caso contrario, el sistema no le permitirá cambiar ningún campo.
4. Guarde los cambios realizados haciendo clic en el botón Guardar de la barra de

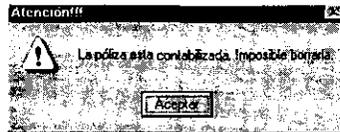
herramientas o seleccionando del menú la opción **Guardar** o bien utilice el método abreviado *Ctrl + G*.

---

- Eliminación de una Póliza

Pasos a seguir:

1. Haga doble *clik* en **Administración de Pólizas**. El sistema presentará automáticamente la ventana del mismo nombre.
2. Haga *clik* sobre la póliza que desee borrar. Cabe mencionar que sólo podrá eliminar una póliza si esta no esta contabilizada, en caso contrario el sistema presentará el siguiente mensaje.



3. Si la póliza no esta contabilizada, haga *clik* en el botón **Borrar** o utilice el método abreviado *Alt + B*, o bien haga clic en el botón **Borrar** de la barra de herramientas. Como confirmación de la operación requerida, el sistema presentará el siguiente mensaje:



Si esta seguro que desea eliminar la póliza, haga *clik* en **Aceptar**, de lo contrario oprima el botón **Cancelar**.

---

- Elaboración de un Reporte de Pólizas

Pasos a seguir:

1. Haga doble *clik* en **Administración de Pólizas**. Automáticamente el sistema presentará la ventana del mismo nombre.

2. Seleccione la póliza a imprimir haciendo clic en la columna Identificación de la ventana y presione el botón Editar, el sistema le presentará la póliza que usted eligió.
3. Oprima el botón **Presentación Preliminar** de la barra de herramientas y el sistema le mostrará la presentación preliminar del reporte. Vea la siguiente figura:

The screenshot shows a window titled 'Sistema ASA' containing a report titled 'SISTEMA DE CONTABILIDAD PÓLIZA CONTABLE'. The report header includes 'Aeropuerto: ZLO' and 'Playa De Oro'. Below this, it specifies 'Tipo de Póliza: DIARIO', 'No. de Póliza: 1', 'Área Generadora: PF', and 'AEROPUERTO DE MANZANILLO'. The 'Concepto' is 'BALANZA DE COMPR MZO 98 ZLO' and the 'Creación' date is '10/20/1999'. The main body of the report is a table with columns for 'Área', 'Cuenta', 'Descripción de la Cuenta', 'Cheque', 'Concepto del Movimiento', and a numerical value.

Área	Cuenta	Descripción de la Cuenta	Cheque	Concepto del Movimiento	
21	101-003-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	548.1
21	101-003-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	
21	104-005-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	
21	107-001-00001	IVA ACREDITABLE TASA 15%		BALANZA DE COMPR MZO 98 ZLO	46.1
21	107-001-00001	IVA ACREDITABLE TASA 15%		BALANZA DE COMPR MZO 98 ZLO	
21	111-001-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	61.1
21	111-001-21001	AEROPUERTO MANZANILLO		BALANZA DE COMPR MZO 98 ZLO	
21	201-005-00014	2 ALMILLAR CAMINALHO CONSTRUCCION		BALANZA DE COMPR MZO 98 ZLO	

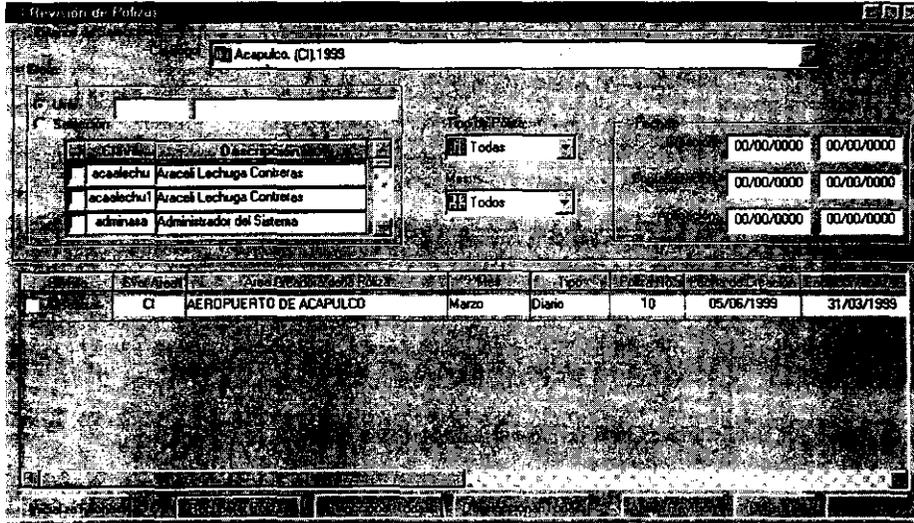
- Impresión del Reporte de Pólizas

Paso a seguir:

1. Haga clic en el botón **Imprimir** de la barra de herramientas o en el botón **Impresión Directa** de la misma o bien seleccione del menú principal la opción **Imprimir** o utilice el método abreviado **Alt + P**.

### 1.1.2 Revisión de Pólizas

En esta opción el Sistema nos permite revisar todas las pólizas con las que cuenta nuestra base de datos, cabe mencionar que el usuario que proceda a la revisión deberá ser un usuario distinto del que capturó la póliza. Vea la siguiente figura:



La ventana Revisión de Pólizas contiene los siguientes elementos:

<b>Catálogo</b>	Permite seleccionar de la lista desplegable el catálogo con el que vamos a trabajar para elaborar las pólizas.	Selección
<b>Uno</b>	Permite introducir el nombre o clave del usuario del que deseamos ver sus pólizas.	Selección
<b>Selección</b>	Permite seleccionar varios usuarios de los que deseamos ver sus pólizas.	Selección
<b>Tipo de Póliza</b>	Lista desplegable con el nombre de la póliza que vamos a dar de alta, las opciones son: Ninguna, Ingreso, Egreso, Diario, Todas.	Selección
<b>Mes</b>	Presenta el mes con en el cual vamos a trabajar.	Selección

Contabilidad

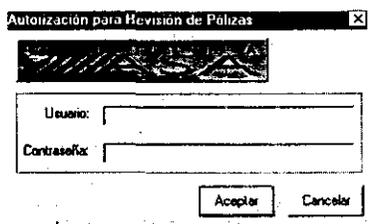
<b>Creación</b>	Fecha en que se crea la póliza.	Edición
<b>Contabilización</b>	Fecha en que se contabiliza la póliza.	Edición
<b>Aplicación</b>	Fecha en que la póliza se va a aplicar.	Edición
<b>Revisa</b>	Permite seleccionar las pólizas a revisar.	Selección
<b>Cve. Area</b>	Muestra la clave del área de ubicación de la póliza.	Consulta
<b>Area Creadora de la Póliza</b>	Muestra el área donde fue creada la póliza.	Consulta
<b>Mes</b>	Muestra el mes en que fue hecha la póliza.	Consulta
<b>Tipo</b>	Muestra el tipo de póliza, puede ser de <b>Ingresos, Egresos, Diario o Unica.</b>	Consulta
<b>Póliza No.</b>	Muestra el número consecutivo asignado a la póliza en base a su tipo.	Consulta
<b>Fecha de Creación</b>	Muestra la fecha de creación de la póliza.	Consulta
<b>Fec. Contabilización</b>	Muestra la fecha de contabilización de la póliza.	Consulta
<b>Concepto</b>	Muestra el concepto de la póliza.	Consulta
<b>Núm. Movtos.</b>	Muestra el número de movimientos que contiene la póliza.	Consulta
<b>Total Debe</b>	Cantidad total de movimientos de cargos dentro de la póliza.	Consulta
<b>Total Haber</b>	Cantidad total de movimientos de abonos dentro de la póliza.	Consulta
<b>Cifras de Control</b>	Cifras de control asignada a la póliza.	Consulta
<b>Elaboró</b>	Usuario que elaboró la póliza.	Consulta
<b>Inicializa Fechas</b>	Borra el rango de fechas que ha sido introducido dando la opción de volver a editar el campo.	Selección / Edición
<b>Recuperar Pólizas</b>	Realiza el proceso de recuperar pólizas que están dentro de nuestra contabilidad.	Selección
<b>Seleccionar Todas</b>	Selecciona todas las pólizas que se encuentran dentro del catálogo actual.	Selección
<b>Deseleccionar Todas</b>	Elimina la selección de todas las pólizas.	Selección
<b>Aplica Revisión</b>	Inicia el proceso de revisión a las pólizas seleccionadas.	Selección
<b>Consultar</b>	Permite ver el contenido de la póliza seleccionada..	Selección
<b>Terminar</b>	Cierra la ventana <b>Revisión de Pólizas</b> y nos regresa al menú principal.	Selección

---

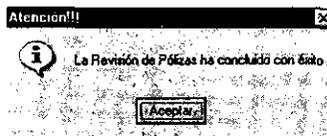
- Revisión de una Póliza

Pasos a seguir:

20. Haga doble *clic* en *Revisión de Pólizas*. El sistema presentará automáticamente la ventana del mismo nombre.
21. Haga *clic* en el campo *Catálogo* y seleccione el mismo de la lista desplegable que aparece del lado derecho.
22. Seleccione el campo *Uno* en caso de que quiera ver las pólizas de un sólo usuario, en caso contrario elija el campo *Selección* para solicitar la lista de usuarios de los cuales desea revisar sus pólizas.
23. Seleccione el tipo de pólizas que desea ver en el campo *Tipo*, estas pueden ser de *Diario, Ingresos, Egresos o Unicas*.
24. Seleccione el mes que desee consultar en el campo *Mes*.
25. Introduzca las fechas de creación, contabilización y aplicación del periodo que desee en los campos del mismo nombre.
26. Haga clic en el botón *Recuperar Pólizas* y el sistema le presentará en pantalla las pólizas encontradas basándose en los criterios señalados.
27. Seleccione en el campo *Revisa la (s) póliza (s) a revisar*.
28. Haga clic en el botón *Aplica Revisión*, el sistema le presentará la siguiente ventana:



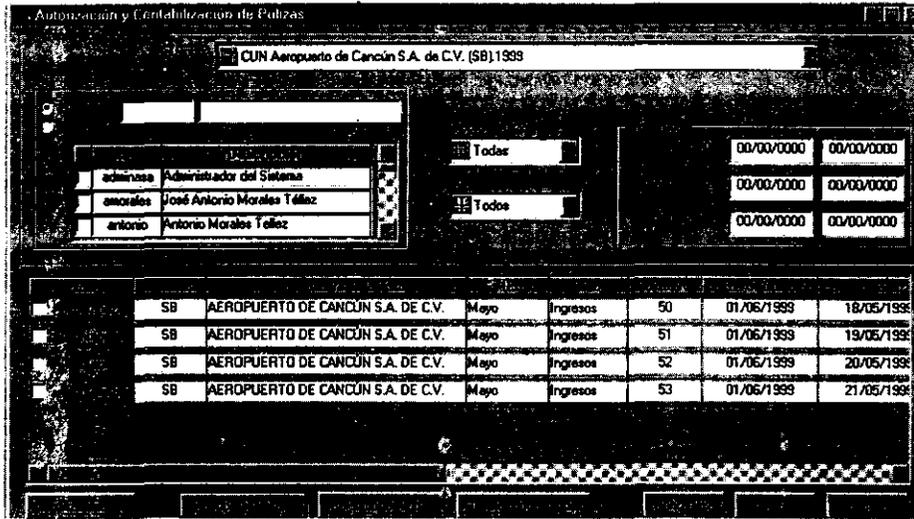
29. Introduzca el nombre del usuario en el campo Usuario y la Contraseña del mismo y haga clic en el botón Aceptar. Lo anterior es una medida de seguridad para proceder a la revisión de las pólizas.
30. Una vez concluido el proceso de revisión, el sistema le presentará el siguiente mensaje:



31. Haga clic en el botón Aceptar y el proceso habrá concluido.

### 1.1.3 Autorización y Contabilización de Pólizas

En esta opción el Sistema nos permite autorizar y contabilizar todas las pólizas con las que cuenta nuestra base de datos, cabe mencionar que el usuario que proceda a la autorización y contabilización deberá ser un usuario distinto del que revisó la póliza. Vea la siguiente figura:



La ventana Autorización y Contabilización de Pólizas contiene los siguientes elementos:

<b>Catálogo</b>	Permite seleccionar de la lista desplegable el catálogo con el que vamos a trabajar para elaborar las pólizas.	Selección
<b>Uno</b>	Permite introducir el nombre o clave del usuario del que deseamos ver sus pólizas.	Selección
<b>Selección</b>	Permite seleccionar varios usuarios de los que deseamos ver sus pólizas.	Selección
<b>Tipo de Póliza</b>	Lista desplegable con el nombre de la póliza que vamos a dar de alta, las opciones son: Ninguna, Ingreso, Egreso, Diario, Todas.	Selección
<b>Mes</b>	Presenta el mes con el cual vamos a trabajar.	Selección
<b>Creación</b>	Fecha en que se crea la póliza.	Edición

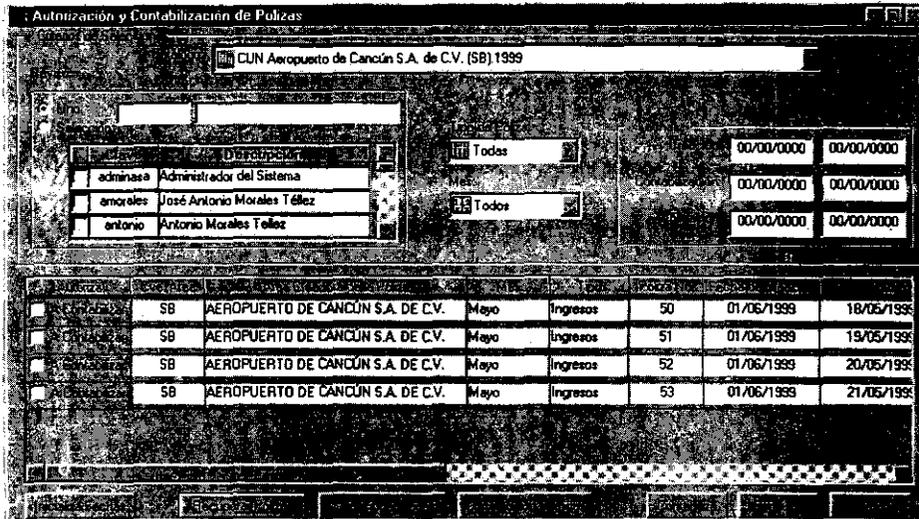
Contabilidad

<b>Contabilización</b>	Fecha en que se contabiliza la póliza.	Edición
<b>Aplicación</b>	Fecha en que la póliza se va a aplicar.	Edición
<b>Revisa</b>	Permite seleccionar las pólizas a revisar.	Selección
<b>Cve. Area</b>	Muestra la clave del área de ubicación de la póliza.	Consulta
<b>Area Creadora de la Póliza</b>	Muestra el área donde fue creada la póliza.	Consulta
<b>Mes</b>	Muestra el mes en que fue hecha la póliza.	Consulta
<b>Tipo</b>	Muestra el tipo de póliza, puede ser de Ingresos, Egresos, Diario o Unica.	Consulta
<b>Póliza No.</b>	Muestra el número consecutivo asignado a la póliza en base a su tipo.	Consulta
<b>Fecha de Creación</b>	Muestra la fecha de creación de la póliza.	Consulta
<b>Fec. Contabilización</b>	Muestra la fecha de contabilización de la póliza.	Consulta
<b>Concepto</b>	Muestra el concepto de la póliza.	Consulta
<b>Núm. Movtos.</b>	Muestra el número de movimientos que contiene la póliza.	Consulta
<b>Total Debe</b>	Cantidad total de movimientos de cargos dentro de la póliza.	Consulta
<b>Total Haber</b>	Cantidad total de movimientos de abonos dentro de la póliza.	Consulta
<b>Cifras de Control</b>	Cifras de control asignada a la póliza.	Consulta
<b>Elaboró</b>	Usuario que elaboró la póliza.	Consulta
<b>Revisó</b>	Clave del usuario que revisó la póliza.	Consulta
<b>Inicializa Fechas</b>	Borra el rango de fechas que ha sido introducido dando la opción de volver a editar el campo.	Selección / Edición
<b>Recuperar Pólizas</b>	Realiza el proceso de recuperar pólizas que están dentro de nuestra contabilidad.	Selección
<b>Seleccionar Todas</b>	Selecciona todas las pólizas que se encuentran dentro del catálogo actual.	Selección
<b>Deseleccionar Todas</b>	Elimina la selección de todas las pólizas.	Selección
<b>Contabiliza</b>	Inicia el proceso de contabilización a las pólizas seleccionadas.	Selección
<b>Consultar</b>	Permite ver el contenido de la póliza seleccionada.	Selección
<b>Terminar</b>	Cierra la ventana Revisión de Pólizas y nos regresa al menú principal.	Selección

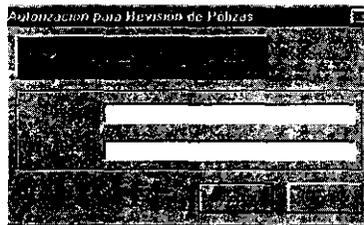
- Autorización y Contabilización de una Póliza

Pasos a seguir:

32. Haga doble *click* en Autorización y Contabilización de Pólizas. El sistema presentará automáticamente la ventana del mismo nombre. Vea la siguiente figura:



33. Haga clic en el campo Catálogo y seleccione el mismo de la lista desplegable que aparece del lado derecho.
34. Haga clic en el campo Area Creadora de la Póliza y seleccione la misma de la lista desplegable que aparece del lado derecho.
35. Seleccione el campo Uno en caso de que quiera ver las pólizas de un sólo usuario, en caso contrario elija el campo Selección para solicitar la lista de usuarios de los cuales desea revisar sus pólizas.
36. Seleccione el tipo de pólizas que desea ver en el campo Tipo, estas pueden ser de Diario, Ingresos, Egresos o Unicas.
37. Seleccione el mes que desee consultar en el campo Mes.
38. Introduzca las fechas de creación, contabilización y aplicación del periodo que desee en los campos del mismo nombre.
39. Haga clic en el botón Recuperar Pólizas y el sistema le presentará en pantalla las pólizas encontradas basándose en los criterios señalados.
40. Seleccione en el campo Revisa la (s) póliza (s) a contabilizar, cabe mencionar que las pólizas que vemos en esta ventana, son únicamente aquellas que ya han sido revisadas.
41. Haga clic en el botón Contabiliza, el sistema le presentará la siguiente ventana:



42. Introduzca el nombre del usuario en el campo Usuario y la Contraseña del mismo y haga clic en el botón Aceptar. Lo anterior es una medida de seguridad para proceder a la actualización y contabilización de las pólizas.
43. Una vez concluido el proceso de revisión, el sistema le presentará el siguiente mensaje:

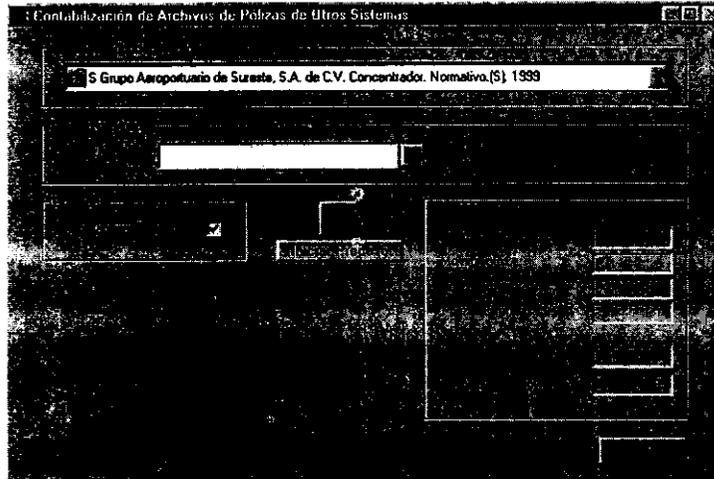


44. Haga clic en el botón Aceptar y el proceso habrá concluido.

### 1.1.4. Carga de Archivos de Pólizas

En Afectación Contable tenemos la posibilidad de cargar archivos de pólizas externos al sistema.

Esta opción del Sistema nos presenta la siguiente figura:



La ventana **Contabilización de Archivos de Pólizas de Otros Sistemas** contiene los siguientes elementos:

<b>Catálogo a donde se aplican las pólizas</b>	Muestra el nombre del Catálogo al que se van a cargar las pólizas que van a ser importadas desde algún archivo de texto.	Selección
<b>Archivo de Pólizas</b>	Muestra la ruta de donde vamos a importar el archivo de pólizas.	Edición / Selección
<b>Contabilizar Pólizas</b>	Permite seleccionar si deseamos contabilizar las pólizas.	Selección
<b>Iniciar Proceso</b>	Procede a iniciar e proceso de importación del catálogo de pólizas desde un archivo externo.	Selección

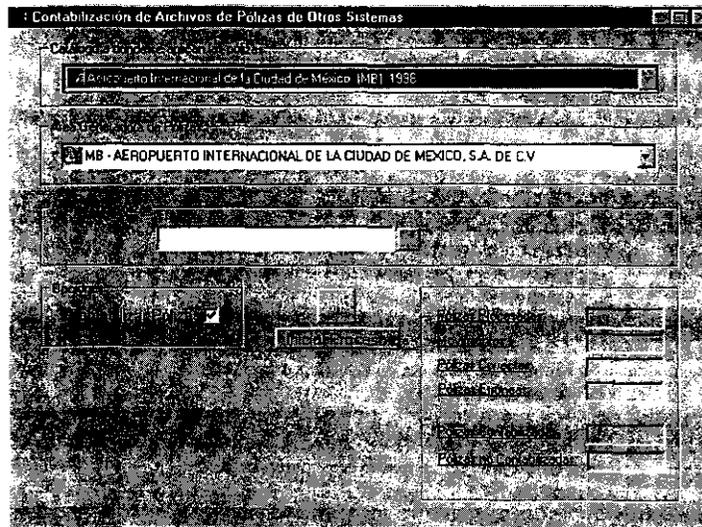
## Contabilidad

<b>Pólizas Procesadas</b>	Muestra el total de pólizas procesadas.	Consulta
<b>Movimientos</b>	Muestra el total de movimientos de cada póliza procesados.	Consulta
<b>Pólizas Correctas</b>	Muestra el total de pólizas que han sido procesadas correctamente.	Consulta
<b>Pólizas Erróneas</b>	Muestra el total de pólizas que han sido procesadas de manera incorrecta.	Consulta
<b>Pólizas Contabilizadas</b>	Muestra el número total de pólizas contabilizadas.	Consulta
<b>Pólizas no Contabilizadas</b>	Muestra el número total de pólizas no contabilizadas.	Consulta

- **Carga de un Archivo de Pólizas**

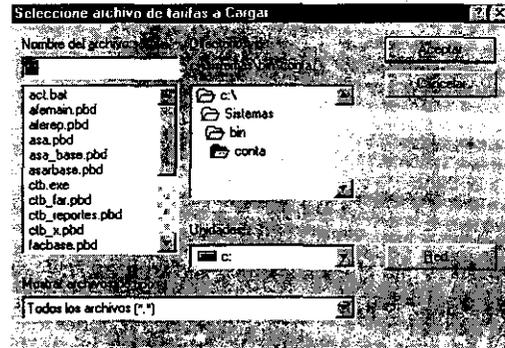
**Pasos a seguir:**

1. Haga doble *click* en **Carga de Archivos de Pólizas**. El sistema presentará automáticamente la ventana **Carga de Archivos de Pólizas de Otros Sistemas**. Vea la siguiente figura:



2. Haga *click* en el campo **Catálogo** a donde se aplican las pólizas y seleccione de la lista desplegable el nombre del catálogo a cargar.

3. Haga *click* en el botón derecho del campo Archivo de pólizas, el Sistema le presentará la siguiente ventana:



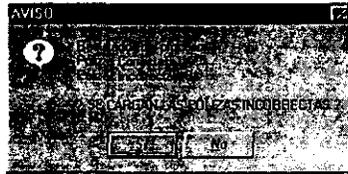
4. Escriba el nombre del archivo de texto de donde se van a importar las pólizas y seleccione la ruta de acceso al mismo, en caso de ser un archivo en un equipo que este dentro de la red haga *click* en el botón Red y seleccione la ruta deseada.
5. Seleccione si desea **Contabilizar Pólizas** en el campo del mismo nombre.
6. Haga *click* en el botón **Iniciar Proceso** para que el sistema proceda a la recuperación de las pólizas y espere a que el proceso termine. El sistema presentará un mensaje como el siguiente:



7. Haga *click* en el botón **Si** en caso de que quiera cargar la cantidad de pólizas que nos reporta el sistema como correctas, en caso contrario oprima el botón **No**.
8. El sistema presentará también la siguiente ventana:



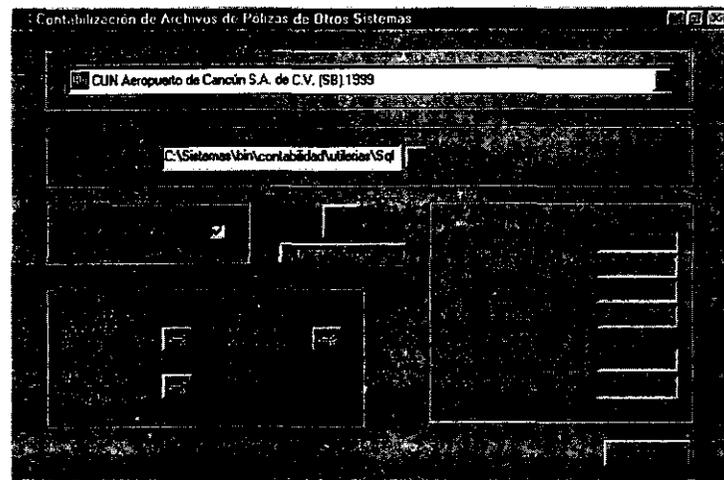
9. Oprima el botón Si en caso de que desee que las pólizas que están correctas sean contabilizadas, en caso contrario, presione el botón No.
10. Adicionalmente se presentará el siguiente mensaje:



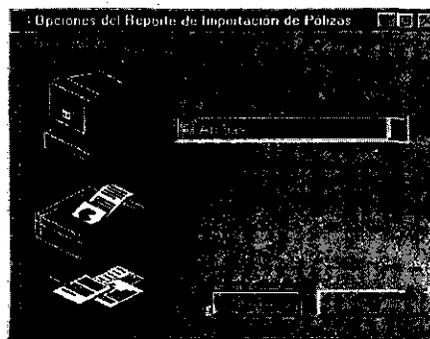
11. Haga clic en Si si desea que las pólizas incorrectas encontradas sean cargadas, en caso contrario presione el botón No.
12. Si el sistema no localiza ninguna póliza errónea los mensajes anteriores no serán presentados en pantalla y el proceso de contabilización ( en caso de haber sido seleccionado ) seguirá su proceso hasta finalizar.
13. Cuando el proceso quede completamente terminado el sistema presentará la siguiente ventana:



14. Haga clic en el botón Aceptar y observe la siguiente figura:



15. Note que la ventana presenta nuevas opciones en la parte inferior izquierda, en esta área es donde podemos imprimir todo lo que requiramos de los archivos de pólizas recién cargados.
16. La opción Resultados de la Importación (Detalle) presenta la siguiente ventana:



17. De la lista desplegable que aparece del lado derecho del campo Pólizas a Visualizar elija una de las opciones Ambas, Correctas o Incorrectas.
18. Haga clic en Aceptar y el sistema le desplegará la Presentación Preliminar del reporte. Vea la siguiente figura.

Sistema ASA

**SISTEMA DE CONTABILIDAD**  
**DETALLE DEL PROCESAMIENTO DE LOS ARCHIVOS DE PÓLIZAS**

Aeropuerto: **MEX** Internacional De La Ciudad De Mexico

**ESTADO DE LAS PÓLIZAS: INCORRECTAS**

**ESTADO CONTABLE: NO CONTABILIZADAS**

Tipo de Póliza: **DIARIO**

Tipo de Póliza: **DIARIO** No. de Póliza Archivo: **12001** No. de Póliza ASA: **2**  
 Área Generadora: **ME** AEROPUERTO INTERNACIONAL DE LA CIUDAD DE MEXICO, S.A. DE C.V.  
 Concepto: **COBZA 31-12-97 012001** Creación: **10/17/1998**  
 Observaciones: Diferencia entre SUMA DE CARROS y SUMA DE ABONOS

Cuenta	Descripción de la Cuenta	Concepto del Movimiento	Debe	Haber
10100201002	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ES: 00	10,000.00	0.00
10100301001	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ES: 00	0.00	111,038.13
10100301002	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ES: 00	175,848.21	0.00
10200101001	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ES: 00	10,072.29	0.00
10400210787	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ES: 00	0.00	11,850.30
10400211997	No existe cuenta en catálogo	<<ERROR>> 0000000 SALDOS INICIAL ES: 00	769.00	0.00

19. La opción Resultados de la Importación ( Concentrado ) muestra una *Presentación Preliminar* del reporte. Vea la siguiente ventana:

Sistema ASA

**RESUMEN DEL PROCESO DE IMPORTACIÓN**

Correctas/Incorrectas	Contabilizadas / No contabilizadas	ID	Gran Total
Incorrectas	No Contabilizadas	3	3
<b>Total Incorrectas</b>		<b>3</b>	<b>3</b>
Correctas	Contabilizadas	1	1
<b>Total Correctas</b>		<b>1</b>	<b>1</b>
<b>Gran Total</b>		<b>4</b>	<b>4</b>

20. La opción Pólizas no Contabilizadas nos muestra una Presentación Preliminar de las mismas.
21. Cabe mencionar que todos estos reportes pueden ser impresos como se describió en la parte de Reportes de Pólizas.

22. Es importante mencionar que todos los reportes que requiera de la importación deberán ser impresos antes de cerrar la ventana Carga de Archivos de Pólizas de Otros Sistemas ya que después no le será posible.

## 2. Cierre Contable

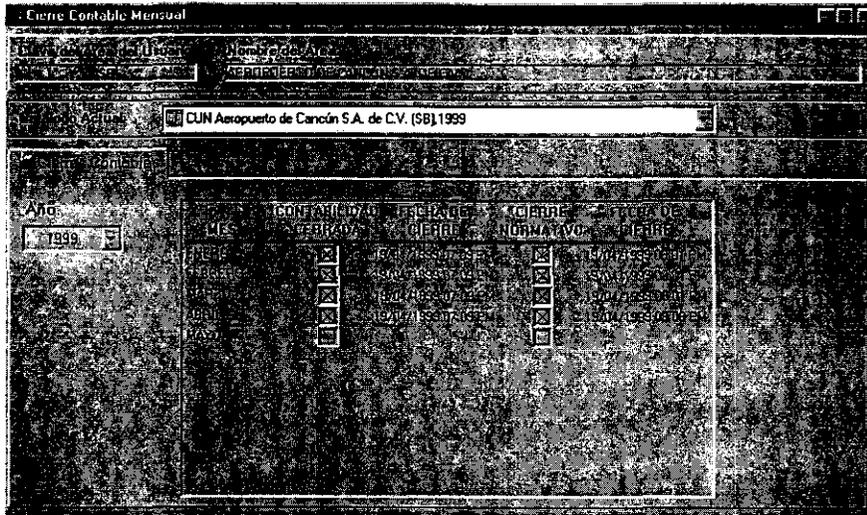
Dentro de la Contabilidad de una empresa, requerimos de hacer Cierres contables para poder observar los resultados de la misma y ver el comportamiento de nuestros recursos. El Sistema de Contabilidad provee de una opción para realizar este proceso.

El contenido de la opción Cierre Contable es el siguiente:



### 2.1 Cierre contable Mensual

La opción **Cierre Contable Mensual** nos presenta la siguiente ventana:



La ventana **Cierre Contable Mensual** contiene los siguientes elementos:

	Descripción	Acción
<b>Clave del Area del Usuario</b>	Muestra la clave del área de trabajo del usuario.	Consulta
<b>Nombre del Area</b>	Nombre completo del área de trabajo donde se encuentra el usuario.	Consulta

<b>Catálogo Actual</b>	Permite seleccionar el catálogo con el cual vamos a trabajar.	Selección
<b>Año</b>	Permite seleccionar el año en el cual vamos a trabajar para iniciar nuestro cierre contable.	Selección
<b>Mes</b>	Muestra el mes al cual vamos a aplicar el cierre contable.	Consulta
<b>Contabilidad Cerrada</b>	Permite seleccionar el mes que queremos cerrar contablemente.	Selección
<b>Fecha de Cierre</b>	Muestra la fecha en que el cierre se esta llevando a cabo, el sistema toma la fecha actual para mostrarla.	Consulta
<b>Cierre Normativo</b>	Permite seleccionar el mes que queremos cerrar contablemente en el ámbito normativo.	Selección
<b>Fecha de Cierre</b>	Fecha en que se genera el cierre contable, el sistema toma la fecha actual.	Consulta

- Proceso de Cierre Contable Mensual

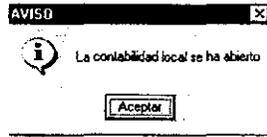
## Pasos a seguir:

45. Haga doble *click* en **Cierre Contable Mensual**. El sistema presentará automáticamente la ventana del mismo nombre.
46. Seleccione el **Catálogo Actual** con el que se va a trabajar en el campo del mismo nombre.
47. Seleccione el **Año** en que se va a trabajar el Cierre en el campo del mismo nombre.
48. Seleccione el mes al que va a aplicar el cierre haciendo *click* en el cuadro derecho de la columna **Contabilidad Cerrada**. El sistema presentará la siguiente ventana:



- 5 Oprima el botón **Aceptar** y el sistema pondrá la fecha actual del sistema.
- 6 En caso de haber cometido un error, o bien su desea que el cierre en ese mes no sea llevado a cabo, vuelva a oprimir el recuadro derecho del mes elegido, el sistema

presentará la siguiente ventana:



7Haga *click* en el botón **Aceptar** y el sistema volverá a abrir la contabilidad del mes seleccionado.

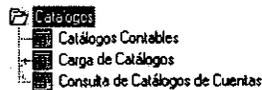
### 3. Catálogos Contables.

Los catálogos contables son aquellos que nos permiten controlar las cuentas que utilizamos para elaborar nuestra contabilidad. El Sistema de Contabilidad nos permite agregar, eliminar, modificar y consultar una cuenta. Cabe mencionar que tenemos tres tipos de catálogos de cuentas:

- 1.- Un Catálogo Normativo, que norma a todos los demás catálogos.
- 2.- Catálogos Concentradores, los cuales no contienen cuentas de detalle.
- 3.- Catálogos Generales, que si contienen cuentas de detalle.

#### 3.1 Catálogos.

El contenido de la opción Catálogos del Sistema de Contabilidad es el siguiente:



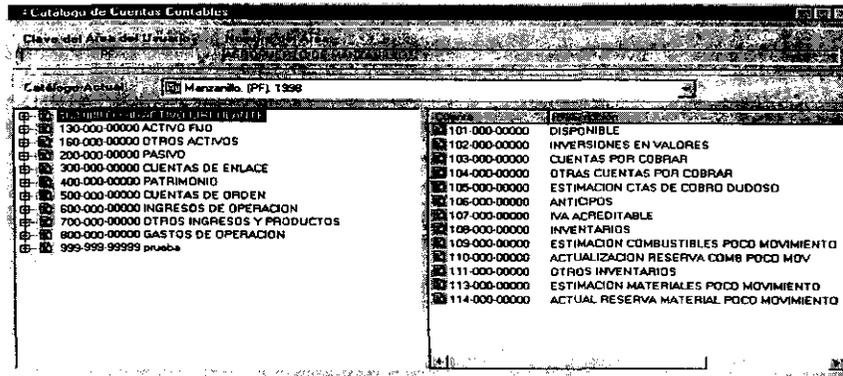
La opción **Catálogos Contables** nos permite controlar las cuentas necesarias para el registro de nuestra contabilidad.

En **Carga de Catálogos** tenemos la posibilidad de cargar las cuentas de detalle desde un archivo texto a algún catálogo en particular.

La opción **Consulta de Catálogos de Cuentas** contiene todos nuestros catálogos para que puedan ser consultados.

##### 3.1.1 Catálogos Contables.

La ventana **Catálogos de Cuentas Contables** nos permite agregar, eliminar, modificar y consultar alguna(s) cuenta (s). Vea la siguiente figura:



La ventana Catalogo de Cuentas Contables contiene los siguientes elementos:

Campo	Descripción	Tipo de Campo
<b>Clave del Area del Usuario</b>	Identificador único del área de trabajo del usuario.	Consulta
<b>Nombre del Area</b>	Nombre completo del área de trabajo del usuario.	Consulta
<b>Catálogo Actual</b>	Lista desplegable con el nombre del catálogo con el cual estamos trabajando.	Edición / Selección
<b>Columna sin nombre</b>	Muestra las cuentas que contiene el catálogo actual.	Consulta / Selección
<b>Cuenta</b>	Muestra el número de cuenta de detalle del catálogo general.	Consulta / Selección
<b>Descripción</b>	Nombre completo de la cuenta de detalle del catálogo general.	Consulta

Adicionalmente en la parte superior de la ventana, tenemos un conjunto de botones que llamamos Barra de Herramientas la cual contiene las siguientes opciones:

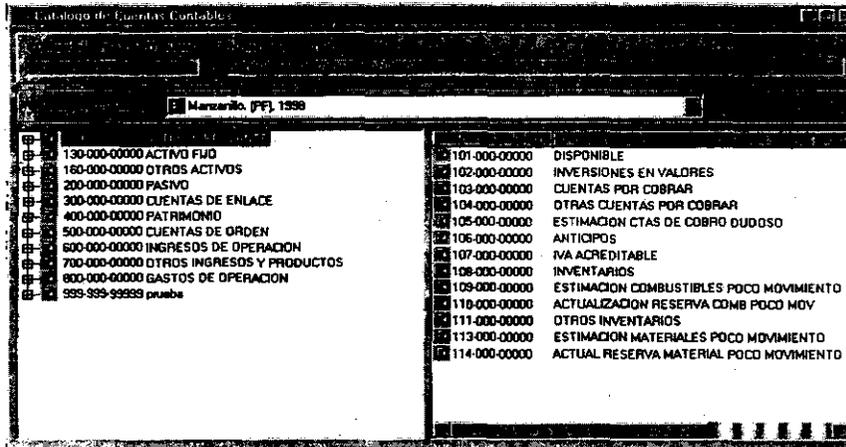
- |                  |   |     |
|------------------|---|-----|
| ↓ 1              | ↓ 2   | ↓ 3 |
| 1.- Nuevo Nivel. | Permite adicionar un nuevo nivel de cuenta. |     |
| 2.- Eliminar.    | Permite eliminar una cuenta.                |     |
| 3.- Modificar.   | Permite modificar una cuenta.               |     |

Lo anterior estará presente siempre que esta ventana sea activada.

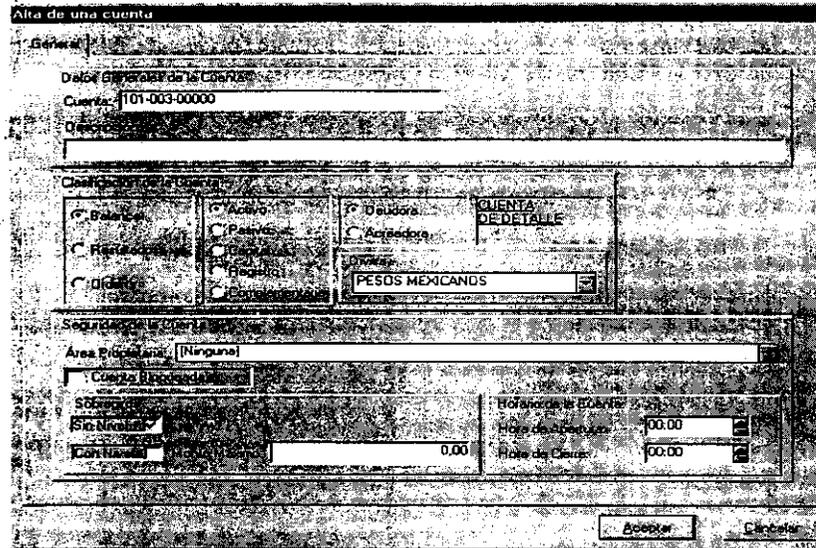
- Registro de una Cuenta

Pasos a seguir:

49. Haga doble *clic* en **Catálogos Contables**. El sistema presentará automáticamente la ventana **Catálogos de Cuentas Contables**. Vea la siguiente figura:



50. Haga *click* sobre la cuenta general en la que desee incluir una nueva cuenta y oprima el botón derecho del Mouse seleccionando la opción Nueva o bien seleccione el botón Nuevo Nivel de la barra de herramientas y haga *click* con el botón izquierdo del Mouse o utilice el método abreviado *Ctrl + N*. El sistema presentará la siguiente ventana:



La ventana Alta de una Cuenta contiene los siguientes elementos:

Datos Generales de la Cuenta		
<b>Cuenta</b>	Número de la cuenta que vamos a incluir.	Consulta
<b>Descripción</b>	Nombre completo de la cuenta que vamos a dar de alta.	Consulta
Clasificación de la Cuenta		
<b>Balance</b>	Nos permite seleccionar si la es propia para el Balance general de nuestra contabilidad.	Selección
<b>Resultados</b>	Nos permite seleccionar si la cuenta es propia para el Estado de Resultados de nuestra contabilidad.	Selección
<b>Orden</b>	Nos permite seleccionar si la cuenta es de Orden dentro de nuestra contabilidad.	Selección

<b>Activo</b>	Nos permite seleccionar si la cuenta pertenece al Activo dentro del Balance General, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de balance.	Selección.
<b>Pasivo</b>	Nos permite seleccionar si la cuenta pertenece al Pasivo dentro del Balance General, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de balance.	Selección.
<b>Capital</b>	Nos permite seleccionar si la cuenta pertenece al Capital dentro del Balance General, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de balance.	Selección.
<b>Registro</b>	Nos permite seleccionar si la cuenta pertenece a Registro, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de Orden.	Selección.
<b>Complementaria</b>	Nos permite seleccionar si la cuenta es complementaria, cabe mencionar que este botón se puede seleccionar únicamente si la cuenta se eligió en el recuadro anterior como de Orden.	Selección.
<b>Deudora</b>	En este campo seleccionamos si la cuenta a incluir en un nuevo nivel es de naturaleza deudora.	Selección.
<b>Acreedora</b>	En este campo seleccionamos si la cuenta a incluir en un nuevo nivel es de naturaleza acreedora.	Selección.
<b>Divisa</b>	Tenemos varios tipos de divisas para operar la cuenta, estos son: Pesos Mexicanos, Dolares, Udis, Libras Esterlinas, Pesetas.	Selección.
<b>Area Propietaria</b>	Muestra la descripción del Area Propietaria donde opera la cuenta.	Selección / Edición.
<b>Cuenta Bloqueada</b>	Este campo nos permite seleccionar si en algún momento queremos bloquear la nueva cuenta creada.	Selección.
<b>Sin Nivel</b>	En este campo seleccionamos si la cuenta nueva que creamos tiene un nivel para sobregiro en la misma, en caso de no tener se selecciona esta opción.	Selección.

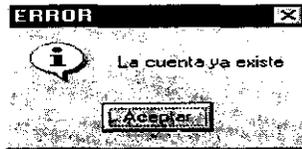
**Contabilidad**

<b>Con Nivel</b>	En este campo seleccionamos si la cuenta nueva que creamos tiene un nivel para sobregiro en la misma, en caso de tenerlo se selecciona esta opción.	Selección.
<b>Monto Máximo</b>	En este campo tecleamos el monto máximo de sobregiro de la cuenta recién creada en caso de que se haya seleccionado que cuenta con un nivel de sobregiro.	Edición
Horario de Operación		
<b>Horario de Apertura</b>	Como una medida de seguridad, el sistema provee de la posibilidad de tener un horario de operación para alguna cuenta en este campo habilitamos el horario de apertura de la misma.	Edición.
<b>Horario de Cierre</b>	Hora en que se cierra la operación de una cuenta.	Edición.

Además la ventana presenta los siguientes botones:

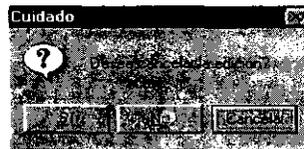
<b>Aceptar</b>	<b>Confirmar</b>
<b>Aceptar</b>	Confirma el alta de la nueva cuenta con todos los datos que han sido capturados.
<b>Cancelar</b>	Cancela la operación de dar de alta una nueva cuenta.

51. Introduzca el número de la nueva cuenta en el campo **Cuenta**. El número de cuenta es único por lo que debe de ser diferente a las ya existentes. En caso de introducir una cuenta ya existente el Sistema presentará el siguiente mensaje:



Oprima el botón **Aceptar** y teclee nuevamente el número de cuenta correctamente.

52. Introduzca el nombre completo de la cuenta en el campo **Descripción**.
53. Seleccione la clasificación de la nueva cuenta a incluir en el catálogo: **Balance, Resultados, Orden** si es de **Activo, Pasivo, Capital, Complementaria, Registro** si su naturaleza es: **Deudora o Acreedora**.
54. Seleccione el tipo de divisa con la que la cuenta va a operar en el campo **Divisa**.
55. Seleccione o introduzca el área propietaria de la cuenta en el campo **Área Propietaria**.
56. Haga *click* en el campo **Cuenta Bloqueada** en el caso de que así se requiera por seguridad o bien en caso contrario deje el cuadro en blanco.
57. Haga *click* en el campo **Sin Nivel** si la cuenta no tiene un nivel de sobregiro.
58. Haga *click* en el campo **Con Nivel** si la cuenta tiene un nivel de sobregiro.
59. Si su selección fue con nivel, introduzca el monto hasta donde se permite el sobregiro de la cuenta en el campo **Monto Máximo**.
60. En caso de requerir la operación de la cuenta en un horario establecido, introduzca la fecha de apertura de la cuenta en el campo **Horario de Apertura**, de igual forma introduzca el horario de cierre de la misma en el campo **Horario de Cierre**.
61. Guarde la información capturada.
62. En caso de que usted intente cancelar la operación de alta de la cuenta, el Sistema presentará el siguiente mensaje:



Si desea cancelar la edición de la cuenta haga clic en **Si**, de lo contrario presione el botón **No** o bien el botón **Cancelar**.

---

- **Modificación de una Cuenta**

**Pasos a seguir:**

5. Haga doble *clik* en **Catálogos Contables**. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**.

- Haga *clic* sobre la cuenta que desee modificar y oprima el botón derecho del mouse seleccionando la opción **Modificar** o bien seleccione el botón **Modificar** de la barra de herramientas y haga *clic* con el botón izquierdo del mouse o utilice el método abreviado *Ctrl + M*. El sistema presentará la siguiente ventana:

Modificar una cuenta

101-003-02001

[Ninguna]

PESOS MEXICANOS

0,00

00:00

00:00

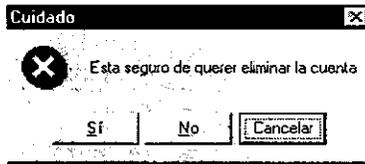
- Seleccione el campo que desee modificar.
- Introduzca el nuevo valor para ese campo.
- Guarde los cambios realizados.

#### • Eliminación de una Cuenta

##### Pasos a seguir:

- Haga doble *clic* en **Catálogos Contables**. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**.
- Haga *clic* sobre la cuenta que desee eliminar y oprima el botón derecho del mouse seleccionando la opción **Eliminar** o bien seleccione el botón **Eliminar de**

la barra de herramientas y haga *clik* con el botón izquierdo del mouse. El sistema presentará la siguiente ventana:



6. Haga *clik* en Si si está seguro de querer eliminar la cuenta seleccionada o en caso contrario seleccione la opción No o Cancelar para abortar el proceso.
7. Guarde los cambios realizados.

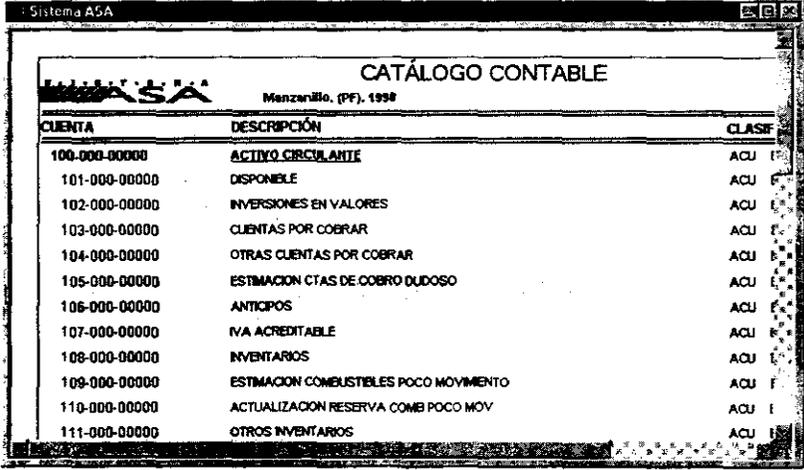
### •Elaboración del Reporte del Catálogo de Cuentas

Pasos a seguir:

1. Haga doble *clik* en Catálogos Contables. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**.
2. Haga *clik* con el botón derecho del mouse y seleccione la opción **Imprimir Catálogo**. El Sistema presentará la siguiente ventana:



3. Haga *clik* en el campo Nivel Contable a Reportar y seleccione el mismo de la lista desplegable que aparece del lado derecho, las opciones a seleccionar son: **Mayor, Cuenta, Subcuenta, SubSubcuenta**.
4. Oprima el botón **Aceptar** y el Sistema le presentará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:



CUENTA	DESCRIPCIÓN	CLASIF
100-000-00000	ACTIVO CIRCULANTE	ACU
101-000-00000	DISPONIBLE	ACU
102-000-00000	INVERSIONES EN VALORES	ACU
103-000-00000	CUENTAS POR COBRAR	ACU
104-000-00000	OTRAS CUENTAS POR COBRAR	ACU
105-000-00000	ESTIMACION CTAS DE COBRO DUDOSO	ACU
106-000-00000	ANTICIPOS	ACU
107-000-00000	IVA ACREDITABLE	ACU
108-000-00000	INVENTARIOS	ACU
109-000-00000	ESTIMACION COMBUSTIBLES POCO MOVIMIENTO	ACU
110-000-00000	ACTUALIZACION RESERVA COMB POCO MOV	ACU
111-000-00000	OTROS INVENTARIOS	ACU

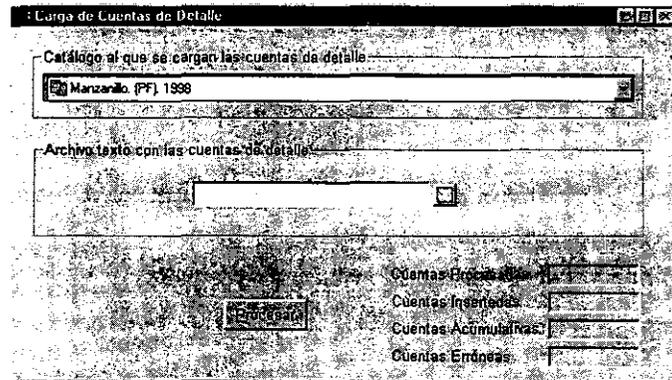
- Impresión del Reporte del Catálogo de Cuentas

Paso a seguir:

1. Seleccione en **A**rchivo el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

### 3.2 Carga de Catálogos

La opción **Carga de Catálogos** nos permite introducir al Sistema cuentas de detalle que se encuentran guardadas en un archivo de texto con un formato previamente establecido. Vea la siguiente figura:



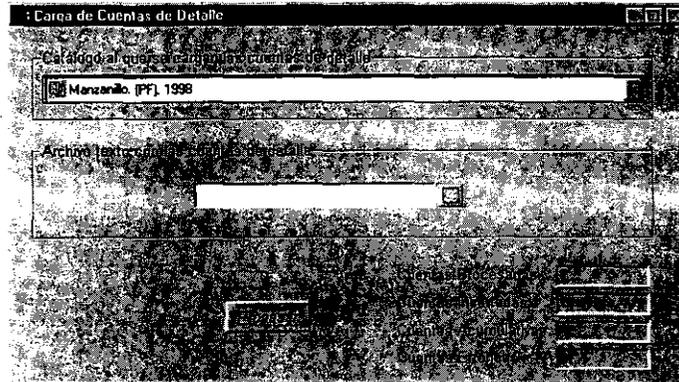
La ventana **Carga de Cuentas de Detalle** contiene los siguientes elementos:

Campo	Descripción	Tipo de Campo
<b>Catálogo al que se Cargan las Cuentas de Detalle</b>	Muestra el nombre del Catálogo al que se van a cargar las cuentas de detalle que van a ser importadas desde algún archivo de texto.	Selección
<b>Archivo Texto con las Cuentas de Detalle</b>	Muestra la ruta de donde vamos a importar el catálogo de cuentas de detalle.	Selección
<b>Procesar</b>	Realiza la operación del proceso del catálogo de cuentas.	Selección
<b>Cuentas Procesadas</b>	Muestra el total de cuentas procesadas.	Consulta
<b>Cuentas Insertadas</b>	Muestra el total de cuentas insertadas al catálogo actual.	Consulta
<b>Cuentas Acumulativas</b>	Muestra el total de cuentas acumulativas incluidas en el catálogo.	Consulta
<b>Cuentas Erróneas</b>	Muestra el número de cuentas erróneas encontradas en el proceso.	Consulta

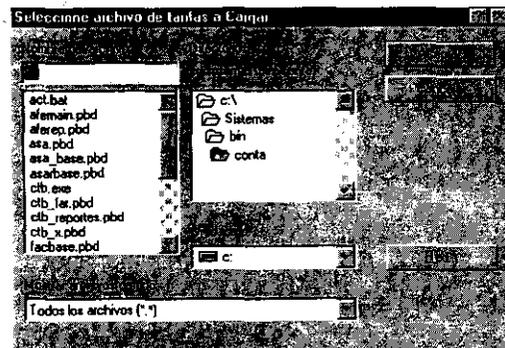
- Carga de un Catálogo

Pasos a seguir:

1. Haga doble *clik* en Carga de Catálogos. El sistema presentará automáticamente la ventana **Carga de Cuentas de Detalle**. Vea la siguiente figura:



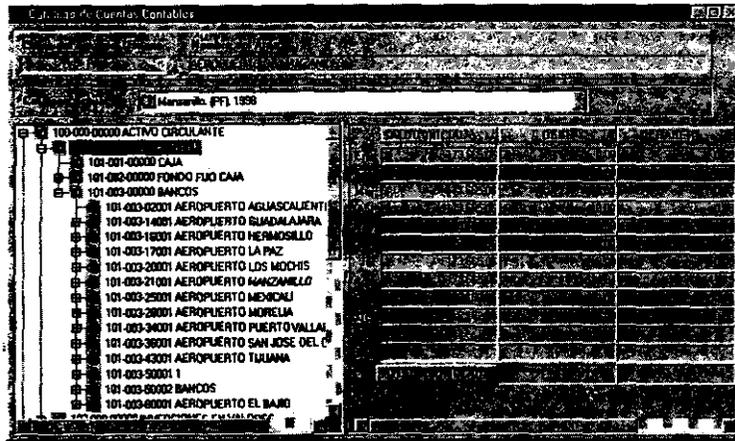
2. Haga *clik* en el campo **Catálogo** al que se cargan las cuentas de detalle y seleccione de la lista desplegable el nombre del catálogo a cargar.
3. Haga *clik* en el botón derecho del campo **Archivo texto** con las cuentas de detalle, el Sistema le presentará la siguiente ventana:



4. Escriba el nombre del archivo de texto de donde se van a importar las cuentas de detalle y seleccione la ruta de acceso al mismo, en caso de ser un archivo en un equipo que este dentro de la red haga *click* en el botón **Red** y seleccione la ruta deseada.
5. Haga *click* en el botón **Aceptar** en caso de haber introducido correctamente todos los datos o bien oprima el botón **Cancelar** para abortar el proceso.
6. Haga *click* en el botón **Procesar** y espere a que el proceso termine.

### 3.3 Consulta de Catálogos de Cuentas

Esta opción nos permite consultar los saldos de las cuentas de todos los catálogos que contiene la base de datos. Dentro de esta pantalla también tenemos la opción de generar varios de los reportes que requerimos para documentar nuestra contabilidad, como son: **Balanza de Comprobación, Libro Diario, Libros Auxiliares, Pólizas Capturadas, Balance General y Estados de Resultados.** Vea la siguiente figura:



La ventana **Catálogo de Cuentas Contables** contiene los siguientes elementos:

<b>Clave del Area del Usuario</b>	Identificador único del área de trabajo del usuario.	Consulta
<b>Nombre del Area</b>	Nombre completo del área de trabajo del usuario.	Consulta
<b>Catálogo Actual</b>	Lista desplegable con el nombre del catálogo con el cual estamos trabajando.	Edición / Selección
<b>Columna sin nombre</b>	Muestra las cuentas que contiene el catálogo actual.	Consulta / Selección
<b>Saldo Inicial</b>	Muestra el saldo inicial de la cuenta que seleccionamos.	Consulta
<b>Debe</b>	Muestra el total de movimientos de cargo de la cuenta que seleccionamos.	Consulta
<b>Haber</b>	Muestra el total de movimientos de abono de la cuenta que seleccionamos.	Consulta
<b>Saldo Final</b>	Muestra el saldo final de la cuenta que	Consulta

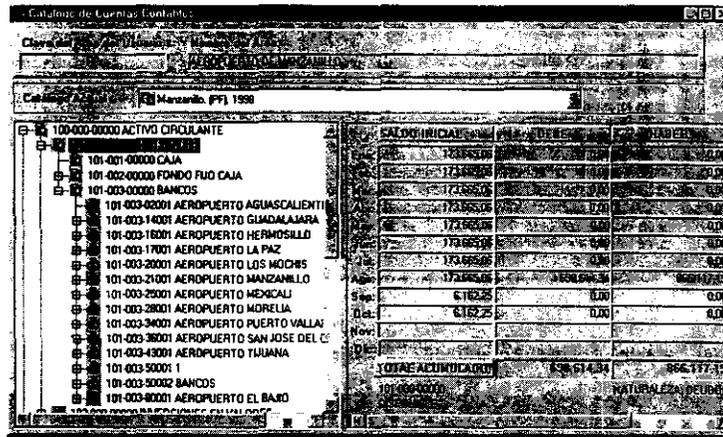
Contabilidad

	seleccionamos.	
<b>Total Acumulado</b>	Muestra el total de movimientos de cargos y abonos al final del recuadro de las cuentas.	Consulta

- Consulta del Saldo de una Cuenta

Pasos a seguir:

1. Haga doble *click* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**. Vea la siguiente figura:

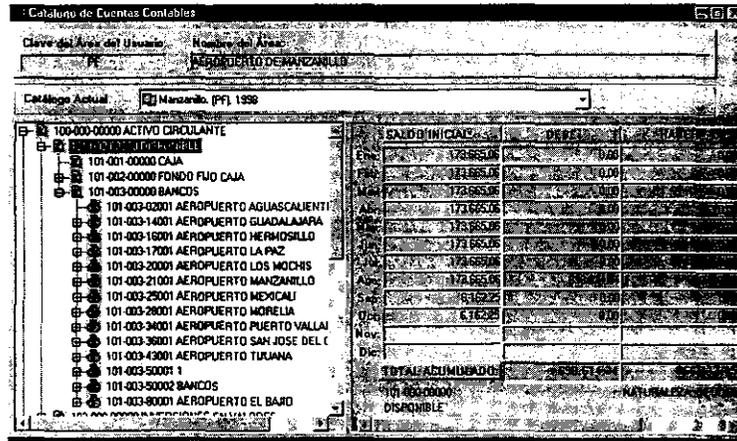


2. Haga *click* en el campo **Catálogo Actual** y seleccione de la lista desplegable que aparece del lado derecho el nombre del catálogo que contiene las cuentas que desee consultar.
3. Seleccione la cuenta que desee consultar y haga *click* sobre ella.
4. El sistema le mostrará el **Saldo Inicial**, los movimientos al **Debe** y **Haber**, el **Saldo Final** y el **Total Acumulado** anual de la cuenta que seleccionó.
5. Para observar todos los campos puede moverse con las barras de desplazamiento que aparecen en la parte de abajo de la ventana.

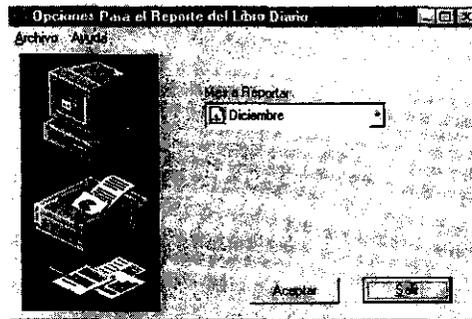
- Elaboración del Reporte del Libro Diario de un Catálogo

Pasos a seguir:

- Haga doble *click* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana Catálogo de Cuentas Contables. Vea la siguiente figura:



- Haga *click* con el botón derecho del mouse y seleccione la opción **Imprimir Libro Diario**. El Sistema presentará la siguiente ventana:



- Seleccione de la lista desplegable que aparece del lado derecho en el campo **Mes a Reportar** el mes a generar el Libro Diario.
- Oprima el botón **Aceptar** y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

Sistema ASA

**SISTEMA DE CONTABILIDAD**  
LIBRO DIARIO

Al: Último día del mes de AGOSTO de 1998

Aeropuerto: HMO General Ignacio Pesqueira

Tipo de Póliza: DIARIO

Tipo de Póliza: DIARIO No. de Póliza: 1

Área Generadora: PI AEROPUERTO DE HERMOSILLO

Concepto: NOMINA EMPL FUNC Y HON

Creación: 11/06/1998

Área	Cuenta	Descripción de la Cuenta	Cheque	Concepto del Movimiento	
00	806-001-00101	SUELDO	0000000	NOMINA EMPL FUNC Y HON	203,9
00	806-001-00116	ESTIMULO POR EFICIENCIA	0000000	NOMINA EMPL FUNC Y HON	26,4
00	806-001-00102	COMPENSACION ASA	0000000	NOMINA EMPL FUNC Y HON	2
00	806-001-00103	PRIMA QUINCENAL	0000000	NOMINA EMPL FUNC Y HON	6,6
00	806-001-00105	TIEMPO EXTRA ORDINARIO	0000000	NOMINA EMPL FUNC Y HON	22,3
00	806-001-00106	20% PRIMA DOMINICAL	0000000	NOMINA EMPL FUNC Y HON	2,6
00	806-001-00101	SUELDO	0000000	NOMINA EMPL FUNC Y HON	(4,06)

- Impresión del Libro Diario de un Catálogo

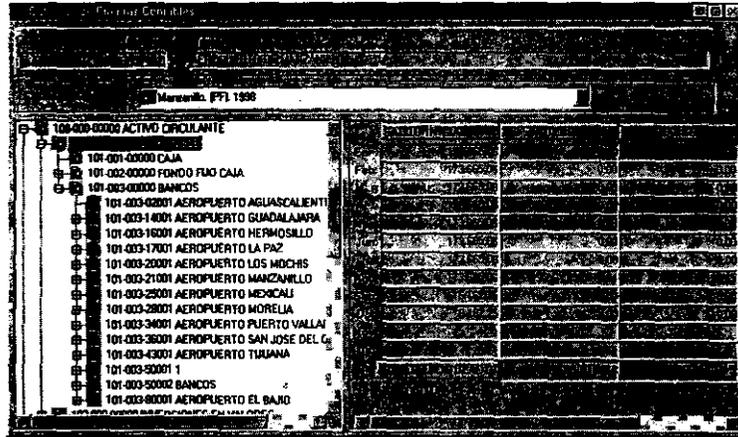
Pasos a seguir:

1. Seleccione en **Archivo** el comando **Imprimir...** o **Impresión Directa** u oprima el botón **Imprimir** o **Impresión Directa**.

- Elaboración del Reporte de Balanza de Comprobación

Pasos a seguir:

1. Haga doble *clic* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana Catálogo de Cuentas Contables. Vea la siguiente figura:



2. Haga *clic* con el botón derecho del Mouse en el área blanca del lado izquierdo de la ventana y seleccione, **Imprimir Balanza**. El Sistema presentará la siguiente ventana:



3. Haga *click* en el campo Nivel Contable a Reportar y seleccione el mismo de la lista desplegable que aparece del lado derecho, las opciones a seleccionar son: Mayor, Cuenta, Subcuenta, SubSubcuenta.
4. Seleccione de la lista desplegable que aparece del lado derecho en el siguiente campo el mes a generar la balanza.
5. Oprima el botón Aceptar y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

Área Cuenta	Descripción de la Cuenta	Saldo Anterior		Movimientos		
		Deudor	Acreedor	Debe	Haber	
21 100-000-00000	ACTIVO CIRCULANTE	23.883,21		0,00	0,00	23
21 101-000-00000	DISPONIBLE	8.182,26		0,00	0,00	8
21 104-000-00000	OTRAS CUENTAS POR COBRAR	(1.089,14)		0,00	0,00	(1)
21 106-000-00000	ANTICIPOS	(7.825,88)		0,00	0,00	(7)
21 107-000-00000	PAACREDITABLE	0,00		0,00	0,00	
21 111-000-00000	OTROS INVENTARIOS	28.215,98		0,00	0,00	28
21 130-000-00000	ACTIVO FIJO	42.350.566,87		0,00	0,00	42.350
21 131-000-00000	BIENES INMUEBLES HISTORICO	8.888.803,00		0,00	0,00	8.888
21 132-000-00000	BIENES INMUEBLES RESERVISION	141.483.220,00		0,00	0,00	141.483
21 133-000-00000	BIENES MUEBLES HISTORICO	3.480.288,00		0,00	0,00	3.480
21 134-000-00000	BIENES MUEBLES RESERVISION	14.708.581,10		0,00	0,00	14.708

- Impresión de la Balanza de un Catálogo

Pasos a seguir:

1. Seleccione en **A**rchivo el comando *Imprimir...* o *Impresión Directa* u oprima el botón Imprimir o Impresión Directa.

- Elaboración del Reporte de Libros Auxiliares

Pasos a seguir:

- Haga doble *click* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**. Vea la siguiente figura:

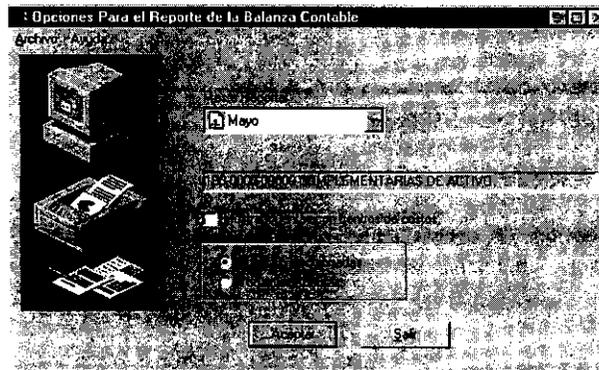
**Catálogo de Cuentas Contables**

Clave del Área del Usuario: Nombre del Área:  
 PF: AEROPUERTO DE MANZANILLO

Catálogo Actual: Manzana. (PF) 1999

	SALDO INICIAL	OPRES.	BASES
Ene	173.685,06	0,00	173.685,06
Feb	173.685,06	0,00	173.685,06
Mar	173.685,06	0,00	173.685,06
Abr	173.685,06	0,00	173.685,06
May	173.685,06	0,00	173.685,06
Jun	173.685,06	0,00	173.685,06
Jul	173.685,06	0,00	173.685,06
Ago	173.685,06	638.614,34	812.299,40
Sep	6.162,25	0,00	6.162,25
Oct	6.162,25	0,00	6.162,25
Nov			
Dic			
<b>TOTAL ACUMULADO:</b>	<b>638.614,34</b>		<b>638.614,34</b>
101-000-0000			NATURALEZA DEL C/008
DISPONIBLE			

- Haga *click* con el botón derecho del mouse y seleccione la opción **Imprimir Libros Auxiliares**. El Sistema presentará la siguiente ventana:



3. Haga *click* en el campo Nivel Contable a Reportar y seleccione el mismo de la lista desplegable que aparece del lado derecho, las opciones a seleccionar son: Mayor, Cuenta, Subcuenta, SubSubcuenta.
4. Seleccione de la lista desplegable que aparece del lado derecho en el siguiente campo el mes a generar el reporte.
5. Oprima el botón Aceptar y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

Cuenta	Descripción de la Cuenta	Saldo Anterior		Movimientos	
		Deudor	Acreedor	Debe	Haber
190-0000-00000	COMPLEMENTARIAS DE ACTIVO		27,016,383.18	0.00	0.00
191-0000-00000	DEPRECIACION ACUMULADA		466,684.70	0.00	0.00
191-0001-00000	MAQUINARIA Y EQUIPO DE OFICINA		87,790.45	0.00	0.00
191-0001-00002	ADMINISTRACION		67,790.45	0.00	0.00
191-0002-00000	MOBILIARIO Y EQUIPO DE OFICINA		29,727.25	0.00	0.00
191-0002-00002	ADMINISTRACION		29,727.25	0.00	0.00
191-0003-00000	EQUIPO DE TRANSPORTE		367,795.25	0.00	0.00

- Impresión del Libros Auxiliares

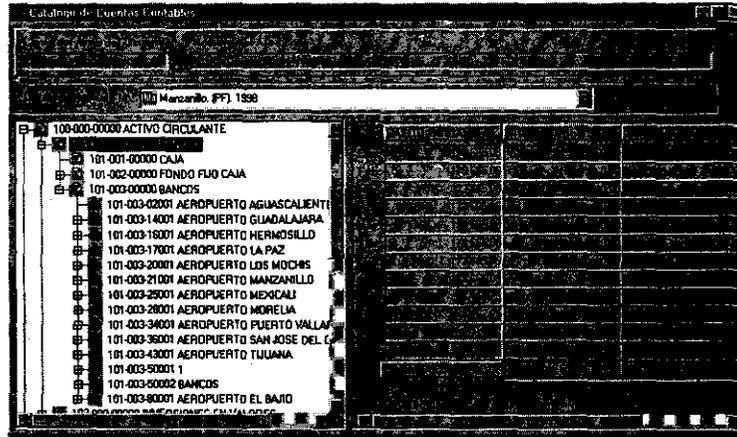
Pasos a seguir:

1. Seleccione en **Archivo** el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

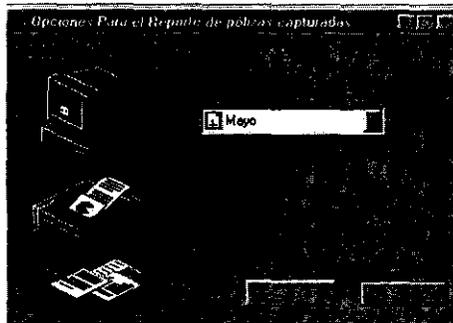
- Elaboración del Reporte de Pólizas Capturadas

Pasos a seguir:

1. Haga doble *click* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**. Vea la siguiente figura:



2. Haga *click* con el botón derecho del mouse y seleccione la opción **Imprimir Pólizas Capturadas**. El Sistema presentará la siguiente ventana:



3. Seleccione de la lista desplegable que aparece del lado derecho en el campo Mes a Reportar el mes a generar el reporte.
4. Oprima el botón Aceptar y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

**SISTEMA DE CONTABILIDAD**  
Pólizas Capturadas  
Al: Último día del mes de AGOSTO de 1998

Aeropuerto: HMO General Ignacio Pesqueira

Tipo de Póliza: DIARIO

Tipo de Póliza: DIARIO No. de Póliza: 1  
Área Generadora: P1 AEROPUERTO DE HERMOSILLO  
Concepto: NOMINA EMPL FUNC Y HON  
Creación: 11/08/1998

Área	Cuenta	Descripción de la Cuenta	Cheque	Concepto del Movimiento	
80	800-001-00101	SUELDO	000000	NOMINA EMPL FUNC Y HON	203.8
80	800-001-00118	ESTIMULO POR EFICIENCIA	000000	NOMINA EMPL FUNC Y HON	29.4
80	800-001-00102	COMPENSACION ASA	000000	NOMINA EMPL FUNC Y HON	2
80	800-001-00103	PRIMA QUINQUENAL	000000	NOMINA EMPL FUNC Y HON	5.5
80	800-001-00105	TIEMPO EXTRA ORDINARIO	000000	NOMINA EMPL FUNC Y HON	32.3
80	800-001-00106	30% PRIMA DOMINICAL	000000	NOMINA EMPL FUNC Y HON	2.8
80	800-001-00101	SUELDO	000000	NOMINA EMPL FUNC Y HON	(4.06)

- Impresión del Reporte de Pólizas Capturadas

Pasos a seguir:

1. Seleccione en Archivo el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

- Elaboración del Reporte de Balance General

Pasos a seguir:

1. Haga doble *click* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**. Vea la siguiente figura:

	SALDO INICIAL	DEBE	HABER
Ene:	173.665,06	0,00	0,00
Feb:	173.665,06	0,00	0,00
Mar:	173.665,06	0,00	0,00
Abr:	173.665,06	0,00	0,00
May:	173.665,06	0,00	0,00
Jun:	173.665,06	0,00	0,00
Jul:	173.665,06	0,00	0,00
Ago:	173.665,06	658.973,34	658.973,34
Sep:	6.162,25	0,00	0,00
Oct:			
Nov:			
Dic:			
<b>TOTAL ACUMULADO:</b>		<b>658.973,34</b>	<b>658.973,34</b>
101-000-00000			NETORALEZA: 000000
DISPONIBLE			

2. Haga *click* con el botón derecho del mouse y seleccione la opción **Imprimir Balance General**. El Sistema presentará la siguiente ventana:



3. Seleccione de la lista desplegable que aparece del lado derecho en el campo **Mes a Reportar** el mes a generar el reporte.

4. Oprima el botón **Aceptar** y el Sistema le mostrará la *Presentación Preliminar* del reporte a **Imprimir**. Vea la siguiente figura:

**SISTEMA DE CONTABILIDAD**  
BALANCE GENERAL  
AL Último día del mes de MAYO de 1999

Aeropuerto: **Cancun S.A. de C.V.**

<u>Activo</u>		
<i>Circulante</i>		
Fondos Fijos	10,412.39	0.00
Cuentas Bancarias	41,939,400.00	0.83
Inversiones en Valores	83,550,191.05	1.26
Cuentas	47,359,020.88	0.94
Reserva P/Ctas Incobrables	(525,436.93)	-0.01
Otras Cuentas por Cobrar	4,088,877.22	0.08
Fideicomisos con Saldo Deudor	10,432,935.57	0.21
Impuestos a Favor	8,352,415.36	0.17
Anticipos a Contratistas y Proveedores	9,487,804.55	0.19
	<b>194,675,429.29</b>	<b>3.67</b>
<i>Fijo</i>		
	0.00	0.00

- Impresión del Reporte Balance General

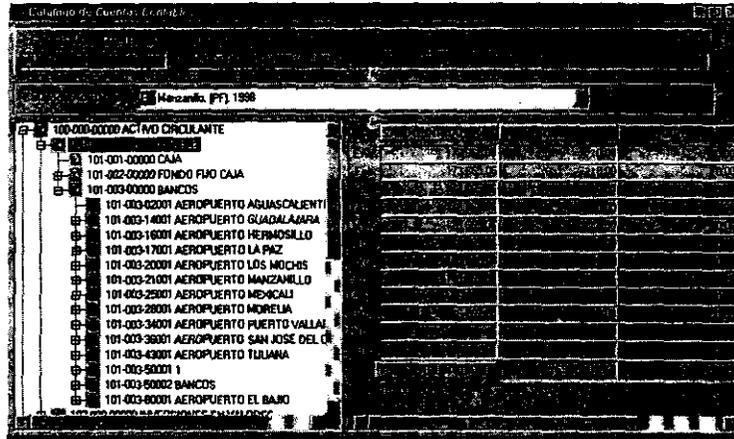
Pasos a seguir:

1. Seleccione en **Archivo** el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

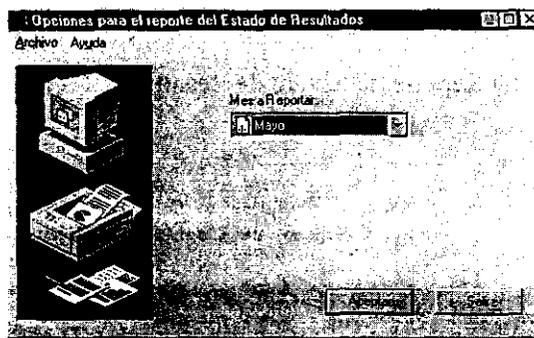
- Elaboración del Reporte de Estado de Resultados

Pasos a seguir:

1. Haga doble *click* en Consulta de Catálogos de Cuentas. El sistema presentará automáticamente la ventana **Catálogo de Cuentas Contables**. Vea la siguiente figura:



2. Haga *click* con el botón derecho del mouse y seleccione la opción **Imprimir Estado de Resultados**. El Sistema presentará la siguiente ventana:



3. Seleccione de la lista desplegable que aparece del lado derecho en el campo Mes a Reportar el mes a generar el reporte.
4. Oprima el botón Aceptar y el Sistema le mostrará la *Presentación Preliminar* del reporte a Imprimir. Vea la siguiente figura:

SISTEMA DE CONTABILIDAD		
ESTADO DE RESULTADOS		
AL Último día del mes de MAYO de 1999		
Aeropuerto: Cancun S.A. de C.V.		
<b>Ingresos de Operación</b>		
Servicios Aeroportuarios	39,133,879.84	30.67 %
Tarifa de uso de Aeropuerto Nacional	14,718,037.58	11.54 %
Tarifa de uso de Aeropuerto Internacional	66,180,506.58	51.87 %
Servicios Comerciales	7,554,315.23	5.92 %
	<b>127,586,739.23</b>	<b>100.00 %</b>
<b>Gastos de Operación</b>		
Sueldos	3,206,008.94	2.52 %
Cuotas IMSS	503,829.50	0.39 %
Cuotas INFONAVIT	93,396.10	0.07 %
	<b>3,803,234.54</b>	<b>2.98 %</b>

- Impresión del Reporte de Estado de Resultados

Pasos a seguir:

1. Seleccione en Archivo el comando *Imprimir...* o *Impresión Directa* u oprima el botón *Imprimir* o *Impresión Directa*.

## 4.1 Cambio de Clave de Acceso

Una medida de seguridad en todo sistema y que puede ser utilizada por diferentes usuarios, es la de cambiar continuamente la clave de acceso o contraseña al mismo, para esto el sistema cuenta con la ventana **Cambio de Clave de Acceso**. Vea la siguiente figura:

La ventana **Cambio de Clave de Acceso** contiene los siguientes elementos:

Campo	Descripción	Tipo de Campo
Usuario	Muestra la clave de identificación única y el nombre completo del usuario.	Consulta
Contraseña Anterior	En este campo se registra la contraseña de identificación actual.	Edición
Nueva Contraseña	Permite introducir la nueva contraseña de identificación.	Edición
Confirmación	Permite confirmar la nueva contraseña de identificación.	Edición

Además, la ventana anterior presenta el siguiente botón:

Botón	Descripción
Cambiar Contraseña	Activa la nueva contraseña del usuario.

- Cambio de Clave de Acceso

### Pasos a seguir:

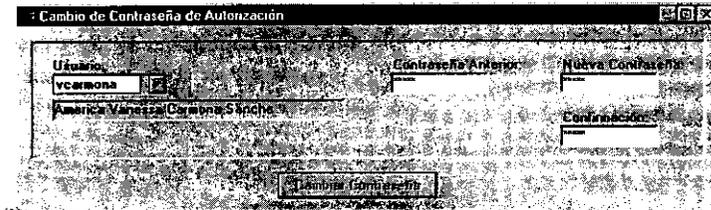
1. Haga doble *click* en **Cambio de Clave de Acceso**. Esta acción le presentará la ventana con el mismo nombre.
2. Escriba en el campo **Contraseña Anterior**, la contraseña de identificación actual y presione *Tab*.

3. Escriba en el campo Nueva Contraseña, la nueva contraseña de identificación y presione *Tab*. La contraseña puede contener hasta 21 caracteres (letras y números) como máximo y 6 caracteres como mínimo, el sistema presenta una mascara de asteriscos como medida de seguridad para que no pueda ser visualizada por otros usuarios.
4. Escriba en el campo Confirmación, nuevamente la contraseña de identificación y presione *Tab*... Al ejecutar este paso se activa el botón Cambiar Contraseña. Nuevamente este campo aparece con una mascara de asteriscos como protección
5. Oprima el botón Cambiar Contraseña. Si la nueva contraseña ha sido registrada con éxito en la base de datos del sistema, se presentará el siguiente mensaje:



## 4.2 Cambio de Contraseña de Autorización

Dentro del Sistema de Contabilidad existen ciertos procesos especiales para los que se requiere la autorización de una persona con la responsabilidad sobre las mismas para que estas se lleven a cabo. Para ello, requiere proporcionar al usuario la clave y su contraseña, esta es distinta a la de acceso. Vea la siguiente figura:



La ventana **Cambio de Contraseña de Autorización** contiene los siguientes elementos:

<b>Usuario</b>	Muestra la clave de identificación única y el nombre completo del usuario.	Consulta
<b>Contraseña Anterior</b>	En este campo se registra la contraseña de identificación actual.	Edición
<b>Nueva Contraseña</b>	Permite introducir la nueva contraseña..	Edición
<b>Confirmación</b>	Permite confirmar la nueva contraseña de identificación.	Edición

Además, la ventana anterior presenta el siguiente botón:

<b>Cambiar Contraseña</b>	Activa la nueva contraseña del usuario.
---------------------------	---

- Cambio de Clave de Contraseña de Autorización

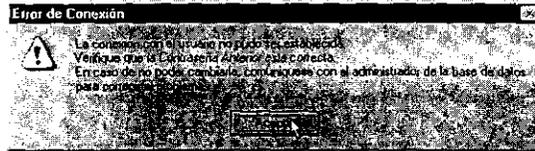
Pasos a seguir:

1. Haga doble *clik* en Cambio de Contraseña de Autorización. Esta acción le presentará la ventana con el mismo nombre.
2. Escriba en el campo Contraseña Anterior, la contraseña de identificación actual y presione *Tab.*, el sistema utiliza una mascara de asteriscos para que la contraseña no pueda ser visualizada.
3. Escriba en el campo Nueva Contraseña, la nueva contraseña de identificación y presione *Tab.* La contraseña puede contener hasta 21 caracteres (letras y números) como máximo y 6 caracteres como mínimo, se utiliza tambien una mascara de asterisco como protección.
4. Escriba en el campo Confirmación, nuevamente la contraseña de identificación y presione *Tab.*, encontramos nuevamente una mascara de asteriscos como medida de seguridad.
5. Oprima el botón Cambiar Contraseña. Si la nueva contraseña ha sido registrada con éxito en la base de datos del sistema, se presentará el siguiente mensaje:



### 4.3 Mensajes de Error en la Ventana Cambio de Clave de Acceso

- a. Se presenta cuando la contraseña introducida no coincide con la registrada en el sistema.



- b. Aparece cuando introduce la contraseña con menos de seis caracteres de longitud.



- c. Se presenta cuando la nueva contraseña es igual a la anterior.



- d. Aparece cuando la contraseña tiene menos de seis caracteres de longitud.



- e. Se presenta cuando la confirmación no es idéntica a la contraseña introducida.



- f. Se muestra cuando la contraseña tiene menos de seis caracteres de longitud.

