



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

31  
2EJ

**SISTEMA PARA EL CONTROL Y ACTUALIZACION DEL ACTIVO FIJO PARA UNA INSTITUCION BANCARIA.**

FALLA DE ORIGEN

T E S I S

QUE PARA OBTENER EL TITULO DE:  
INGENIERO EN COMPUTACION

P R E S E N T A N :

JOSE JAIME DAVILA BENITEZ

MIGUEL ANGEL LEON YVARRA

LUIS GIL PAEZ ORTIZ

EDILBERTO REYES LUNA

SAUL TORRES CHAVEZ

*Director de Tesis M. I. LAURO SANTIAGO CRUZ*



México, D. F.

Diciembre, 1995

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

**A la memoria de mi padre, por su rectitud, por su ejemplo, y por la amistad que siempre nos brindó.**

**A mi madre, por su inmenso amor.**

**A Carmen, por compartir su vida a mi lado, y por darme la mayor de las satisfacciones, mis hijos.**

**A mis hermanos, por la gran unión que siempre ha existido entre nosotros.**

**Al Ing. Carlos Gonzalez, por su valiosa colaboración y su apoyo en la elaboración de este trabajo de tesis, pero sobre todo por su amistad.**

**A mis compañeros de oficina, por la gran amistad que siempre me han brindado.**

**A mis compañeros en la elaboración del presente trabajo de tesis, por su dedicación y su empeño.**

**Al M.I. Lauro Santiago, por su valiosa dirección y apoyo en la elaboración de nuestra tesis.**

**SINCERAMENTE**

**Miguel Angel León Yvarra.**

**A MIS PADRES: EMMA Y JAIME.**

***POR DARME LA OPORTUNIDAD DE UNA EDUCACION DIGNA Y  
POR TODO SU APOYO EN LOS MOMENTOS MAS DIFICILES DE  
MI EXISTENCIA.***

**A MIS HERMANOS:**

***QUIENES CON SU EJEMPLO SIEMPRE FUERON UN ESTIMULO  
PARA SEGUIR ADELANTE EN MIS METAS.***

**A MI ESPOSA:**

***POR TODA SU PACIENCIA Y ANIMO CUANDO LOS RETOS  
ESTABAN CASI PERDIDOS.***

**A MI PEQUEÑA DANIELA:**

***POR SER LA LUZ QUE ILUMINO MI VIDA. A QUIEN ESPERO,  
ESTE ESFUERZO SIRVA DE EJEMPLO EN SU FUTURO.***

**A LA UNAM Y FACULTAD DE INGENIERIA:**

***POR TODOS LOS CONOCIMIENTOS Y EXPERIENCIAS  
ACADEMICAS OTORGADAS A UN SERVIDOR.***

**CON EL MAS PROFUNDO AGRADECIMIENTO  
JOSE JAIME DAVILA BENITEZ.**

**A MIS PADRES: PIEDAD Y MIGUEL.**

***PORQUE GRACIAS A SU DEDICACION Y ESFUERZO HE LOGRADO SEGUIR SIEMPRE ADELANTE, POR ESTAR CONMIGO EN LOS MOMENTOS FELICES PERO TAMBIEN EN LOS DIFICILES***

**A MIS TIOS:**

***POR CREER EN MI, POR DARME EL APOYO NECESARIO PARA SEGUIR ADELANTE, POR ESTAR A MI LADO Y A PESAR DE TODO SEGUIR UNIDOS COMO HASTA HOY***

**A MIS HERMANAS Y ABUELITA:**

***POR EL APOYO BRINDADO A LO LARGO DE MI CARRERA***

**A MIS GRANDES AMIGOS:**

***JORGE ARREOLA, HORACIO HERNANDEZ, EDILBERTO REYES, LAURA MENDEZ, RUBEN ELIZALDE, LUIS DAVID TREVILLA, ROBERTO BISTRIN, SANDRA MEZA, GABRIEL OSORIO, ISAAC ZEPEDA, AIDA GUTIERREZ, RAUL MARTÍNEZ, DALIA FLORES, CLAUDIA REYES, JOSE ANTONIO JUAREZ Y A TODOS AQUELLOS QUE NO HE MENCIONADO GRACIAS POR ESTAR A MI LADO A LO LARGO DE LA CARRERA COMPARTIENDO LOGROS, AVENTURAS Y FRACASOS.***

**SINCERAMENTE  
LUIS GIL PAEZ ORTIZ**

**A MIS PADRES: ANGELA Y ANGEL**

***POR EL ESFUERZO Y DEDICACION QUE SIEMPRE HAN TENIDO HACIA MI. POR MOSTRARME CON SU EJEMPLO EL CAMINO A SEGUIR. POR QUE CON SU APOYO, COMPRESION Y CONSEJOS HE LOGRADO ESTA META. MUCHAS GRACIAS.***

**A MIS HERMANOS:**

***ALFONSO, JOSE LUIS, DANIEL, CLAUDIA Y PABLO. POR EL APOYO INCONDICIONAL QUE SIEMPRE ME HAN BRINDADO. POR LA CONFIANZA QUE ME HAN TENIDO. POR QUE HAN SIDO UN APOYO EN TODO MOMENTO. MUCHAS GRACIAS.***

**A LA UNAM Y FACULTAD DE INGENIERIA**

***CON PROFUNDO AGRADECIMIENTO POR REPRESENTAR NUESTRA ALMA MATER COMO UNIVERSITARIOS.***

**SINCERAMENTE  
EDILBERTO REYES LUNA**

**A mis padres:**

**Este trabajo es parte de lo que están cosechando después de arduos días de sembrar, gracias por todo el apoyo que me dieron y por formar en mi un carácter con disciplina y perseverancia, que me ha servido para lograr todas las metas que me he propuesto.**

**A mi esposa:**

**Azucena tu eres el pilar de mis logros, gracias por ayudarme en todo lo que me propongo, gracias por sufrir conmigo y por sacrificarte cuando así tenia que ser, pero poco a poco estamos logrando todo lo que en su momento nos propusimos.**

**A mi hijo:**

**Paul Alejandro, tu eres el guía de todo lo que he hecho y de todo lo que me falta por hacer; eres junto con tu mama y sin temor a equivocarme lo mejor que me ha ocurrido en la vida.**

**A mis hermanos :**

**Gracias por toda la ayuda y comprensión que de ustedes he recibido, pero principalmente por que pase lo que pase siempre hemos estado juntos para apoyarnos.**

**A todos mis amigos:**

**Por todo lo que vivimos y todo lo que hicimos, pero sobre todo porque nunca perdimos el espíritu de lucha que desde un principio nos unió.**

**Mi mas sincero agradecimiento.**

**Saúl Torres Chávez.**

## AGRADECIMIENTO

**QUEREMOS DAR LAS GRACIAS A TODAS AQUELLAS PERSONAS QUE PARTICIPARON EN LA ELABORACION DE ESTA TESIS, Y QUE SIN SU COLABORACIÓN NO HUBIERA SIDO POSIBLE LLEVARLO A SU CULMINACIÓN.**

**AL M.I. LAURO SANTIAGO CRUZ POR SU DIRECCION Y ASESORAMIENTO EN LA ELABORACION DE ESTA TESIS.**

**A TODA LA GENTE DEL BANCO POR EL APOYO QUE NOS BRINDARON.**

**A NUESTROS AMIGOS DE LA FACULTAD DE INGENIERIA POR ESTAR A NUESTRO LADO A LO LARGO DE LA CARRERA.**

**FINALMENTE QUEREMOS AGRADECER A LA UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO Y FACULTAD DE INGENIERIA POR LOS CONOCIMIENTOS RECIBIDOS Y POR DARNOS LA OPORTUNIDAD DE DESARROLLARNOS COMO PROFESIONISTAS.**

**A TODOS ELLOS NUESTRO MAS SINCERO AGRADECIMIENTO**

**ATENTAMENTE**

**MIGUEL ANGEL LEON YVARRA  
LUIS GIL PAEZ ORTIZ  
EDILBERTO REYES LUNA  
SAUL TORRES CHAVEZ  
JAIME DAVILA BENITEZ**



# INDICE GENERAL

INTRODUCCIÓN.....	1
-------------------	---

## CAPITULO 1 "ANTECEDENTES GENERALES"

1. 1. Registro de los activos fijos en la institución	5
1. 2. Activos.....	6
1. 2. 1. Clasificación de los activos fijos .....	10
1. 2. 2. Características de los activos fijos .....	11
1. 2. 3. Depreciación de los activos fijos .....	11
1. 2. 4. Importancia de los activos fijos en una empresa .....	12
1. 3. Conceptos generales de análisis y diseño de sistemas .....	12
1. 3. 1. Análisis estructurado .....	14
1. 3. 2. Diseño del sistema .....	18
1. 3. 3. Desarrollo estructurado .....	25
1. 3. 4. Pruebas y liberación .....	29
1. 4. Redes Locales .....	35
1. 4. 1. Introducción .....	35
1. 4. 2. Ventajas de una Red Local .....	36
1. 4. 3. Componentes de una Red Local .....	37
1. 4. 4. Topología de una red .....	39
1. 4. 5. Sistema de cableado .....	42
1. 4. 6. Esquemas de acceso a la red .....	43
1. 4. 7. Tarjetas de red .....	44
1. 4. 8. Paquetes de software para red .....	46
1. 4. 9. Sistemas operativos para red .....	49

## CAPITULO 2 "ANÁLISIS ESTRUCTURADO"

2. 1. Definición del Problema .....	52
2. 1. 1. Procedimientos Existentes .....	55

2.1.2. Identificación del Problema .....	56
2.2. Información Recabada .....	56
2.3. Análisis de Costos .....	58
2.4. Recursos Disponibles para la Realización del Sistema .....	68
2.4.1. Hardware .....	68
2.4.2. Software .....	70
2.5. Justificación de Hardware y Software .....	73
2.5.1. Pruebas de eficiencia de manejadores de bases de datos ..	74

### **CAPÍTULO 3 "DISEÑO DEL SISTEMA"**

3.1. Diseño conceptual del sistema .....	80
3.2. Diseño modular del sistema (menú principal) .....	83
3.3. Diagrama de módulos del sistema .....	85
3.3.1. Módulo de altas .....	86
3.3.2. Módulo de bajas .....	87
3.3.3. Módulos de cambios .....	88
3.3.4. Módulo de mantenimiento .....	88
3.3.5. Módulo de reportes .....	89
3.3.6. Módulo de cifras de control .....	90
3.4. Diseño de entradas al sistema .....	91
3.5. Diseño de salidas del sistema .....	93
3.6. Definición de archivos .....	94
3.7. Definición de campos .....	97
3.8. Diagrama de entidad-relación .....	100
3.9. Referencias cruzadas (programas-archivos) .....	104
3.10. Referencias cruzadas (archivos-campos) .....	104
3.11. Diccionario de datos .....	105

## CAPÍTULO 4 "DESARROLLO DEL SISTEMA"

4.1. Programación del menú principal .....	107
4.2. Programación de módulos .....	112
4.2.1. Descripción de subrutinas .....	112
4.2.2. Altas .....	115
4.2.3. Bajas .....	120
4.2.4. Cambios .....	122
4.2.5. Mantenimiento .....	126
4.2.6. Reportes .....	130
4.2.7. Cifras de control .....	131
4.3. Reportes generados por el sistema .....	134

## CAPÍTULO 5 "PRUEBAS, MANTENIMIENTO E INSTALACIÓN"

5.1. Instalación del sistema .....	137
5.2. Directorios del sistema .....	137
5.3. Pruebas del sistema .....	140
5.4. Mantenimiento .....	142
5.4.1. Mantenimiento a la red .....	146
5.4.2. Mantenimiento al sistema .....	147
RESULTADOS Y CONCLUSIONES .....	149
BIBLIOGRAFÍA .....	153
APÉNDICES .....	154
GLOSARIO DE TÉRMINOS .....	171

## INTRODUCCIÓN

Los activos fijos son una parte fundamental en la conformación de cualquier empresa o institución, ya que esto representa el mobiliario, equipo de transporte, maquinaria, equipo de cómputo, equipo de comunicaciones, equipo de seguridad, etc.

Al ser una parte tan importante para el funcionamiento de cualquier institución o empresa, y en particular para nuestro caso de una institución bancaria, y debido a los montos de inversión tan cuantiosos que representan, es sumamente importante tener un pleno control sobre todos y cada uno de estos bienes.

Los Activos Fijos se han manejado desde la conformación de la institución en el año de 1925, ya que representan los bienes y equipos con los cuales se desempeñan muchas de las tareas diarias. Estos representan el mobiliario, enseres de oficina, equipos especiales, maquinaria, equipo de transporte, y más recientemente, equipo de cómputo y de telecomunicaciones.

En algunos casos, las políticas para la determinación de los bienes que pasan a conformar los activos fijos han variado con el tiempo. Esto depende en primera instancia de la vida útil de los bienes y de su costo. Si la vida útil es larga, generalmente pasan a formar parte de la inversión física de la institución, y por lo tanto, de sus activos fijos. Sin embargo, cuando se deben adquirir bienes en cantidades grandes (del orden de cientos o aún miles) estos también pueden ser considerados como activos fijos, aún cuando su costo individual sea bajo, de otra forma, pasarían a formar parte del gasto corriente del ejercicio en curso. La validez de estas políticas las determina el gobierno federal a través de la Secretaría de Hacienda y Crédito Público.

Inicialmente el registro de los activos fijos de la Institución se tenía exclusivamente en libros contables.

Sin embargo, con la introducción de los primeros equipos de cómputo se implementó un sistema de registro a base de tarjetas perforadas. Posteriormente, con el uso de terminales remotas, fue posible implantar el sistema actual de registro de los bienes de la Institución.

La vida útil del mobiliario y equipo varía de acuerdo a la naturaleza de éstos. Debido a que la cantidad de mobiliario y equipo que puede llegar a tener una institución es muy extensa, la tarea de dar de alta, baja y obtener la información necesaria para efectuar las depreciaciones para fines contables puede convertirse en una labor ardua y tediosa.

En el caso de la institución bancaria se cuenta con un número de artículos que ronda entre los 55,000 que representa una inversión de aproximadamente N\$100,000,000.00 a valor histórico.

Actualmente se tiene el registro de los bienes muebles y equipo de la Institución en un computador central, lo cual ya resulta obsoleto para un control adecuado. Esto debido principalmente a la falta de flexibilidad y de un adecuado registro de los datos principales que pueden identificar a un artículo en particular, ya que la consulta de la información sólo se puede dar a través de reporte.

### **Problemática y limitaciones**

En instituciones medianas y grandes es frecuente encontrar que para los bienes y equipos no se cuenta con un control adecuado respecto a su ubicación, estado físico, o adscripción de la dependencia administrativa a la que corresponde.

Esto es especialmente importante para la contabilidad de la institución, ya que en base a esta información se deben tomar decisiones sobre las cuentas que conforman los activos fijos. En particular es importante tener un conocimiento sobre los bienes que causan baja y alta dentro de la institución.

Esta información también es útil para la reexpresión, actualización y depreciación de activos fijos, ya que los montos pueden ser significativos.

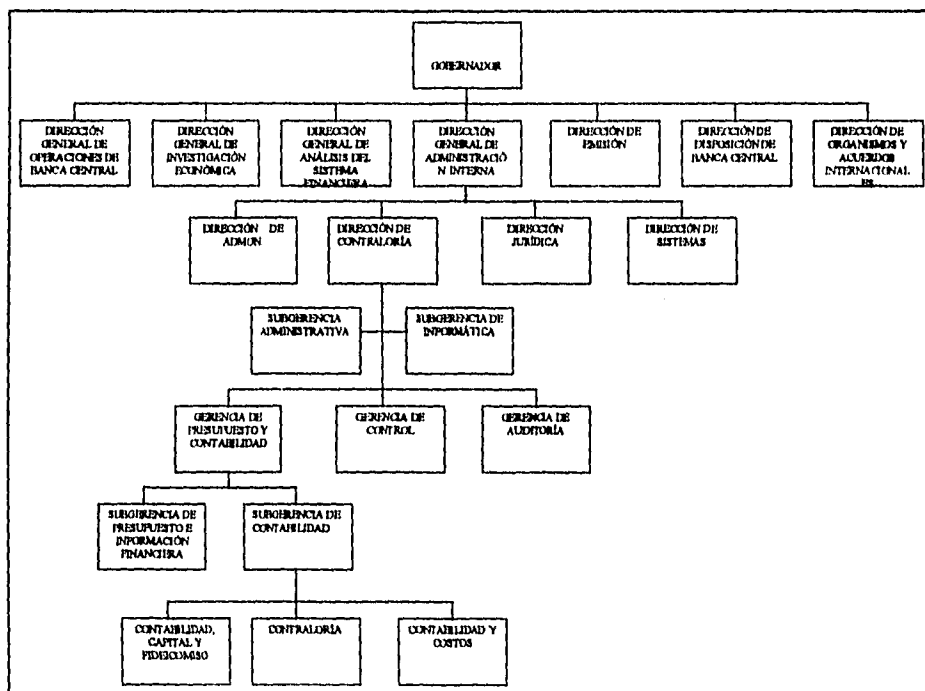
Bajo estas condiciones la recaudación de la información para efectuar las depreciaciones y las bajas se convierte en una tarea manual muy pesada que además consume mucho tiempo.

Es por ello que se ha propuesto desarrollar un Sistema de Control y Actualización del Activo Fijo (SICOFI) cuyo propósito fundamental sería el registro adecuado de los bienes muebles y equipo de la institución. Para ello, La Oficina de Contabilidad de Capital y Fideicomisos en Administración, requiere desarrollar

los elementos necesarios para este fin. (Ver *figura 1*, en la que se muestra la ubicación de esta oficina en la institución).

El sistema por desarrollar estará basado en el manejo de catálogos. El uso de catálogos permite un mejor control en el registro y actualización del activo fijo, ya que mantiene la consistencia entre los datos y sólo permite la entrada de artículos catalogados.

Al mismo tiempo el manejo de catálogos es flexible, ya que se permitirá su actualización en línea, aun desde la misma pantalla de captura, además el uso de catálogos nos da la posibilidad de explotar la información en una forma más fácil y consistente, esto debido a que se tienen los artículos agrupados por su naturaleza o uso.



**Figura 1. Organigrama general de la institución.**

En el presente trabajo se presenta el sistema desarrollado, dicho trabajo está organizado de la siguiente manera:

En el capítulo 1 se plantea la situación y problemática en el manejo de los activos fijos en la institución bancaria. Así como la teoría general de análisis estructurado, diseño estructurado, desarrollo del sistema, pruebas y liberación, mantenimiento de nuestro sistema, y teoría de redes de computadoras.

En el capítulo 2 se aplica la metodología de análisis estructurado para conocer a fondo el problema y plantear una solución. Así como, conocer el medio ambiente en el que se desarrollará este sistema.

En el capítulo 3 se analizan los principios fundamentales del diseño, así como, las técnicas de diseño que nos permitirán la realización de este sistema.

En el capítulo 4 se da un panorama general del desarrollo de la programación de nuestro sistema a partir del análisis y diseño definidos en los capítulos anteriores.

En el capítulo 5 se muestran algunas pruebas que nos van a permitir detectar errores y en base a esto hacer mejoras al sistema. Así como técnicas de mantenimiento, las cuales nos permiten hacer correcciones y mejoras al sistema de acuerdo a las necesidades del usuario, también se mostrará la forma de instalación del sistema.

Finalmente se presentan los resultados y conclusiones, bibliografía y apéndices.

# CAPITULO I

## ANTECEDENTES GENERALES

Este capítulo tiene como objetivo primordial el iniciar a los lectores en los conceptos teóricos más importantes que posteriormente se usarán en el desarrollo del actual trabajo. Pretendemos mostrar en principio toda la teoría, para que en capítulos posteriores se lleve a la práctica, al desarrollar un sistema de activos fijos para una institución bancaria.

En este capítulo se pueden distinguir tres grandes bloques de información: el primero describe la situación actual en cuanto al manejo de activos fijos en la institución, para continuar con los conceptos necesarios para entender que representan los activos en una empresa.

El segundo bloque está compuesto por toda la teoría que respalda el diseño estructurado, el cual será aplicado en el desarrollo del sistema, aquí, tratamos de tocar todos los puntos medulares de esta técnica, sin entrar a grandes detalles, si desea ampliar algún concepto en la bibliografía detallamos libros con estudios más profundos del tema.

Finalmente en el último bloque se hace un estudio de las redes locales, introduciendo al lector al tema, definiendo ventajas, detallando componentes, describiendo necesidades de software y hardware y mostrando las características de los programas y sistemas que se ejecutan en una red.

A continuación damos paso al desarrollo del capítulo con el estudio de los activos fijos y sus conceptos principales.

### **1.1 Forma actual del registro de los activos fijos en la institución**

En la actualidad el registro de los bienes muebles y equipo de la institución se efectúa en la Computadora Central.



Este registro lo lleva al cabo la Oficina de Contabilidad de Capital, en base a las facturas que le son entregadas por las dependencias que tienen la facultad de efectuar las compras.

Con base a las facturas se captura la información de fecha de adquisición, importe, y una descripción a texto libre de las características del artículo, que incluye también su número de serie y modelo. La dependencia de asignación es la que compra. Puesto que no existe una retroalimentación posterior acerca del destino de los artículos comprados, la dependencia a la que se le asignó la compra originalmente es la que permanece registrada. Esto origina que no se tenga un control adecuado del destino actual de los bienes y equipos de la institución.

Por otro lado, el único elemento que se tiene en la Oficina de Contabilidad de Capital para hacer consultas de los artículos registrados son reportes que se emiten periódicamente de los bienes y equipos a nivel detalle.

Estos reportes son muy voluminosos, ya que se tiene un registro cercano a los 55,000 artículos. Cuando tiene que efectuarse una baja, por ejemplo, primero debe buscarse el artículo en el reporte para identificarlo, lo que representa una tarea ardua y que requiere mucho tiempo; una vez identificado, debe darse de baja a través de un programa de aplicación de la Computadora Central.

Algunas características del sistema actual en el que se basa el registro de bienes y equipos son las siguientes:

1. Falta de mantenimiento en lo referente a la ubicación física de los bienes.
2. El Sistema actual carece prácticamente de documentación. Esto hace que el mantenimiento se convierta en una tarea en extremo difícil.
3. No existen consultas por terminal.
4. Los reportes que se emiten son de formato fijo.
5. En algunos casos se tiene el registro a nivel factura es decir, se tienen globalizados varios artículos con un sólo importe por el total de la factura.

## **1.2 Activos**

EL ACTIVO, para los economistas, es el patrimonio de la empresa y está integrado por dos grupos de bienes: el primero se refiere a los artículos de primera necesidad, como prendas de vestir, alimentos cotidianos, etc., que cubren necesidades inmediatas para poder subsistir.

El segundo grupo está formado por los instrumentos necesarios para producir los bienes que ya citamos, o sea los más apremiantes. Para el economista Adolfo Wagner, el Activo es el patrimonio destinado a la producción de bienes.

Schumpeter habla del Capital como la palanca que pone en condiciones al empresario, de someter a su soberanía los bienes que necesita, para imponer a la producción determinada tendencia, o sea esta concepción define al Capital como aquel patrimonio o suma de valores que sirve al empresario para obtener la disposición o el dominio sobre la fuerza del trabajo y los medios de producción.

Adam Smith clasifica las propiedades en forma muy parecida a la que empleamos en contabilidad, pues dicho economista distinguió dos grandes grupos que son: el que forma el Activo Circulantes y el que corresponde al Activo Fijo. La propiedades del Circulantes (en economía) se prestan una sola vez y desaparecen en el proceso de la producción, como las materias primas, la existencia de productos, etc. Este grupo se caracteriza porque la existencia de productos se recupera al venderse, retornando nuevamente, en otra forma, al empresario.

Las propiedades que van en el grupo del Activo Fijo son las que se utilizan más de una vez en la vida de la empresa, como la maquinaria, el edificio, etc.

Los economistas nos dicen que el Activo se puede considerar como el total de los bienes invertidos para obtener satisfactores.

Es tan grande su importancia, que el Activo de las empresas ha sido tomado como base para gravar impuestos. En México, como en otros muchos países, se creó el Impuesto al Activo de las Empresas (IADE o IMPAC) que grava el 2% sobre el valor del mismo en las empresas ganadoras, las perdedoras tendrán 3 años para poder compensar el 2% pagado contra el I.S.R. (Impuestos Sobre la Renta).

### **Desde el punto de vista jurídico**

Los bienes o propiedades tienen dos elementos, los llamados derechos reales y los derechos personales, los primeros están formados por la relación que se establece entre una persona y otra, con referencia a un bien, y que origina para el sujeto pasivo la obligación de respetar ese derecho.

Los derechos personales son las relaciones que se establecen entre personas, una acreedora, que exige a otra, la deudora, que dé un bien.

Los licenciados en Derecho nos dicen que, propiamente, no hay diferencia entre las clases de derecho que hemos mencionado.

Las propiedades que forman el Activo, tratadas desde el punto de vista del derecho real, presentan características reconocidas desde los remotos tiempos del Derecho Romano hasta nuestros días, y que son las siguientes:

### **El derecho de usar**

Consiste en que el propietario puede usar sus posesiones conforme a sus necesidades, siempre y cuando se respeten los derechos de los demás, por ejemplo, si una persona es dueña de un automóvil, puede usarlo cuantas veces lo desee o prestarlo a otra, o sencillamente dejarlo de usar, sin permitir que nadie disponga lo contrario.

### **El derecho de disfrutar**

El dueño de un bien lo puede rentar para disfrutar el precio del alquiler, por ejemplo, tomamos el mismo caso del automóvil: el propietario lo puede rentar y disponer de la renta, o bien, la renta recibida puede obsequiarse a un familiar, por simple gusto. Probablemente se piense que al obsequiar el importe de la renta no se disfruta de dicho valor, más para el propietario existe la satisfacción de regalar el beneficio de la renta, y éste es una forma de disfrutarla.

### **El derecho de disponer**

El propietario del bien lo puede disfrutar, enajenar o destruir, siempre y cuando no lesione el propietario los intereses de un tercero. Ejemplo: Nuevamente tomamos al dueño del automóvil, éste tiene todo el derecho de usar su bien como mejor le convenga, pero sujetándose a los reglamentos de tránsito.

### **El derecho de propiedad.**

No indica un poder absoluto sobre el bien que se posee, pues tiene ciertas limitaciones, en algunos casos, por ejemplo, las que la Constitución señala en su artículo 27, cuando se refiere a la propiedad de la tierra.

### **Desde el punto de vista contable.**

El término Activo, además de significar propiedades de una persona física, sociedad mercantil o corporación, incluye también derechos positivos sobre bienes tangibles e intangibles. Es requisito indispensable que exista el derecho legal específico para demostrar, sobre los demás, la propiedad de la cosa o bien de un servicio presente o futuro.

El "activo circulante" contiene a los recursos productores de ingresos de las empresas, y por lo tanto poseen mayor grado de disponibilidad, ya que en el corto plazo (período no mayor de 1 año) convierten su valor a dinero, como ejemplo tenemos:

- |                                  |                                     |
|----------------------------------|-------------------------------------|
| 1. Caja                          | 5. Documentos por Cobrar            |
| 2. Bancos                        | 6. Deudores Diversos                |
| 3. Acciones y Valores a la Vista | 7. Almacén de Mercancías            |
| 4. Clientes                      | 8. Pagos Anticipados a Corto Plazo. |

El "activo no circulante" son aquéllos que presentan menor grado de disponibilidad (la convertibilidad de su valor a dinero es mayor a un año). Dentro de esta clasificación se tiene los bienes que la empresa adquiere con la finalidad de usar y de no comerciar, así como pagos anticipados cuyos beneficios se recibirán a través de varios años. Como ejemplo de estos recursos se tiene a:

- |   |                              |
|---|------------------------------|
| 1. Documentos por Cobrar a Largo Plazo. | 7. Maquinaria.               |
| 2. Terrenos                             | 8. Depósitos en Garantía     |
| 3. Edificios.                           | 9. Acciones y Valores a L.P. |
| 4. Mobiliario y Equipo de Ofic.         | 10. Patentes y Marcas.       |
| 5. Equipo de Reparto                    | 11. Crédito Mercantil.       |
| 6. Equipo de Transporte.                | 12. Pagos Anticipados.       |

### Activo fijo

En contraposición a las circunstancias que caracterizan al activo circulante, pero atendiendo también a las posibilidades de recuperación y productividad de la inversión, existe el grupo denominado *activo fijo*, o de inversiones permanentes, el cual se forma por los terrenos, edificios, muebles, maquinarias y equipo en general.

Esta clase de inversiones está constituida por propiedades que no se destinan a la especulación sino al uso dentro del negocio. En consecuencia, no es posible pensar en que la inversión que representan deba recuperarse íntegramente en un momento determinado, como al tratarse de las mercancías, salvo el caso excepcional de una venta. En estos casos la recuperación debe procurarse en el transcurso del tiempo en que los bienes que representen la inversión proporcionen satisfactoriamente el servicio para el cual se han adquirido. Además, desde el punto de vista de su productividad, las inversiones representadas por activos fijos no generan beneficios directos, puesto que se destinan al uso y no a la venta. Su recuperación se opera de manera indirecta, pues los ingresos obtenidos por el activo circulante deben ser suficientemente amplios para absorber las inversiones permanentes.

Tal es el caso en las empresas de transporte, por ejemplo: compañías de aviación, navegación, ferrocarriles, autobuses, etc. En ellas las inversiones en equipos son sumamente cuantiosas y es de esperarse que su recuperación se realice en función de los ingresos que se obtienen de los usuarios del servicio.

En las fábricas sucede una cosa semejante: la inversión en maquinaria, equipo e instalaciones suele ser cuantiosa y su recuperación estará también en función de los ingresos que produzca el activo circulante, en este caso, por la venta de mercancías.

En ambos casos la recuperación será paulatina, durante todo el tiempo que el activo fijo esté en condiciones de buen uso y permita, en consecuencia, obtener ingresos por la venta de bienes o servicios.

### **1.2.1 Clasificación de los activos fijos**

Para un análisis más efectivo de los activos fijos éstos se clasifican en varios grupos; cabe señalar que cada empresa tiene su forma propia de clasificación, dependiendo del giro o funciones de ésta; sin embargo, se pueden decir que los siguientes grupos aparecen en casi todas las empresas:

**Maquinaria y Equipo.** Son todas las máquinas, equipos de transporte, motores, herramientas automáticas, máquinas de oficina y otros equipos especializados con que cuenta una empresa y que en determinado momento puede vender o rentar para obtener un beneficio económico.

**Terrenos.** Es toda porción de tierra que se utiliza en las operaciones normales de un negocio, el cual debe mostrarse separadamente de los edificios, por que la depreciación afecta, en general, a los edificios solamente. Al costo del terreno debe incluirse los gastos notariales, comisiones a agentes, abogados, costos de urbanización y gastos de localización entre otros.

**Edificios.** Un negocio puede poseer edificios que son utilizados en la operación normal del negocio y que han sido construidos en terreno propio o arrendado. El costo de un edificio incluye la construcción, instalación de equipo con carácter permanente, permisos de construcción, honorarios de arquitectos e ingenieros, gastos de supervisión y administración, etc. Los costos terminan cuando el inmueble está en condiciones de servicio.

**Herramientas.** Estas pueden dividirse en herramientas de máquinas y herramientas de mano. Las primeras por lo general son pesadas, de duración prolongada y fáciles de controlar. Las segundas por lo contrario, son generalmente pequeñas, de corta vida y fáciles de perder, por tanto, son difíciles de controlar.

**Otros Grupos.** Existen otros grupos o clasificaciones de activos que en nuestro punto de vista son especializados y depende en gran parte del servicio que preste la empresa, entre estos podemos mencionar los siguientes: Recursos Naturales, Equipo de Cómputo, Derechos de Propiedad Literaria, Arrendamientos, Franquicias, Patentes, Marcas, Créditos, Aire Acondicionado, etc.

### **1.2.2 Características de los activos fijos**

Una vez que los activos fijos son clasificados por su naturaleza o utilización, existen ciertos datos particulares de cada activo que deben ser tomados en cuenta para determinar posteriormente su valor a futuro. Los datos que deben ser recabados por cada activo fijo son los siguientes:

1. La fecha de adquisición del activo fijo.
2. El costo original del activo fijo.
3. La vida de servicio estimada.
4. El valor de desecho estimado.
5. El grado en que se usa la propiedad.
6. Las normas para reparación y mantenimiento.
7. La eficiencia actual del activo en comparación con activos nuevos similares.

Todos estos datos nos ayudarán a determinar en forma clara y sistematizada el monto actual de la inversión en una empresa; además, el contar con un conocimiento de los valores estimados actuales es muy útil para fines contables y administrativos.

### **1.2.3 Depreciación de los activos fijos**

La depreciación es un procedimiento que tiene como fin distribuir de manera sistemática y razonable el costo de los activos, menos su valor de desecho (si lo tiene), entre la vida útil estimada de la unidad. Por lo tanto, la depreciación contable es un proceso de distribución y no de valuación.

Esta distribución puede hacerse conforme a dos criterios generales, uno basado en tiempo y otro en unidades producidas.

Dentro de cada uno de los criterios generales existen varios métodos alternativos y debe adoptarse el que se considere más adecuado, según las políticas de la empresa, y características del bien. La depreciación debe calcularse sobre bases y métodos consistentes a partir de la fecha en que empiecen a utilizarse los activos fijos.

El período para depreciar un activo fijo está en función del grupo en el cual fue incluido dicho activo, esto es, el tiempo en el cual se deprecia cada grupo, por ejemplo, los edificios se deprecian en 20 años, mientras que la maquinaria es depreciada en 10 años.

Mientras un activo fijo se esté depreciando, dicho activo va adquiriendo un valor actualizado, esta actualización se determina en base a la inflación mensual determinada por el Banco de México. Este organismo emite mensualmente índices para cada grupo de activo, donde indica un factor, el cual es aplicado a la inversión histórica del activo para darnos como resultado el valor actual del mismo.

#### **1.2.4 Importancia de los activos fijos en una empresa**

**Valor histórico original.** De acuerdo con los principios de contabilidad, las inversiones en activos tangibles (como inmuebles, maquinaria y diversas clases de equipo) deben valuarse al costo de adquisición, y tener presente que dicho costo cesa en el momento en que dichos bienes están en condiciones de usarse.

En caso de que la propia empresa los produzca, será la suma de todos los elementos necesarios para su fabricación (materia prima, mano de obra directa y gastos indirectos) hasta que estén totalmente terminados. Es importante considerar que el costo mencionado es aquel que se eroga cuando la economía resulta más o menos estable y, como consecuencia, no existen los efectos del fenómeno inflacionario, sino que el mercado se encuentra en situación normal por lo que se considera que los costos de materia prima, mano de obra y gastos indirectos se mantienen constantes.

### **1.3 Conceptos generales de análisis y diseño de sistemas**

#### **Marco conceptual**

A pocos años del inicio del siglo XXI, los cambios en la economía mundial continúan presentándose para modificar el contexto internacional hacia la unificación comercial de diversos países.

Hoy en día en nuestro país presenciamos momentos cruciales de cambio, en los cuales nuestras decisiones tendrán vital importancia para el desarrollo y crecimiento de nuestra economía.

Particularmente importante será la nueva dimensión competitiva, emanada de la infraestructura tecnológica que proyectará a las diversas organizaciones hacia una competencia internacional.

Este panorama hace que las empresas establezcan como prioritaria la generación de planes estratégicos para el desarrollo de sistemas de información que les permitan cosechar las nuevas oportunidades que el entorno nos brinda, buscando obtener una ventaja competitiva sin olvidar los riesgos que trae consigo.

El inicio de la década de los noventa se ha caracterizado por el surgimiento de bloques comerciales, que fuerzan a las organizaciones a alcanzar un alto nivel competitivo, aplicando alta calidad en los bienes y servicios ofrecidos garantizando así, su subsistencia dentro del bloque comercial.

El nivel competitivo de cualquier empresa estará sustentado en la infraestructura tecnológica y en el grado de automatización que puede alcanzar. Para lograr lo anterior se requiere de un fuerte apoyo del área de sistemas. Por ello la presente técnica descrita a continuación pretende orientar en la metodología, control y administración, configuración del equipo de trabajo y empleo de herramientas automatizadas de alta productividad para el desarrollo de sistemas de información.

Conocidos son los problemas relacionados con el desarrollo de aplicaciones, destacando entre otros:

- Alto costo en el desarrollo de sistemas.
- Incumplimiento de los planes de trabajo y fechas compromiso.
- Aplicaciones desarrolladas que no satisfacen los requerimientos del usuario.
- Complejidad y alto costo en el mantenimiento de sistemas de operación.
- Sistemas no implantados debido a la resistencia al cambio presentada por usuarios no comprometidos con el proyecto.
- Proveedores improvisados de servicios informáticos ofreciendo servicios de calidad paupérrima.

Los beneficios obtenidos por el área de sistemas al aplicar la metodología adecuada serán los siguientes:

- Reducción de costos derivados de un mejor aprovechamiento de los recursos humanos y tecnológicos.
- Mayor satisfacción del cliente al lograr una conceptualización adecuada de los requerimientos del usuario, minimizando los cambios efectuados a la aplicación. Lo anterior redundará en períodos de desarrollo más cortos.



- Incrementar la calidad de los sistemas debido a la fuerte participación del usuario y al empleo de herramientas automatizadas de alta productividad.

### **1.3 Análisis estructurado**

El análisis estructurado no sólo tiene como objeto especificar requisitos, sino que se diseñó como una técnica para dividir, estructurar y expresar ideas, con independencia del lenguaje en el que se establezcan tales ideas. El análisis estructurado se basa en gran medida en gráficas para indicar la estructura y las relaciones, para lo que utiliza alrededor de 40 símbolos gráficos distintos, una descripción expresada de esta forma guarda la misma relación con el sistema aplicado que la que guarda un anteproyecto con el sistema de ingeniería que describe.

### **Análisis de sistemas**

El análisis de sistemas es una actividad que engloba la mayoría de las tareas que hemos llamado colectivamente ingeniería de sistemas, basados en computadora. El análisis de sistemas se realiza con los siguientes objetivos presentes:

1. Identificar las necesidades del cliente.
2. Evaluar la viabilidad del sistema.
3. Realizar el análisis técnico y económico.
4. Asignar funciones al Software, al Hardware, a la base de datos y a otros elementos del sistema.
5. Establecer restricciones de costos y tiempo.
6. Crear una definición del sistema que sea la base para todo el trabajo de ingeniería subsecuente.

Excelentes conocimientos en hardware y software, así como la adecuada capacitación del personal en el manejo de los sistemas, son requeridos para alcanzar con éxito los objetivos expuestos anteriormente.

### **Identificación de las necesidades**

El primer paso del proceso de análisis de sistemas implica la identificación de las necesidades. El Analista se entrevista con el cliente o su representante

La identificación de las necesidades es el punto de partida en la evolución de un sistema basado en computadora.

El analista se asegura de distinguir entre lo que necesita el cliente (elementos críticos para la realización) y lo que el cliente quiere (elementos deseables pero no esenciales)

### **Estudio de viabilidad**

Todos los proyectos son realizables si cuentan con recursos ilimitados y tiempo infinito, desafortunadamente el desarrollo de un sistema basado en computadora se caracteriza por la escasez de recursos y la dificultad de los plazos de entrega. Es necesario y prudente evaluar la viabilidad de un proyecto lo antes posible; si un sistema mal concebido se reconoce al principio de la etapa de definición, se evitarán meses o años de esfuerzo e inversiones de miles o millones de pesos. El estudio de viabilidad se centra en cuatro áreas de interés primario:

**Viabilidad económica.** Es una evaluación del costo de desarrollo frente al beneficio final producido por el sistema desarrollado.

**Viabilidad técnica.** Es un estudio de funcionalidad, rendimiento y restricciones que puede afectar la posibilidad de realizar un sistema aceptable. Las consideraciones que se asocian normalmente a la viabilidad técnica comprenden:

- *Riesgo de Desarrollo.* ¿Puede el término del sistema ser diseñado de tal forma que las funciones y rendimiento necesarios se consigan dentro de las restricciones determinadas en el análisis?
- *Disponibilidad de recursos.* ¿ Hay personal calificado para desarrollar el elemento sistema en cuestión? o bien ¿están disponibles para el sistema otros recursos necesarios como hardware y software?
- *Tecnología.* ¿Ha progresado la tecnología lo suficiente para soportar el sistema?

**Viabilidad legal.** Es una determinación de cualquier infracción, violación o ilegalidad que pudiera resultar del desarrollo del sistema. La viabilidad legal comprende un amplio rango de aspectos que incluyen contratos, responsabilidad, infracciones y un millar de otros detalles frecuentemente desconocidos para el personal técnico.

### **Etapas del estudio de viabilidad**

Las etapas del estudio de viabilidad comprenden siete etapas de importancia para su realización, a continuación definimos éstas:

1. **Introducción.** Un breve comentario del Problema, del entorno en que se va a desarrollar el sistema y las restricciones que afectan el proyecto.
2. **Resumen de gestión y recomendaciones.** Un resumen de los hallazgos y recomendaciones importantes para el desarrollo posterior del sistema .
3. **Alternativas.** Una presentación de configuraciones alternativas del sistema; son criterios que fueron usados al seleccionar la configuración final.
4. **Descripción del sistema.** Una versión abreviada de la información contenida en la especificación del sistema o una referencia a esta especificación.
5. **Análisis de costos y beneficios.** Una justificación económica del sistema.
6. **Evaluación del riesgo técnico.** Una presentación de la viabilidad técnica.
7. **Ramificaciones legales.** El estudio de viabilidad es revisado primero por el jefe de proyectos para asegurar la fiabilidad del contenido y por el jefe administrativo superior para determinar el estado del proyecto.

### **Análisis económico**

Entre la información más importante contenida en el estudio de viabilidad está el análisis de costos y beneficios, un intento de justificación económica para un proyecto de un sistema basado en computadora. El análisis de costos y beneficios marca los costos del desarrollo del proyecto y los contrasta con los beneficios tangibles e intangibles de un sistema.

### **Análisis técnico**

Durante el análisis técnico el analista evalúa los méritos técnicos del concepto de sistema, mientras que al mismo tiempo recoge información adicional sobre el rendimiento, la fiabilidad, la facilidad de mantenimiento y la posibilidad de producción.

El análisis técnico empieza con una definición de la viabilidad técnica del sistema propuesto ¿qué tecnologías se requieren para conseguir la funcionalidad y el rendimiento del sistema? ¿qué nuevos materiales, métodos, algoritmos o procesos se requieren y cuál es el riesgo de su desarrollo? ¿Cómo afectarán al costo estos elementos de tecnología?

Los resultados obtenidos del análisis técnico forman la base para otra decisión en el sistema de tipo "seguir/no seguir". Si el riesgo técnico es alto, si los modelos indican que la funcionalidad deseada o el rendimiento no pueden ser alcanzados, si las piezas no encajan bien -hay que volver a la mesa de trabajo-.

### **Lista de comprobaciones para el análisis de sistemas**

El proceso que llamamos análisis de sistemas es una actividad de resolución de problemas que quieren una comunicación intensiva entre el cliente y el que desarrolla el sistema, esta lista se compone de :

- Planificación del análisis.
- Contactos con los usuarios.
- Objetivos del sistema.
- Sistema actual.
- Elementos y estructura de datos.
- Entrevista a los usuarios.
- Investigación en otros sistemas.
- Propositiones alternativas.
- Seleccionando un diseño alternativo.
- Análisis estructural.
- Planes para la próxima fase.
- Presentaciones y revisiones de la gestión.

### **Análisis de requerimientos del usuario**

Uno de los aspectos más importantes en el Análisis es la identificación de las necesidades del usuario. Cuando el analista se entrevista con el usuario debe definir los objetivos del sistema, pero debe tener la capacidad de identificar y diferenciar lo que el usuario necesita y lo que el usuario quiere.

El analista, además, debe realizar una evaluación de la información suplementaria que se le presentará en un documento de conceptos del sistema. Este puede ser preparado por el usuario, pero es de suponerse que una vez que lo haya estudiado el analista, surgirán modificaciones al mismo.

### **Especificación del sistema**

La especificación del sistema es la primera documentación en el proceso de ingeniería de sistemas de computadoras. Cada sección principal de la especificación se describe a continuación:

- **Introducción.** Describe los objetivos del sistema y del entorno en el que se operará el sistema. Esta sección también contiene un resumen ejecutivo que especifica el alcance del proceso de desarrollo del sistema.

- **Descripción funcional.** En esta sección se suministra una descripción de cada función del sistema. La descripción incluye una narración funcional que describe la información de entrada, las tareas a realizar, la información resultante y los datos de interfaz adicionales.
- **Restricciones.** Las categorías típicas incluyen el entorno externo, las interfaces, el diseño y la implementación, los recursos y el costo o la planificación del tiempo.
- **Costo.** En esta etapa el proceso de ingeniería de sistemas de computadora puede ser imposible determinar estimaciones de costo precisas. La planificación del software y su equivalente en hardware debe ser realizada para permitir estimaciones de costos detalladas.
- **Planificación temporal.** Un plan de desarrollo de sistema se puede determinar sobre una fecha de finalización suministrada por el cliente, por razones de mercado y por fuerzas externas.

### **Revisión de la definición del sistema**

La revisión es realizada por el que desarrolla y por el cliente para asegurar que :

1. El alcance del proyecto se ha perfilado adecuadamente
2. La funcionalidad, el rendimiento y las interfaces se han definido correctamente
3. El análisis del entorno y el riesgo en desarrollo justifican el sistema
4. El que desarrolla y el cliente tienen la misma percepción de los objetivos del sistema.

### **1.3.2 Diseño del sistema**

El diseño del sistema es un proceso creativo que requiere del diseñador ciertas cualidades y el diseño final suele ser una repetición de varios diseños preliminares. Un sistema de software bien diseñado es fácil de aplicar y mantener, además de ser comprensible y confiable. Los sistemas mal diseñados, aunque puedan funcionar, suelen ser caros de mantener, difíciles de probar y poco confiables. La etapa del diseño es, por tanto, la parte más importante del proceso de desarrollo del software.

Hasta hace poco tiempo, el diseño del software era un proceso destinado a un fin determinado. Dado un conjunto de requisitos, por lo general en lenguaje natural, se preparaba un diseño informal, a menudo en forma de organigrama. Entonces se empezaba la codificación y se modificaba el diseño a medida que se aplicaba el sistema. Al acabar la etapa de aplicación, el diseño solía diferir tanto de las especificaciones iniciales que el documento del diseño original resultaba de una descripción totalmente inadecuada del sistema.

Este enfoque del diseño del software fue la causa de muchos fallos graves de caros proyectos. Ahora se ha comprendido que las notaciones completamente informales, como los organigramas, semejantes al lenguaje de programación, son vehículos inadecuados para formular y expresar el diseño del sistemas. Se ha reconocido que una especificación precisa (aunque no necesariamente formal) es una parte esencial del proceso de diseño y que el desarrollo del software es una actividad repetitiva de múltiples etapas que no puede representarse en una sola notación. En consecuencia, se han desarrollado varias notaciones de diseño como diagramas de flujo de datos, diagramas HIPO, diagramas de estructura y lenguajes para la descripción del diseño, que son más apropiados que los organigramas para expresar los diseños del software.

Dada una definición de requisitos, el ingeniero de sistemas debe utilizarla para desarrollar el diseño de un sistema de programación que satisfaga esos requisitos. Esto se realiza en varias etapas:

1. Deben establecerse los subsistemas que componen el sistema de programación.
2. Cada subsistema debe dividirse en componentes individuales y se ha de establecer la especificación de los subsistemas definiendo la operación de esos componentes.
3. Después cada programa se puede diseñar a base de subcomponentes que actúen recíprocamente.
4. Posteriormente hay que refinar cada componente. Esto suele implicar la especificación de cada componente como una jerarquía de subcomponentes.
5. En algún momento de este proceso de refinamiento hay que especificar con detalle los algoritmos utilizados en cada componente.

Además de estas etapas del diseño de sistemas de programación, el ingeniero de sistemas también debe diseñar los mecanismos de comunicación que permitan la interacción entre los procesos.

Tiene que diseñar estructuras de archivo y con toda seguridad deberá diseñar las estructuras de datos que se utilicen en sus programas. También tendrá que diseñar los casos de prueba para comprobar sus programas.

No hay una manera de establecer lo que se entiende por un "buen diseño". Dependiendo de la aplicación y los requisitos del proyecto particular, el buen diseño puede ser uno que permita producir una codificación muy eficiente, puede ser un diseño mínimo donde la aplicación sea lo más compacta posible o puede ser el diseño de más fácil mantenimiento. Un diseño mantenible implica minimizar el costo de los cambios del sistema, y eso significa que el diseño tiene que ser comprensible y que las modificaciones deben tener un efecto local. Ambas cosas se logran si el diseño del software es muy coherente y poco acoplado.

## **Proceso de diseño**

El diseño es una solución, es decir, es la traducción de los requerimientos en forma que los satisfagan. Las características de un nuevo sistema se establecen en los dos niveles de diseño: *lógico* y *físico*. El proceso de diseño describe el objetivo de cada nivel, la relación con el análisis de los requerimientos y cómo otras personas están involucradas en este aspecto del desarrollo del sistema.

## **Niveles de diseño**

El diseño de sistemas se presenta a través de dos fases: el *diseño lógico* y el *diseño físico*. Cuando los analistas formulan un diseño lógico, escriben las especificaciones detalladas del nuevo sistema; esto es, describen sus características: las salidas, entradas, archivos, base de datos y procedimientos, todos de manera que cubran los requerimientos del proyecto.

**Diseño lógico.** El diseño lógico de un sistema de información es como el plano de un ingeniero para armar un automóvil: muestra las características principales (motor, transmisión y área para los pasajeros) y cómo se relacionan con otras (donde se conectan entre sí los componentes o, por ejemplo, cuán separadas estarán las puertas).

Los informes y la producción del analista son los componentes de todo el mecanismo que emplea el ingeniero. Los datos y procedimientos se ligan y entonces se produce un sistema que trabaje.

Al diseñar un sistema de Activos Fijos, por ejemplo, las especificaciones de sistemas incluyen informes y definiciones de pantalla de salida que describen los activos existentes, las adiciones al mismo y las salidas, resumen las transacciones que ocurren a través de, por ejemplo, un mes de operación. El diseño lógico también especifica las formas de entrada y las descripciones de las pantallas para todas las transacciones y archivos a fin de mantener los datos del Activo, los detalles de las modificaciones y las bajas de artículos. Las especificaciones de los procedimientos describen métodos para introducir los

datos, corridas de informes, copiados de archivos y detección de problemas, si es que ocurren éstos.

**Diseño físico.** El diseño físico, actividad que sigue el diseño lógico, produce programas de software, archivos y un sistema en marcha. Las especificaciones de diseño indican a los programadores qué deben hacer al sistema. Los programadores a su vez escriben los programas que aceptan entradas por parte de los usuarios, procesan datos, producen los informes y almacenan estos datos en los archivos.

El diseño físico para el sistema de Activo Fijo, ya mostrado, consiste en pasos de programas escritos en un lenguaje de programación que cambian la ubicación de los artículos existentes, utilizan datos de transacción, imprimen los informes e imprimen y almacenan los datos.

El analista especifica en los algoritmos cómo cambiar la catalogación de los artículos. Durante el diseño físico, los programadores escriben las instrucciones de programa para calcular los cambios y producir los resultados.

### **Clasificación de metodologías de diseño**

El diseño efectivo del software se logra utilizando una metodología consistente de diseño. Hay una gran cantidad de metodologías de diseño desarrolladas y que se utilizan en diferentes aplicaciones. En esencia, la mayoría de estas metodologías se pueden clasificar en una de las tres áreas siguientes:

**Diseño funcional descendente.** El diseño descendente o refinamiento por pasos, se basa en la noción de que la estructura del problema debe determinar la estructura de la solución del software. El diseño de un sistema de software se efectúa realizando una descomposición funcional descendente. La metodología utiliza la característica humana más fundamental para la solución de problemas: la abstracción.

**Diseño orientado al objeto.** En vez de desarrollar el diseño de un sistema de software por medio de una descomposición funcional descendente, se dice que la mejor metodología es el diseño orientado a objeto. En este diseño, los componentes del software se ve más como objetos que como funciones.

Cada objeto tiene un conjunto asociado de operaciones permitidas, y los objetos se comunican mediante el paso de mensajes que, por lo general, incluyen una instrucción para activar una función determinada.

**Diseño controlado por los datos.** Esta metodología, propuesta por Jackson y Warnier, plantea que la estructura de un sistema de software debe



reflejar la estructura de los datos que éste procesa. Por lo tanto, el diseño del software se obtiene de un análisis de los datos del sistema de entrada y salida.

### **Formulación y descripción de un diseño de software**

El diseño del software es un proceso creativo que no se puede formular como un conjunto de reglas. Sin embargo, el uso de una metodología sistemática simplifica el proceso de diseño y da como resultado un software comprensible, comprobable y confiable.

La formulación y descripción de un diseño de software incluye varias etapas:

1. Estudiar y comprender el problema. Sin esta comprensión es imposible el diseño efectivo del software.
2. Identificar las características generales de, al menos, una posible solución. En esta etapa suele ser útil identificar varias soluciones y evaluar cada una de ellas. Se debe elegir la solución más simple posible. Es de particular importancia no permitir detalles de aplicación de bajo nivel, de los cuales tenga conocimiento el diseñador, que interfieran en la elección de una solución.
3. Construir un diagrama de flujo de datos que muestre las transformaciones generales de los datos del sistema. Si esto parece imposible, quizá no se haya comprendido de manera adecuada el problema.
4. Mediante el diagrama de flujo de datos, construir un diagrama de estructura que muestre las unidades de programa relacionadas con la solución.
5. Describir cada abstracción utilizada en la solución mediante un lenguaje de descripción. Es probable que en las primeras etapas del diseño esto consista casi exclusivamente en una descripción en lenguaje natural.

### **Notaciones del diseño**

Unas notaciones consistentes y completas son muy apreciadas en la creación de objetos abstractos como los sistemas de software. Sin tales notaciones, los diseños no se pueden evaluar, comparar, comprobar, probar o comunicar. Aunque el programa de computación es en sí mismo la especificación absoluta del diseño, el grado de detalle que se presenta en el programa es tal que resulta inadecuado para transmitir el diseño a los lectores.

Esto resulta especialmente cierto en los niveles más altos del diseño, donde un sistema grande se descompone en unidades funcionales como subsistemas o subprogramas. Es muy difícil expresar esto con claridad mediante un lenguaje de programación. Las notaciones descritas son:

reflejar la estructura de los datos que éste procesa. Por lo tanto, el diseño del software se obtiene de un análisis de los datos del sistema de entrada y salida.

### **Formulación y descripción de un diseño de software**

El diseño del software es un proceso creativo que no se puede formular como un conjunto de reglas. Sin embargo, el uso de una metodología sistemática simplifica el proceso de diseño y da como resultado un software comprensible, comprobable y confiable.

La formulación y descripción de un diseño de software incluye varias etapas:

1. Estudiar y comprender el problema. Sin esta comprensión es imposible el diseño efectivo del software.
2. Identificar las características generales de, al menos, una posible solución. En esta etapa suele ser útil identificar varias soluciones y evaluar cada una de ellas. Se debe elegir la solución más simple posible. Es de particular importancia no permitir detalles de aplicación de bajo nivel, de los cuales tenga conocimiento el diseñador, que interfieran en la elección de una solución.
3. Construir un diagrama de flujo de datos que muestre las transformaciones generales de los datos del sistema. Si esto parece imposible, quizá no se haya comprendido de manera adecuada el problema.
4. Mediante el diagrama de flujo de datos, construir un diagrama de estructura que muestre las unidades de programa relacionadas con la solución.
5. Describir cada abstracción utilizada en la solución mediante un lenguaje de descripción. Es probable que en las primeras etapas del diseño esto consista casi exclusivamente en una descripción en lenguaje natural.

### **Notaciones del diseño**

Unas notaciones consistentes y completas son muy apreciadas en la creación de objetos abstractos como los sistemas de software. Sin tales notaciones, los diseños no se pueden evaluar, comparar, comprobar, probar o comunicar. Aunque el programa de computación es en sí mismo la especificación absoluta del diseño, el grado de detalle que se presenta en el programa es tal que resulta inadecuado para transmitir el diseño a los lectores.

Esto resulta especialmente cierto en los niveles más altos del diseño, donde un sistema grande se descompone en unidades funcionales como subsistemas o subprogramas. Es muy difícil expresar esto con claridad mediante un lenguaje de programación. Las notaciones descritas son:

**Diagrama de flujo de datos.** Son diagramas que se utilizan para describir un diseño de sistemas de alto nivel, muestran cómo se transforman los datos al pasar de un componente del sistema a otro. Los diagramas de flujo de datos constan de tres componentes:

1. Flechas con anotaciones.
2. Burbujas con anotaciones.
3. Operadores \* y  $\oplus$ .

Las burbujas con anotaciones representan centros de transformación en los que la anotación especifica la transformación. Las flechas representan el flujo de datos hacia adentro y afuera de los centros de transformación, donde las anotaciones dan nombre al flujo de datos. Los diagramas de flujo o de datos describen como una entrada se transforma en una salida. Los operadores \* y  $\oplus$  se utilizan para unir flechas, \* significa AND ("y" lógico) y  $\oplus$  significa OR EXCLUSIVO ("o" Exclusivo lógico).

### **Diagramas de estructuras**

Los diagramas de estructura describen el sistema de programación como una jerarquía de partes y lo muestran gráficamente como un árbol. Estos diagramas documentan cómo se pueden aplicar los elementos de un diagrama de flujo de datos como una jerarquía de unidades de programa.

Un diagrama de estructura muestra la relaciones entre las unidades de programa sin incluir ninguna información acerca del orden de activación de esas unidades. Se trazan mediante tres símbolos:

1. Un rectángulo con el nombre de la unidad,
2. Una flecha que conecta los rectángulos,
3. Una flecha con un círculo, con el nombre de los datos que se pasan entre los elementos de diagrama de estructura . Las flechas con círculo suelen dibujarse paralelas a las flechas que conectan los rectángulos del programa.

### **Un lenguaje para la descripción del diseño**

Es una notación con algunos atributos de los lenguajes de programación, adecuada para describir operaciones de control y diseño detallado. Se puede usar en cualquier nivel de diseño del sistema.

El nivel más bajo de un diseño de software se describe mejor por medio de un lenguaje formal, se ha dicho que la notación más apropiada para esto es un lenguaje de programación de alto nivel. Si bien este enfoque tiene las ventajas evidentes de que el diseño es ejecutable si se dispone un compilador adecuado de lenguaje, también tiene las desventajas siguientes:

1. Los lenguajes de programación de alto nivel -al tener que ser compilables- no son fáciles de ampliar para incluir nuevos conceptos.
2. Los tipos de datos, las estructuras y las operaciones disponibles como primitivas en los lenguajes de programación, a menudo son de un bajo nivel relativo. Esto significa que la representación de algunas construcciones de alto nivel intuitivamente sencillas, por ejemplo una secuencia ilimitada de datos, a veces se torna detallada y confusa.
3. Puesto que el pensamiento está limitado por el lenguaje, a menor nivel de construcciones disponibles para el diseñador, mayor será la posibilidad de que su pensamiento se vea influido por las construcciones del lenguaje.
4. Si la aplicación inicial de un diseño, especificada en un lenguaje de programación, ha de volver a ser aplicado posteriormente, es difícil realizar esta aplicación en lenguaje de mayor nivel que el del lenguaje de especificación del diseño.

En vez de utilizar un lenguaje de programación existente como vehículo para expresar el diseño, una alternativa mejor es emplear un lenguaje para la descripción del diseño especialmente destinado a comunicar y documentar diseños de software.

En esencia, los lenguajes de descripción del diseño utilizan las familiares estructuras de control de los lenguajes de programación de alto nivel para especificar el flujo de control, pero permiten al diseñador una considerable flexibilidad en la descripción de las operaciones.

### **Confirmación del diseño**

La confirmación de un diseño de software es de gran importancia. Los errores y las omisiones no detectados que se arrastran hasta la fase de aplicación del proyecto y que no se detectan hasta la prueba del sistema, pueden resultar muy caros de corregir. Pueden necesitar el rediseño y la reaplicación de partes completas del sistema. El objeto de la confirmación de un diseño de software es conseguir dos cosas:

1. Mostrar que el diseño de software es correcto, esto es, el diseño debe aplicar de manera correcta la intención del diseñador. En ocasiones, este proceso se llama verificación, para distinguirlo del proceso más general de confirmación .
2. Mostrar que el software es válido, es decir, debe mostrarse que el diseño cumple con los requisitos en su totalidad. Cada requisito debe disponer de un fragmento de diseño que lo cumpla.

### 1.3.3 Desarrollo estructurado

La fase de desarrollo de sistemas tiene que ver con la traducción de las especificaciones de diseño a código fuente. El objetivo principal del desarrollo es el escribir código fuente y la documentación interna de modo que la concordancia del código con sus especificaciones sea fácil de verificar, y que se faciliten la depuración, pruebas y modificaciones. Este objetivo puede alcanzarse haciendo el código fuente tan claro y sencillo como sea posible. Sencillez, claridad, y elegancia son los sellos de los buenos programas; oscuridad, ingeniosidad, y complejidad son indicaciones de un diseño inadecuado y un pensamiento mal orientado.

La claridad del código fuente se mejora mediante técnicas de codificación estructurada, buen estilo de codificación, documentos adecuados de apoyo, buenos comentarios internos, y por las características que proporcionan los lenguajes de programación modernos.

#### Técnicas de desarrollo estructurado

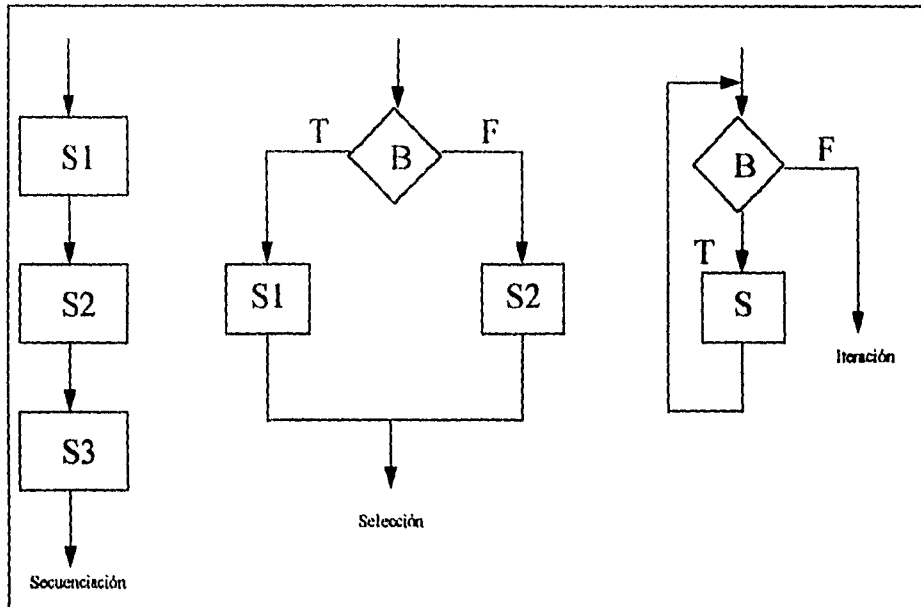
El objetivo del desarrollo estructurado es linealizar el flujo de control a través de un programa de computadora, de modo que la secuencia de ejecución siga a la secuencia en que está escrito el código. La estructura dinámica de un programa a medida que se ejecuta se parece entonces a la estructura estática del texto escrito. Esto mejora la legibilidad del código, lo cual facilita la comprensión, depuración, prueba, documentación y modificación de programas. También facilita la verificación formal de programas. El flujo de control lineal se puede lograr restringiendo el conjunto de construcciones de programas permitidos, a formatos de entrada única y salida única. ( Teorema de Bohm y Jacopini).

**Teorema de Bohm y Jacopini.** Cualquier segmento de programa con una entrada y una salida que tenga todas las proposiciones en algún camino de la entrada a la salida puede especificarse usando solamente secuenciación, selección e iteración.

Un conjunto suficiente de construcciones con una entrada y una salida para especificar el flujo del control en algoritmos es:

Secuenciación: S1; S2; S3;  
Selección: if B then S1 else S2;  
Iteración: while B do S;

La naturaleza de una entrada y una salida de estas construcciones se ilustra en la **figura 1.1**



**Figura 1.1.- Diagramas de flujo para codificación estructurada**

## **Recursividad**

Un subprograma recursivo es aquel que se llama a sí mismo, ya sea directamente o bien indirectamente; la recursividad es una técnica de programación poderosa y elegante. Cuando se le usa apropiadamente, los subprogramas recursivos son fáciles de entender, y son tan eficientes, si no es que más, que las instrumentaciones iterativas de algoritmos inherentemente recursivos. De esta manera mediante un uso apropiado de la recursividad se cumplen los objetivos de claridad y eficiencia.

La recursividad es una poderosa técnica de especificación, y en circunstancias apropiadas una poderosa técnica de instrumentación. Sin embargo una especificación recursiva no necesariamente implica una instrumentación recursiva. Las especificaciones funcionales establecen qué se requiere, no cómo alcanzarlo. La forma algorítmica de una especificación recursiva a menudo se expresa mejor en una manera iterativa.

## **Estilo en la codificación**

En programación, el estilo de codificación se manifiesta en las rutas que usa el programador para expresar una acción o un resultado deseado.

Los programadores que trabajan juntos pronto llegan a reconocer los estilos de codificación de sus colegas.

Se ha reconocido que un buen estilo de codificación puede superar muchas de las deficiencias de un lenguaje de programación primitivo, mientras que un estilo pobre puede frustrar los propósitos de un excelente lenguaje. El objetivo de un buen estilo de codificación es proporcionar un código fácil de comprender, sencillo y elegante.

**Acciones a seguir para un buen estilo de programación:**

1. Empléense unas cuantas construcciones estándar de control.
2. Utilícense las estructuras GOTO de manera disciplinada.
3. Introdúzcanse tipos de datos definidos por el usuario para modelar entidades en el dominio del problema.
4. Cúbranse las estructuras de datos bajo las funciones de acceso.
5. Aíslense las dependencias de máquina en unas cuantas rutinas.
6. Proporcionéense prólogos estándar de documentación para cada subprograma.
7. Examínese cuidadosamente la rutinas que tengan menos de 5 o más de 25 proposiciones.
8. Utilícense sangrías, paréntesis, espacios y líneas en blanco, y márgenes alrededor de los bloques de comentarios para mejorar la legibilidad.

**Acciones a evitar para un buen estilo de programación:**

1. No hay que ser demasiado complicado.
2. Evítense las proposiciones THEN nulas.
3. Evítense las proposiciones THEN\_IF nulas.
4. No se anide en forma muy profunda.
5. Evítense efectos colaterales oscuros.
6. No se suboptimice.

7. Examinense cuidadosamente las rutinas que tengan mas de cinco parámetros formales.
8. No se emplee un identificador para propósitos múltiples.

### **Estándares y directrices**

Los estándares de codificación son especificaciones para un estilo de codificación preferido. Dada una situación de elegir los caminos para lograr un efecto, se especifica un camino preferido. Los estándares de codificación a menudo son vistos por los programadores como mecanismos para restringir y devaluar las habilidades para resolver problemas creativos de los programadores.

Así, es deseable que todos los programadores de un proyecto adopten un estilo de codificación similar, de modo que se produzca un código de calidad uniforme. Esto no significa que todos los programadores deben pensar igual, o que deben instrumentar servilmente todos los algoritmos en la misma forma. En realidad, el estilo individual de cada programador en un proyecto puede identificarse aun cuando se observe un apego rígido a los estándares del estilo de programación.

### **Directrices de documentación**

La programación por computadora incluye el código fuente de un sistema y todos los documentos de apoyo generados durante el análisis, diseño, instrumentación, pruebas y mantenimiento del sistema. La documentación interna incluye prólogos estándar para unidades de compilación y subprogramas, los aspectos autodocumentados del código fuente, y los comentarios internos incrustados en el código fuente.

### **Documentos de apoyo**

Como documentos de apoyo se pueden utilizar las especificaciones de requisitos, documentos de diseño, planes de prueba, manuales de usuario, instrucciones de instalación y los reportes de mantenimiento.

Un enfoque sistemático al desarrollo de la programación garantiza que los documentos de apoyo se desarrollen de una manera ordenada, y que esos documentos se encuentren disponibles cuando se necesiten. En el enfoque adecuado para el desarrollo de la programación, la preparación de documentos de apoyo normalmente se difiere hasta que se termine la instrumentación del sistema.

Estos documentos deben desarrollarse como un producto natural paralelo al proceso de desarrollo. Las necesidades y restricciones del cliente se registran



en la especificación de requisitos; los requisitos proporcionan el marco de trabajo para el diseño estructural; el diseño detallado se desarrolla a partir del diseño estructural; el código fuente se desarrolla a partir de los diseños estructural y detallado. Los planes de prueba, manuales de usuario, programas de entrenamiento, instrucciones de instalación y procedimientos de mantenimiento evolucionan a través de todo el ciclo de desarrollo.

La calidad, cantidad, duración, y utilidad de los documentos de apoyo son las principales medidas de la salud y la bondad de un proyecto de programación.

### **Notas de unidad de programa**

Una unidad de programa es una unidad de código fuente que es desarrollada y/o mantenida por una persona, esa persona es la responsable de la unidad. En un sistema bien diseñado, una unidad de programa es un subprograma o grupo de subprogramas que cumplen una función bien definida o forman un subsistema bien definido. Una unidad de programa también es lo suficientemente pequeña y modular que puede ser probada totalmente en forma aislada por el programador que la desarrolla o modifica. Las notas de cada unidad de programa son utilizadas por cada programador para organizar sus actividades de trabajo y conservar la documentación de sus unidades de programa.

Cada unidad de programa y de compilación deberá contar con una documentación interna, consistente en un prólogo estándar, y de los aspectos autodocumentados del código fuente y los comentarios internos intercalados en la porción ejecutable del código.

### **1.3.4 Pruebas y liberación**

#### **El proceso de prueba**

La prueba de los programas es la parte del proceso de confirmación que suele realizarse durante la aplicación y también, en una forma distinta, cuando ésta ha terminado. La prueba consiste en ejercitar el programa utilizando datos similares a los datos reales que habrán de ser ejecutados por el programa, observar los resultados y deducir la existencia de errores o insuficiencias del programa, a partir de las anomalías de ese resultado.

Los sistemas grandes se componen de subsistemas formados por módulos que, a su vez, pueden componerse de procedimientos. Si se intenta probar el sistema como una sola entidad, es posible que no se identifique más que un pequeño porcentaje de errores. El proceso de prueba, al igual que el de programación debe avanzar en etapas, siendo cada una de ellas la continuación lógica de la etapa anterior.

en la especificación de requisitos; los requisitos proporcionan el marco de trabajo para el diseño estructural; el diseño detallado se desarrolla a partir del diseño estructural; el código fuente se desarrolla a partir de los diseños estructural y detallado. Los planes de prueba, manuales de usuario, programas de entrenamiento, instrucciones de instalación y procedimientos de mantenimiento evolucionan a través de todo el ciclo de desarrollo.

La calidad, cantidad, duración, y utilidad de los documentos de apoyo son las principales medidas de la salud y la bondad de un proyecto de programación.

### **Notas de unidad de programa**

Una unidad de programa es una unidad de código fuente que es desarrollada y/o mantenida por una persona, esa persona es la responsable de la unidad. En un sistema bien diseñado, una unidad de programa es un subprograma o grupo de subprogramas que cumplen una función bien definida o forman un subsistema bien definido. Una unidad de programa también es lo suficientemente pequeña y modular que puede ser probada totalmente en forma aislada por el programador que la desarrolla o modifica. Las notas de cada unidad de programa son utilizadas por cada programador para organizar sus actividades de trabajo y conservar la documentación de sus unidades de programa.

Cada unidad de programa y de compilación deberá contar con una documentación interna, consistente en un prólogo estándar, y de los aspectos autodocumentados del código fuente y los comentarios internos intercalados en la porción ejecutable del código.

### **1.3.4 Pruebas y liberación**

#### **El proceso de prueba**

La prueba de los programas es la parte del proceso de confirmación que suele realizarse durante la aplicación y también, en una forma distinta, cuando ésta ha terminado. La prueba consiste en ejercitar el programa utilizando datos similares a los datos reales que habrán de ser ejecutados por el programa, observar los resultados y deducir la existencia de errores o insuficiencias del programa, a partir de las anomalías de ese resultado.

Los sistemas grandes se componen de subsistemas formados por módulos que, a su vez, pueden componerse de procedimientos. Si se intenta probar el sistema como una sola entidad, es posible que no se identifique más que un pequeño porcentaje de errores. El proceso de prueba, al igual que el de programación debe avanzar en etapas, siendo cada una de ellas la continuación lógica de la etapa anterior.

### **En el proceso de prueba se pueden identificar cinco etapas:**

1. Prueba de funciones . La prueba de funciones o de unidades es el nivel básico de donde se prueban las funciones que componen un módulo para garantizar que operen de manera correcta.
2. Prueba de módulos. Un módulo se compone de varias funciones que pueden cooperar entre sí. Después de haber probado cada función individual, es necesario probar la cooperación de estas funciones cuando componen un módulo.
3. Prueba de subsistemas. Esta prueba es el siguiente paso del proceso en el cual los módulos se agrupan para formar subsistemas. Puesto que los módulos cooperan y se comunican, la prueba de subsistemas se debe centrar en la prueba de las interfaces de aquellos, dando por supuesto que los módulos son correctos.
4. Prueba del sistema. La prueba del sistema a veces llamada prueba de integración se lleva a cabo cuando se integran los subsistemas para conformar el sistema completo. En esta etapa, el proceso de prueba tiene que ver con el hallazgo de errores en el diseño y la codificación.
5. Prueba de aceptación. Hasta esta etapa, todas las pruebas se realizan mediante el empleo de datos generados por la organización encargada de construir el sistema. La prueba de aceptación del sistema se efectúa con datos reales: la información con la que el sistema deberá operar.

### **Pruebas descendentes y ascendentes**

La prueba descendente empieza al nivel de los subsistemas, representando los módulos como cabos; estos son objetos que tienen la misma interfaz que el módulo, pero que son mucho más simples. Después de completar la prueba de los subsistemas, se hacen lo mismo en cada módulo; las funciones se presentan en cabos. Por último las funciones son reemplazadas por el código real y se prueba éste.

La prueba ascendente invierte el proceso. Primero se prueban de manera individual las funciones que componen un módulo. Después se integran para formar un módulo y éste se aprueba. Una vez hecho esto con los módulos, éstos se integran y se aprueba el subsistema.

Si se usa la prueba descendente, es probable que los errores de diseño no vistos se detecten en una etapa inicial del proceso, estos errores suelen cometerse en los niveles más altos del sistema, si se detectan pronto, se puede ahorrar una buena cantidad de tiempo. La detección de errores antes que esté

muy avanzada la aplicación del sistema puede evitar una gran cantidad de rediseño y la reaplicación.

Además, la prueba descendente tiene la ventaja de disponer de un sistema que funciona, aunque sea de forma limitada, en esta etapa temprana del proceso de desarrollo. Lamentablemente, la prueba descendente estricta puede ser muy difícil, debido a la necesidad de producir los cabos de programas que simulan los niveles inferiores del sistema. Otro inconveniente es que el resultado puede ser difícil de observar.

La prueba ascendente, por otra parte, implica probar los módulos inferiores en la jerarquía y después trabajar hacia arriba, hasta llegar al módulo final. Las ventajas de la prueba descendente son los inconvenientes de la prueba ascendente y viceversa.

Si se usa la prueba ascendente para los módulos de menor nivel, hay que construir conductores que presenten las entradas adecuadas. El empleo de un enfoque de comprobación ascendente suele revelar que es más fácil crear casos de pruebas y observar las entradas de ésta. La prueba ascendente tiene el inconveniente de no disponer de un programa cuyo funcionamiento se pueda demostrar hasta haber probado el último módulo. Además, si existen errores de diseño en los módulos de alto nivel, se detectan hasta una etapa avanzada de la prueba del sistema. En la *figura 1.2.* se ilustra la diferencia entre estos dos enfoques.

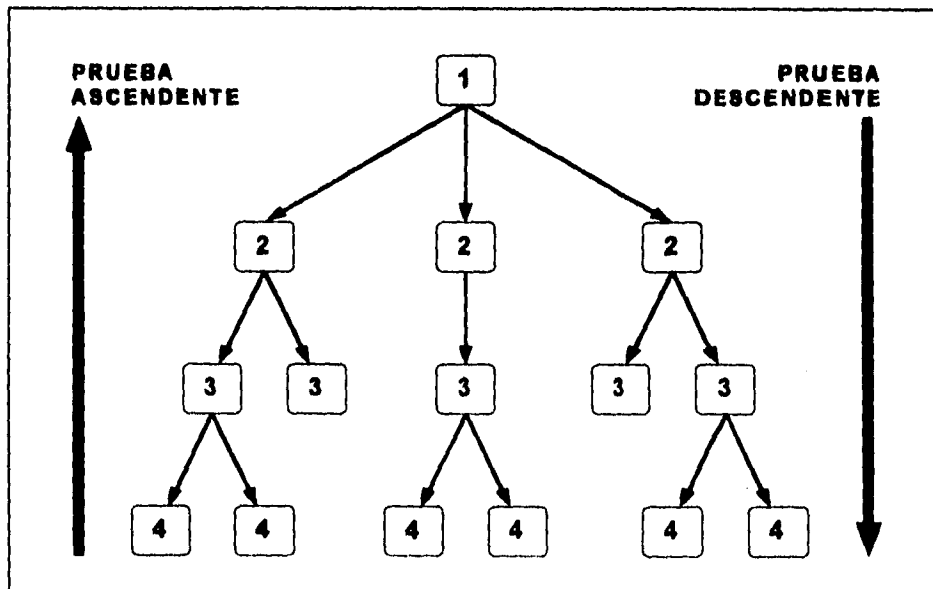


Figura 1.2. Prueba ascendente y descendente.

En vista de las ventajas e inconvenientes de cada método de prueba, no se puede hacer ninguna afirmación definitiva sobre cual es mejor para probar un programa.

### **Verificación de programas**

La noción de verificación de programas es atractiva. En vez de confirmar un programa mediante pruebas, la verificación implica demostrar con métodos matemáticos la correspondencia entre un programa y sus especificaciones. En esencia, se demuestra que el programa es correcto. Sin embargo, hay varios inconvenientes en verificar el programa y no el diseño, entre los cuales podemos destacar los siguientes:

- El programa puede ser escrito de manera que la verificación sea muy difícil. Se han podido haber utilizado construcciones dependientes de la aplicación.
- El lenguaje de programación utilizado puede ser de tan bajo nivel que haga imposible la verificación, no hay forma de probar que un programa en código de máquina es correcto.
- Si la verificación sigue a la aplicación y se descubren errores en el diseño, esto puede significar un considerable trabajo en el rediseño y la reaplicación del programa.
- Puesto que una aplicación suele ser más grande que un diseño, la verificación del programa es mayor, más compleja y más cara que la del diseño.
- Puede ser factible transformar automáticamente el diseño en varios programas, dependiendo del sistema requerido. Por tanto, al verificar el diseño, sólo se necesita una revisión de todos los programas generados.

Hasta el momento, la verificación de los programas sólo es rentable para partes muy críticas en sistemas muy grandes. Es poco probable que en un futuro próximo se realicen verificaciones normales de todo el sistema. Dado los costos y la dificultad que esto conlleva.

### **Inspección de código**

La inspección del código es una formalización de la programación sin ego, que implica que un programador guíe a un grupo de inspección por su código. Este proceso implica la distribución previa de las especificaciones del diseño al grupo de inspección, que las estudia e intenta comprender el diseño. Esto se puede complementar con una visión general presentada por el diseñador.

Es conveniente que en el primer grupo de inspección se integre por cuatro miembros, que tendrían las siguientes funciones:

- Un moderador que es un programador competente pero que no participa de forma personal en el proyecto.
- El diseñador del programa.
- El programador que participa en la aplicación.
- El responsable directo de probar el código.

Si el responsable del diseño y la codificación, o de la codificación y la prueba es el mismo individuo, se debe llamar a una persona ajena para realizar alguna de las funciones.

La ventaja de la inspección de código sobre la prueba, es que la primera encuentra muchos errores en una sola sesión. Cuando las pruebas descubren errores, a menudo se requiere correcciones antes de poder continuar con más pruebas.

### **Instrumentos de confirmación**

El proceso de confirmación de un sistema es laborioso y a menudo constituye la etapa más cara del desarrollo de un sistema, es por ello, que suele ser rentable utilizar al máximo instrumentos automatizados que ayuden en el proceso de confirmación.

Después de los instrumentos de aplicación, los de prueba y confirmación son quizás los más utilizados y mejor desarrollados. A continuación proporcionamos la descripción de algunos instrumentos de prueba.

- **Generadores de datos de prueba.** Son programas que generan en forma automática una gran cantidad de entradas de prueba para un sistema. Son muy útiles para situaciones en las cuales se debe probar el funcionamiento de un sistema en un ambiente práctico. Por ejemplo, la prueba de un sistema de administración de bases de datos puede empezar utilizando bases de datos muy pequeñas, las pruebas se diseñan inicialmente para detectar errores en los programas que originen una salida incorrecta.
- **Generadores de resúmenes del flujo de ejecución.** Son programas utilizados para analizar las veces que se ha ejecutado cada proposición en otro programa. A veces se les llama analizadores dinámicos y tienen dos partes fundamentales:

- ◊ Una parte que añade proposiciones de instrumentación a un programa durante la compilación o antes de ella. Cuando se ejecutan comparan información sobre las veces que se ha ejecutado una proposición.
- ◊ Una parte de despliegue, que reúne la información suministrada por la proposiciones de instrumentación y las imprime de una manera comprensible para el lector.
- **Comparadores de archivos.** Es un instrumento de propósito general que informa de las diferencias entre archivos y tiene un papel importante en la prueba del sistema. Como la actividad de probar suele implicar el examen de grandes volúmenes de salidas de prueba, debe automatizarse al máximo el proceso de revisión y comparación. La automatización implica preparar un archivo que contenga el resultado esperado de un programa si no se detectan errores. Entonces se ejecuta la prueba y la salida real se envía a algún otro archivo. Después se comparan ambos archivos por medio del comparador y se destacan las diferencias entre ellos. Si la salida esperada del programa y la real son iguales, la prueba no detectará ningún error.

### **Depuración de programas**

La depuración es el proceso de identificar las áreas del programa que causan errores y que deben ser modificadas para eliminar éstos. El proceso de depuración implica dos etapas; primero, se deben localizar aquellas partes incorrectas del código del programa y, segundo, el programa se debe modificar de manera que cumpla con sus requisitos.

Después de esto debe repetirse la prueba del programa para garantizar que la modificación se ha hecho de manera correcta.

### **Instrumentos de depuración**

Dado que existen problemas en la inclusión de proposiciones de salida, se han desarrollado varios instrumentos de depuración que realizan la misma función, a continuación se describen algunos de ellos.

- **Depuración por vaciado del núcleo.** En esencia es la impresión del contenido de la memoria de la máquina cuando se presenta el error en el programa.
- **Depuración por vaciado por símbolos.** Esta técnica puede tomar dos formas: una interactiva y otra por proceso de lotes. En esta última el sistema lista los nombre de todas la variables del programa junto con sus valores, ya sea al terminar el programa o con la conclusión específica de una instrucción. Un sistema así evita al programador incluir proposiciones de salida explícitas para buscar los valores de sus variables.

- **Depuración por rastreo del programa.** Son paquetes que proporcionan información acerca de la ejecución dinámica de un programa, estos paquetes imprimen información acerca de las entradas y salidas a los procedimientos, transferencias de control, selección de brazos en proposiciones if, etc.
- **Depuración por análisis estático de programas.** Estos analizadores no necesitan la ejecución del programa, dado que rastrean el texto y buscan anomalías que pudieran provocar errores.
- **Depuración por ambientes interactivos.** Estos sistemas dan al programador la impresión de que pueden interactuar con su programa mientras está en ejecución y puede desplegar valores de variables, devolver secuencias de ejecución, etc. De hecho, tales sistemas operan mediante la ejecución y la construcción de un archivo histórico que registra sus cambios de estado. También proporcionan la posibilidad de interrogar al archivo histórico desde el punto de vista del programa.

En los siguientes capítulos se presenta la aplicación del análisis estructurado para desarrollar el Sistema de Control de Activo Fijo para una Institución Bancaria.

## **1.4 Redes locales**

### **1.4.1 Introducción**

Con la invención de la microcomputadora en la década de los 70s se inició una revolución en el procesamiento electrónico de datos, creando la posibilidad de que cada persona pudiera tener una microcomputadora en su escritorio, para así procesar su información. El inicio de la década de los 80s fue la era del Floppy Disk (Disco Flexible), cada usuario tenía su información almacenada en sus propios discos. Los problemas se presentaron cuando esa información era compartida por varios usuarios, los discos se pasaban de mano en mano y debido a la poca capacidad del mismo, la información que se podía compartir era limitada. Con la invención del disco duro la capacidad de almacenamiento de las microcomputadoras aumentó. El costo y la capacidad de almacenamiento del disco duro influyeron en el nacimiento de las redes locales.

El uso de las microcomputadoras se ha ido incrementando fuertemente desde fines de los 70s. Esta explosión, particularmente en las PCs, nos ha llevado a que muchas empresas posean no sólo una sino varias.

Pero junto con la explosión de las microcomputadoras han venido aparejados algunos problemas inherentes al crecimiento: problemas en compartir



datos y programas, problemas al compartir ciertos dispositivos, falta de estandarización del software, poca seguridad de los sistemas y datos, falta de integración con los equipos mayores, etc. Todos estos problemas son resueltos a través de la denominada Red Local.

*Qué es una red:* Se puede decir que una red es un conjunto de computadoras enlazadas entre sí y/o con otros equipos, cuya configuración permita que éstos sean un medio para transmitir, recibir, compartir y manejar información.

*Qué hace una red:* Tiene como objetivo principal compartir recursos materiales (equipos y sus periféricos) y recursos informáticos (datos y programas), actualizándolos, organizándolos y explotándolos.

*Porqué una red:* Es la respuesta correcta a la necesidad de compartir entre usuarios, los recursos más costosos del equipo y la información centralizada y/o dispersa de una organismo, obteniendo con esto, la tan necesaria organización y economía en la informática.

Antes de hablar de las ventajas de una red local, revisemos la tres principales características que la definen:

- Los equipos que la forman deben estar enlazados por algún medio físico, ordinariamente cable.
- Al menos uno de los equipos debe ser capaz de compartir algún o algunos de sus recursos hacia los demás equipos.
- Todos los equipos que conformen la red deben estar físicamente cercanos (dentro de un edificio, o planta, o en un mismo campus).

#### **1.4.2 Ventajas de una red local**

A menudo la instalación de una red local se justifica inicialmente como un medio de compartir recursos (dispositivos periféricos). Pero según la red va creciendo y se integra en la organización, el compartir los recursos es insignificante en comparación con las demás ventajas de la conexión de redes. Las redes de área local enlazan también a las personas. El resultado es tanto una red electrónica como una red humana. Las redes locales proporcionan una herramienta efectiva para la comunicación entre los componentes del grupo de trabajo.

La conexión de redes puede ayudar a cambiar la forma en que la propia empresa conduce ciertas actividades de negocios:

El empleo lógico de los grupos de trabajo reduce la necesidad de las reuniones cara a cara. Igualmente, la conexión de redes permite incrementar la interacción entre los trabajadores desde su propia estación de trabajo y la optimización de la efectividad de la información, por que se tiende a poner más conocimiento en las comunicaciones escritas que en las informales.

Las redes locales al proporcionar un acceso directo a la información del grupo desde cada terminal, contribuyen a mejorar la productividad. Todos tienen acceso a los datos y por utilizar las herramientas disponibles en la red, pueden gestionar y compartir los resultados con otros.

Un beneficio colateral muy importante de las redes locales es que la lógica y los datos son mucho más fáciles de mantener y proteger que los de un entorno individual.

A partir de todo lo anteriormente expuesto, podemos identificar los siguientes seis grandes beneficios que nos proporciona el uso de una red local:

1. Permiten compartir datos y programas.
2. Ayuda a compartir recursos periféricos.
3. Brinda alta seguridad en el manejo de la información desde su sistema operativo.
4. Ayuda a estandarizar y reglamentar el uso del software.
5. Permite una mayor integración hacia los equipos grandes por medio de la utilización de enlaces múltiples (gateways).
6. Ayuda en la comunicación entre las personas de la institución.

### **1.4.3 Componentes de una red local**

El diseño de una red local nos permite personalizar la arquitectura de la misma para satisfacer las necesidades específicas de cada organización.

Esta flexibilidad se extiende no solamente a las aplicaciones que corren en su red, sino además al hardware y a las funciones que se pretenden implementar.

Hay que aclarar que algunas redes se diseñan para cumplir servicios relativamente simples, como puede ser compartir alguna aplicación, unos archivos o una impresora. Otras redes en cambio, soportan las comunicaciones con otros ordenadores (mainframes y/o micros), en función a esto se determinan los componentes necesarios y sus características particulares.

A continuación describimos brevemente algunos de los principales componentes a considerar en toda red local:

1. *Server*. El cual puede ser dedicado o no dedicado. Cuando es dedicado, se usa exclusivamente para administrar los recursos de la red; cuando es no dedicado, además de administrar los recursos, funciona como estación de trabajo.
2. *Estaciones de Trabajo*. Están representadas por cada una de las computadoras que están conectadas a la red, éstas pueden ser PCs.
3. *Tarjetas de Interface*. Instaladas dentro de cada computadora, según su especificación, cada tarjeta determina la forma de conexión. Existen tres tipos de tarjetas que dominan el mercado a nivel internacional: Arcnet, Ethernet y Token-ring.
4. *Canal de Comunicación*. Por lo general es un cable dedicado a las comunicaciones, mismo que puede ser:
  - Tipo telefónico
  - Par trenzado
  - Coaxial: broadband (lento, varios canales), baseband (rápido, un canal)
  - Fibra óptica (más rápido, varios canales)
5. *Repetidores*. En algunos casos los repetidores son necesarios por la distancia entre unidades de la red, mediante éstos se refuerza la señal, sin importar la topología, pueden ser tarjetas internas o cajas externas.
6. *Cajas de Conexión*. Por lo general son siempre necesarias en toda red.
7. *Sistema de Cableado*. Su forma de conexión en los equipos (topología) está en función de las tarjetas que se hayan seleccionado. Dependiendo del tipo de tarjeta se conecta por medio de un cable determinado.
8. *Sistema Operativo de Red*. Auxilia o sustituye al sistema operativo normal en el trabajo de compartir recursos, además, permitirá explotar ampliamente los recursos del server. Nosotros contaremos con un ambiente operativo Novell Netware v3.11 (50 usuarios).
9. *Software de Aplicación*. En última instancia, todos los elementos anteriores, son el fundamento para que el usuario de cada estación pueda utilizar sus programas, archivos y software específico.

#### 1.4.4 Topología de una red

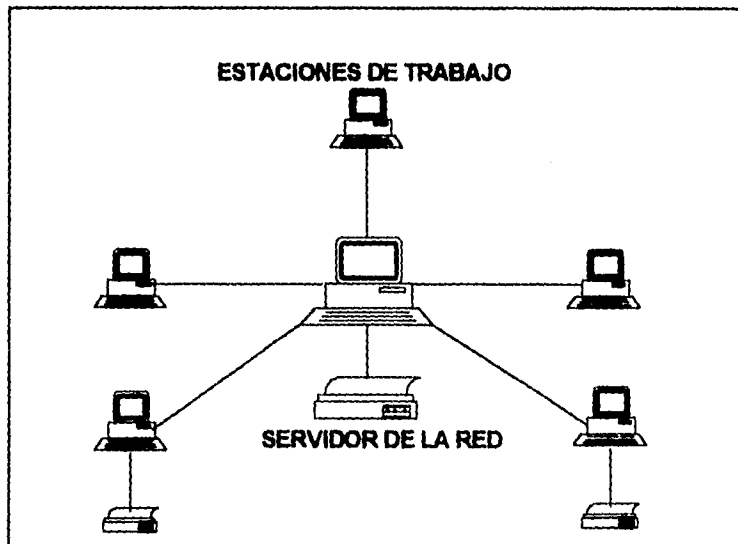
La topología de una red hace referencia a la ruta por la que discurren los datos a través de la red. Prácticamente existen cuatro tipos básicos de Topología, a saber.

- De Estrella
- De Bus
- De Anillo
- De Árbol

##### **Topología de Estrella**

En este tipo de conexión, el elemento central es el server con sus periféricos. El sever se mantiene preguntando constantemente a cada estación de trabajo, mediante comunicación exclusiva y por turno, si desea transmitir información, en caso de ser afirmativo, la atiende y al terminar, prosigue con otra su interrogatorio permanente.

En el caso de la regla de comunicación pregunta-respuesta-pregunta, conocida como protocolo polling (poleo) , ésta es empleada en gran medida en equipos minis. En el despertar de las redes locales, esta topología fue la que se utilizó; empero, resultaba una de las más caras. El diagrama conceptual de esta topología se muestra en la **figura 1.3**.

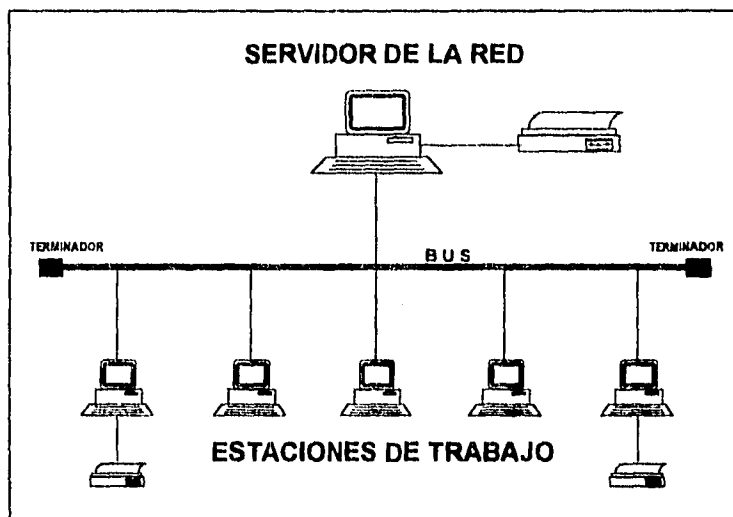


*Figura 1.3. Topología de Estrella*

### **Topología de Bus**

Esta conexión se considera que es la más sencilla de todas, donde las micros, incluyendo al server, están enlazadas por un solo cable (par roscado) , y la información viaja en ambos sentidos, por lo que es necesario prevenir las colisiones. Por ello el Protocolo apropiado es CSMA-CD (Carrier Sense Multiple Access-Collision Detection).

Con el protocolo CSMA-CD la RED transmite y espera a que se le confirme que la información fue recibida correctamente, de otra forma, detecta la posible colisión, espera un tiempo a que el canal esté desocupado y la información se transmite nuevamente. El diagrama conceptual de esta topología se muestra en la **figura 1.4.**



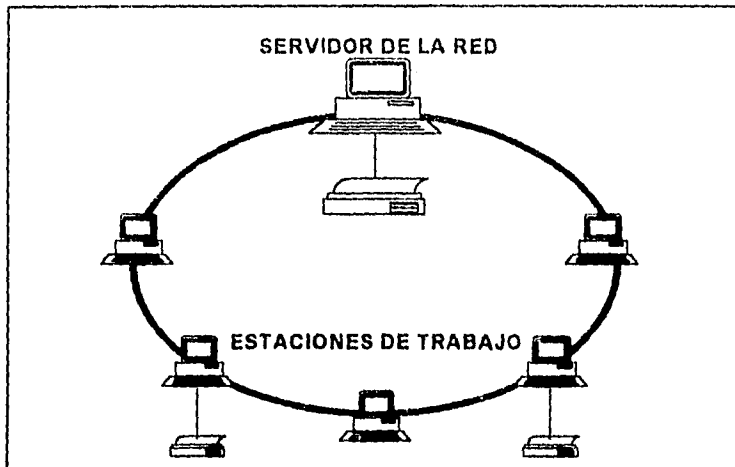
**Figura 1.4. Topología de Bus**

### **Topología de Anillo**

En esta conexión la información viaja ordenadamente en un sólo sentido a través de un sólo cable, descubriendo un ángulo de 360 en cuyo anillo imaginario, están conectadas en serie las estaciones de trabajo y el server.

Una señal llamada token ("receptáculo" a modo de estafeta), va circulando por la red y pasando por cada estación, si la primera resultó ser la solicitante, previa identificación entrega la información, de lo contrario la deposita en "sobre cerrado", para que ésta a su vez así la envíe a la siguiente, llevando consigna de entregarla hasta identificar a la solicitante.

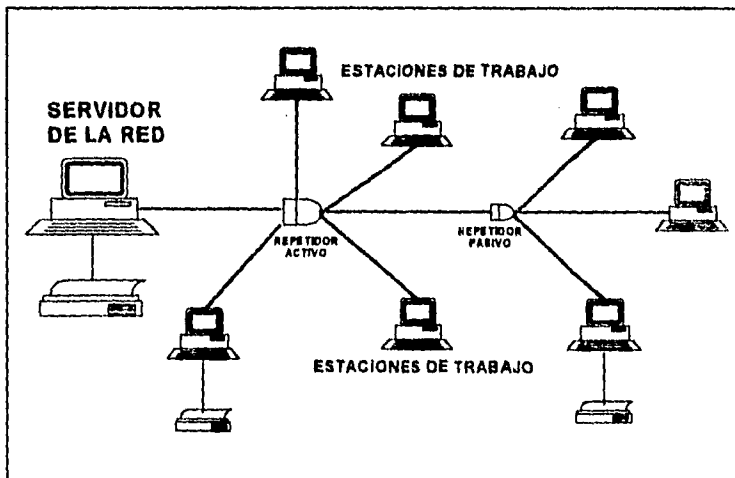
Cada estación de paso, cuando más, colecta información adicional enviándola a la siguiente y así se le pasa la señal cerrando ciclos "circulares", por ello el protocolo apropiado para este caso se conoce como token passing. El diagrama conceptual de esta topología se muestra en la *figura 1.5*.



*Figura 1.5. Topología de Anillo*

### **Topología de Árbol**

Esta conexión es combinada y es una opción más para implementar redes, según las necesidades del usuario. Normalmente trabaja con el Protocolo token passing, tarjeta ArcNet y conectores tanto pasivos como activos. El diagrama conceptual se muestra en la *figura 1.6*.



*Figura 1.6. Topología de Árbol*

### 1.4.5 Sistema de cableado

Existen básicamente cuatro tipos de cableado.

1. Par trenzado (UTP: Unshielded Twisted Pair).
2. Par trenzado blindado (STP: Shielded Twisted Pair).
3. Coaxial.
4. Fibra óptica.

Los tres primeros conducen la señal eléctrica a través de hilos de cobre. Los cables de fibra óptica transportan la luz a través de hilos de vidrio.

Los cables merecen una consideración mucho más seria de lo que continuamente suele percibirse. El cable que se seleccione ahora afectará a las futuras expansiones de la red. La mayoría de las redes admiten varias opciones de cableado. Se ha de comprender las posibles consecuencias al utilizar un determinado tipo de cable en la red.

En general, el orden en que están puestos respeta sus características ascendentes en cuanto a:

- Velocidad de transmisión.
- Precio.
- Distancia máxima.
- Dificultad en instalación.

En los últimos años, con la popularidad de ArcNet y Ethernet, el cable coaxial ha sido el más usado. Sin embargo, en los últimos meses, han aparecido modificaciones de ArcNet y Ethernet que utilizan twisted pair, por lo que algunos expertos le auguran un futuro muy promisorio al twisted pair.

Si bien el twisted pair normal es más barato que el coaxial, hay que tener en cuenta que cuando éste es blindado, su precio es superior al coaxial.

La mayor ventaja que podría tener el twisted pair, es aprovechar en cierto tipo de redes locales, el cable que ya se encuentra tendido a través de las oficinas. Pero en México, sinceramente, el aprovechar esa característica se ve difícil.

Por otra parte, la fibra óptica presenta grandes ventajas al tener mayores velocidades de operación, en general no se utiliza hoy en día por el tipo de dispositivos con los que se cuenta. Por ejemplo, si instalamos fibra óptica para una red ArcNet, la velocidad a la que transmitiremos seguirá siendo 2.5 Mbits-seg y no 100 o 200 Mbits-seg. que son velocidades a las que se pretende llegar.

#### **1.4.6 Esquemas de acceso a la red**

Cualquier topología que utiliza un medio común debe tener ciertos mecanismos para regular el acceso al mismo. De otra forma, nada prevendría el que dos PCs que hicieran una transmisión simultánea se bloquearan una a la otra.

Existen varios esquemas de acceso, pero dos son los más comunes en el medio, estos son: Carrier Sense Multiple Access, referido por sus siglas como CSMA, y el esquema de TOKEN PASSING. Los esquemas de acceso son parte de los niveles de hardware del modelo OSI. El esquema de CSMA se utiliza principalmente en Ethernet y el esquema de Token-Passing se utiliza en ArcNet y Token-Ring.

#### **CSMA**

En una red con acceso CSMA pasa algo similar. Cada uno de los equipos está escuchando hasta que el canal se encuentra sin transmisión (lo cual se detecta por un nivel de voltaje en el canal). Entonces empieza a transmitir, y si nadie empieza junto con él, la transmisión será exitosa. Si detecta que otro equipo está "hablando" al mismo tiempo (lo cual puede saber de acuerdo al nivel de voltaje de la línea) entonces se dice que ha ocurrido una colisión. Frena entonces su transmisión, espera nuevamente el silencio del canal, y vuelve a intentar transmitir su mensaje.

Apoyado en esta característica, cuando un usuario desea grabar un dato, Netware lo graba a un bloque en memoria principal y avisa al usuario que su pedimento de grabación ha sido hecho, y algunas fracciones de segundo después dicha grabación se realizará físicamente al disco.

Hasta ahora, el acceso CSMA ha sido el predominante en el mercado, situación que ha empezado a cambiar con la introducción de Token-Ring de IBM que utiliza Token Passing.

La mayoría de las redes que utilizan CSMA-CD incluyen también un método de detección de colisiones, para poder hacer la operación de escuchar y retransmitir en su caso, estos esquemas se conocen como CSMA-CD (Collision Detection)

#### **Token passing**

El "token" es en sí, un patrón especial de bits que está circulando continuamente por la red, indicando a todos los nodos que el canal se encuentra disponible. Para enviar un mensaje, la estación debe poner su mensaje en una forma apropiada, esperar el token y transmitir los datos.



El método de Token Passing posee diversas ventajas sobre los protocolos CSMA o CSMA-CD. Dado que una estación no puede enviar un mensaje a menos de que tenga el token, solamente un nodo a la vez puede transmitir. Esto elimina las colisiones de datos y la necesidad de los algoritmos de detención de colisiones y reenvío.

Las redes que utilizan Token-Passing son más fácilmente expandibles que aquellas que utilizan protocolos CSMA-CD. Conforme se van agregando más usuarios a un red CSMA-CD, la probabilidad de colisiones en la transmisión aumenta. Este problema puede ser resuelto parcialmente incrementando la velocidad de transferencia de los datos, pero muchas veces el algoritmo de detección debe ser muy versátil.

#### **1.4.7. Tarjetas de red**

Es difícil poder definir una sola tarjeta de red que resulte la mejor para todos los casos, de manera que en los párrafos siguientes, revisaremos algunas consideraciones en donde alguna de ellas resulta más eficiente o más adecuada que las otras, dependiendo de diversos factores. Sin lugar a dudas que uno de los factores críticos para redes medianas y grandes es el rendimiento de las mismas.

Comúnmente medimos el rendimiento en términos de velocidad de respuesta. En ese sentido, podríamos pensar que la red más rápida sea la que tiene mayor velocidad del canal(10,4,2.5 Mbits-seg), pero dicho rendimiento viene influenciado por muchas otras cosas. De la tarjeta de red, influye lo siguiente:

- *Forma de acceso.* Para cargas de trabajo medianas y pesadas, es decir mucho tráfico de muchos nodos, en general el protocolo CSMA se comporta menos eficiente que el Token Passing.
- *Tipo de comunicación entre tarjeta y CPU.* Existen tres tipos de comunicación. Memoria compartida, canal DMA y puertos de E-S. Dentro de estas tres formas, la más eficiente es la memoria compartida.
- *Eficiencia de los drivers.* Recordemos que el driver es la parte del software que dialoga directamente con la tarjeta. Cuando un fabricante diseña una tarjeta de red en la que implementa nuevos circuitos, o manejos internos distintos a los tradicionales, debe igualmente reescribir o rediseñar su driver, para que saque provecho de dichas potencialidades.

#### **Ethernet**

Ethernet surge a mediados de los 70s, como un desarrollo de los Laboratorios de Xerox (PARC), posteriormente se modifica con aportaciones de

DEC e INTEL, sufriendo algunas modificaciones los diseños originales, hasta que se define como un estándar (aunque el comité 802 de IEEE estableció el estándar 802.3, estrictamente no es igual a Ethernet aun cuando para fines prácticos sí se les considera iguales). Hemos resumido en el siguiente párrafo las principales características técnicas de Ethernet.

Características de la tarjeta: Existen algunas consideraciones muy especiales para ciertas tarjetas Ethernet. En el mercado se pueden adquirir tarjetas Ethernet pensadas en los slots de PCs y que por tanto se pueden insertar en una AT o incluso en un 386 (tarjetas de 8 bits) y tarjetas Ethernet pensadas especialmente para los slots de ATs (tarjetas de 16 bits). Algunos fabricantes les llaman Ethernet\_PLUS a los modelos especiales para ATs. Sin embargo, muchas de esas tarjetas especiales para ATs traen procesadores 80186, que ocasionan una baja en el rendimiento de la red.

### **ArcNet**

ArcNet es un estándar creado originalmente por Data Point Corporation. A partir de 1986, y apoyándose en las evaluaciones de la revista PC\_Magazine, el mercado de ArcNet se hizo aún más poderoso, y se calcula que al momento, existen aproximadamente alrededor de 2,000,000 de nodos de red utilizando ArcNet en todo el mundo (lo que le da una participación de alrededor del 17%). En el párrafo siguiente hemos resumido las características técnicas de ArcNet.

Características de la tarjeta: Las tarjetas ArcNet utilizan un microprocesador especial, COM9026, y para comunicarse con el CPU, utilizan memoria compartida (Shared Memory), que no tiene las desventajas de los canales DMA cuando se trabaja en equipos ATs. La tarjeta ArcNet para el microcanal de los equipos PS/2 tiene básicamente las mismas características, pero viene especialmente diseñada para dicho bus.

### **Token Ring**

Token-ring es un desarrollo hecho por IBM, quien definió la arquitectura general desde 1982. La tarjeta surge al mercado hasta 1987, pero aun desde esa fecha se considera un estándar, debido a que la IEEE lo define en su norma 802.5, y además, porque el patrocinio de IBM hace prever una importancia estratégica en ciertas corporaciones.

El desarrollo de la circuitería de la tarjeta lo hicieron IBM y Texas Instruments (TI) de forma conjunta, de manera que actualmente TI vende el conjunto de Chips (Chips Set TMS380) a todos los fabricantes que deseen construir su propia tarjeta Token Ring.

#### **1.4.8. Paquetes de software para red**

En las líneas siguientes describiremos los tipos más comunes de paquetes de software. Por desgracia, no existe una forma estándar ni de instalar los paquetes ni de manejar sus accesos (ni de fijar los precios) para cuando se ponen en una red local, de forma que el administrador o supervisor de la red deberá tener buen cuidado de leer las instrucciones necesarias.

Es muy importante también que el supervisor organice los directorios de tal forma que, cuando el usuario llame a aquellos programas para los que sí se tenga derecho, no existan problemas al crear sus archivos, o al utilizar caminos de búsqueda (comando Path de MS-DOS, o MAP de Netware).

#### **Procesadores de palabra**

En general los procesadores de palabra que trabajan en red local no permiten compartir un documento por más de un usuario. Lo que muchos hacen entonces, es un bloqueo al archivo, y cuando otra persona trata de utilizarlo, le avisa que ese documento está en uso.

Algunos productos como el Smart System permiten que varios usuarios accedan un documento a la vez, pero sólo el primero de ellos lo puede salvar con ese nombre. Los demás reciben un aviso de que el documento está en uso, y deberán guardarlo con otro nombre.

Word de Microsoft realiza bloqueo de archivos, y tiene una política muy sana, dá licencias adicionales por cada usuario.

#### **Hojas de cálculo**

Muchos de los paquetes de hojas de cálculo, encabezados por Lotus 1-2-3 y Excel de Microsoft, están moviéndose a poder trabajar en redes locales. La mayoría de ellos llegan a soluciones similares a los procesadores de palabra. Realizan un bloqueo por archivo y reconocen el tipo de estación desde el cual se llaman. Sin embargo existen algunos pocos paquetes, como el Open-Access II, que permiten que una hoja de cálculo sea utilizada por varias personas a la vez, implementando lo que han llamado bloqueo por grupos de celdas<sup>o</sup> (cell-locking). Esto, en algunos casos, permite una mayor versatilidad para los usuarios del paquete, aunque puede prestarse a cierta confusión.

La versión de red de Lotus 1-2-3 se denominaba Lotus-networker pero recibió críticas muy desfavorables por parte de los usuarios. Una de las principales fue su costo, la versión para 5 usuarios costaba 5 veces lo que la versión individual. La versión actual es mucho más sana en sus precios.

Excel de Microsoft, en cambio, tiene una cuota por licencia adicional razonable y acorde con el mercado. Adicionalmente ofrece la posibilidad de usar la liga dinámica entre aplicaciones, corriendo incluso en diferentes estaciones de trabajo.

### **Paquetes integrados**

La mayoría de los paquetes integrados abarcan las siguientes aplicaciones: Hoja de cálculo con gráficas, Procesador de palabras y Bases de datos.

En cuanto a las tres primeras, hoja de cálculo, gráficas y procesador de palabras, lo dicho en las dos secciones anteriores prácticamente se aplica al 100% a los paquetes integrados como son, Framework-III, Enable-LAN, System, Open-Access o Works.

### **Correo electrónico**

Una aplicación que ha hecho muy populares a las redes locales es el correo electrónico. Bajo un correo electrónico, un usuario es capaz de enviar y recibir mensajes, memorándums y todo tipo de documentos de otros usuarios. Adicionalmente, puede "pegar" cualquier tipo de archivo a sus mensajes.

Los límites de un correo electrónico van mucho más allá de la mera red local. De esta forma, utilizando módems y líneas telefónicas, es posible comunicarse con usuarios que se encuentren en redes de micros situadas en lugares distantes.

Muchos paquetes de correo electrónico para redes de micros han empezado a incorporar otras "herramientas" muy útiles, tales como la agenda (a diferencia de una agenda para una sola PC, las agendas en red permiten coordinar citas, juntas y otras actividades entre varios usuarios a la vez), listas de pendientes ("to do list"), recados telefónicos, etc.

### **Windows 3.1**

La nueva versión de Microsoft-Windows, además de tener un mucho mejor ambiente gráfico que la versión anterior está totalmente pensada en poder trabajar en redes locales. De hecho desde su instalación es configurada de acuerdo al tipo de red que se tenga.

### **Sistemas administrativos**

Además de los paquetes genéricos que se han mencionado antes, hoy en día casi todos los paquetes administrativos populares se han liberado también para red local.

En México, entre los paquetes más vendidos para red local están. Aspel-COI y SAE, Multivisión, MCBA entre los contables-administrativos, además de diversos paquetes para mercados verticales (hoteles, control de producción, etc).

## **Bases de datos**

Sin lugar a dudas que una de las principales aplicaciones para manejar en una red de microcomputadoras son las bases de datos. En Estados Unidos son la segunda en importancia después del correo Electrónico, y en nuestro país están actualmente a la cabeza.

En las siguientes líneas, trataremos de hacer una breve revisión de diferentes aspectos sobre las bases de datos y más correctamente sobre los manejadores de bases de datos (DBMS por sus siglas en inglés).

Definiremos en forma correcta lo que es un manejador de base de datos y lo que significa el enfoque relacional, hablaremos sobre los dos principales estándares actuales del mercado, dBase y SQL, para discutir entonces, las características más importantes para un manejador de red.

*Qué es un manejador de bases de datos:* Un manejador de Bases de datos es, informalmente, un programa (o conjunto de programas) que actúan como intermediario entre el usuario y los datos. Los datos tomados como un conjunto de archivos estructurados y relacionados de alguna forma se denominan base de datos.

Dependiendo del nivel y tipo de los usuarios, el manejador de bases de datos o DBMS, permitirá diferentes tipos de lenguajes para accederlo.

Al usuario inexperto le interesa un lenguaje sencillo de manejar, para poder consultar la información, y posiblemente, emitir ciertos reportes. A este lenguaje se le llama normalmente Query Language o simplemente Query.

Al usuario-programador le es muy útil una interfaces de programación, o todo un lenguaje, desde donde pueda realizar programas para dar de alta la información de los archivos, para consultarla y darle mantenimiento. A este tipo de lenguaje se le llama normalmente DML (Data Management Language).

Y por último, al encargado de organizar la información, de permitir o restringir el acceso a los datos, es decir, el Administrador de la base de datos, le interesará el poder definir como estarán formados los archivos, qué relaciones tendrán, etc. A esta visión general de los archivos que formarán toda una base de datos es común que se le llame esquema. Y al lenguaje con el cual podemos definir un esquema se le llamará DDL (Data Definition Language).

### **1.4.9 Sistemas operativos para red**

Partiendo de la definición de un sistema operativo como un "manejador de recursos", podemos decir que el objetivo de un sistema operativo de red, genéricamente llamado NOS, por sus siglas en inglés, es poder administrar correctamente estos recursos. Enumeramos a continuación las principales tareas de un sistema operativo:

- ◆ Administración del server.
  - Quién puede entrar y con que derechos.
  - Quién puede definir derechos a los demás.
  - Cómo organizar los discos duros.
  - Cómo se maneja la cola de impresión.
  - Facilidades de respaldo de información.
  
- ◆ Protección en caso de fallas.
  
- ◆ Reportes de control y facilidades de monitoreo.
  
- ◆ Facilidades de comunicación (avisos).

Con base en este resumen de características, comentaremos los detalles más sobresalientes de los sistemas operativos de IBM, Novell y Microsoft.

#### **IBM PC-LAN Program**

Tanto este sistema operativo como algunos otros (3com, 3+share) derivan de un producto que desarrolló Microsoft para redes locales: Microsoft-networks o simplemente MS-Net.

La versión actual elimina muchas limitantes de la anterior, pero sigue adoleciendo de otras tantas. El software se instala sobre MS-DOS, de manera que el manejo de archivos es nativo de MS-DOS, y la limitante de 640 Kb sigue existiendo.

El server generalmente es no dedicado, esto aunado a varias deficiencias en su diseño ocasionan un rendimiento realmente pobre.

La seguridad del paquete son las mínimas, a partir de la versión 1.3 ya reconoce usuarios y grupos de recursos (dominios), de forma que existe un poco más de administración que en las anteriores.

El manejo de cola de espera es aceptable, al igual que su sistema de mensajes. Otra desventaja del producto es sus consumo de memoria, tanto en el server como en las estaciones de trabajo.

Trabaja sobre NETBIOS, y aunque IBM sólo lo soporta para sus tarjetas de red (Broadband, baseband y token-ring), es factible utilizarlo en otras muchas tarjetas de estos tipos.

### **Novell-Netware**

Casi desde los inicios de este producto, a fines de 1983, Novell ha adquirido una alta reputación en el mercado de redes locales. Hoy en día es el número 1 en sistemas operativos, tanto en ventas como en potencia y rapidez.

Las característica más ventajosa de Novell-Netware es que puede trabajar con un sin fin de tarjetas de red, de manera que la elección del sistema operativo no queda atada a un cierto tipo de tarjeta. Como utiliza servers que tengan procesador 80286 o mayor, trabaja en modo protegido, permitiendo hasta 15 Mbytes de memoria en el server.

Permite definir una estructura de usuarios y grupos de usuarios, y el supervisor determina los derechos que tendrán cada directorio (ya sea por grupo o por usuario individual). Netware permite declarar hasta 8 derechos distintos:

R	Read	Leer archivos
W	Write	Escribir archivos
O	Open	Abrir archivos
C	Close	Cerrar archivos
D	Delete	Borrar Archivos
S	Search	Buscar en un directorio
P	Parental	Declarar subdirectorios
M	Modify	Modificar atributos de archivos

Para mayor facilidad, los derechos pueden ser restringidos desde cada directorio. Además de las restricciones de acuerdo a los 8 derechos, a partir de la versión 2.1 de Netware, el supervisor puede controlar diversas funciones adicionales de seguridad: restringir el acceso de acuerdo a cada estación de trabajo y por horario, hacer cobros y auditar por tiempo de conexión, bloques grabados o leídos, etc.

### **LAN MANAGER de Microsoft**

A fines de 1988 Microsoft lanzó al mercado el LAN-Manager un nuevo sistema operativo, totalmente distinto a su anterior MS-Net. LAN-Manager, como Netware, corre en modo protegido en el server, de forma que requiere de procesadores 80286 o mayores. Adicionalmente utiliza OS/2 como su núcleo para manejo de archivos, declaración de procesos, etc.

Trabaja sobre NETBIOS, y aunque IBM sólo lo soporta para sus tarjetas de red (Broadband, baseband y token-ring), es factible utilizarlo en otras muchas tarjetas de estos tipos.

### **Novell-Netware**

Casi desde los inicios de este producto, a fines de 1983, Novell ha adquirido una alta reputación en el mercado de redes locales. Hoy en día es el número 1 en sistemas operativos, tanto en ventas como en potencia y rapidez.

Las característica más ventajosa de Novell-Netware es que puede trabajar con un sin fin de tarjetas de red, de manera que la elección del sistema operativo no queda atada a un cierto tipo de tarjeta. Como utiliza servers que tengan procesador 80286 o mayor, trabaja en modo protegido, permitiendo hasta 15 Mbytes de memoria en el server.

Permite definir una estructura de usuarios y grupos de usuarios, y el supervisor determina los derechos que tendrán cada directorio (ya sea por grupo o por usuario individual). Netware permite declarar hasta 8 derechos distintos:

R	Read	Leer archivos
W	Write	Escribir archivos
O	Open	Abrir archivos
C	Close	Cerrar archivos
D	Delete	Borrar Archivos
S	Search	Buscar en un directorio
P	Parental	Declarar subdirectorios
M	Modify	Modificar atributos de archivos

Para mayor facilidad, los derechos pueden ser restringidos desde cada directorio. Además de las restricciones de acuerdo a los 8 derechos, a partir de la versión 2.1 de Netware, el supervisor puede controlar diversas funciones adicionales de seguridad: restringir el acceso de acuerdo a cada estación de trabajo y por horario, hacer cobros y auditar por tiempo de conexión, bloques grabados o leídos, etc.

### **LAN MANAGER de Microsoft**

A fines de 1988 Microsoft lanzó al mercado el LAN-Manager un nuevo sistema operativo, totalmente distinto a su anterior MS-Net. LAN-Manager, como Netware, corre en modo protegido en el server, de forma que requiere de procesadores 80286 o mayores. Adicionalmente utiliza OS/2 como su núcleo para manejo de archivos, declaración de procesos, etc.



Algunas personas critican la unión de OS/2 - LAN Manager, comentando que ello tiene dos desventajas muy fuertes: Por una parte el consumo de memoria (un server requiere fácilmente arriba de 4 Mb para empezar a operar). Por la otra, el rendimiento no se espera muy alto, debido a todas las capas de software que se tendrán que atravesar para realizar ciertas tareas. Pese a lo anterior se espera que LAN Manager vaya ganando terreno, y que empiece a ser un competidor real de Novell-Netware.

Debido a que corre sobre OS/2, y a que tiene un protocolo de red (Named-Pipes) muy enfocado a realizar fácilmente procesos cooperativos, la mayoría de los servidores de bases de datos en el mercado tienen versiones para LAN-Manager. Internamente, además de trabajar en modo protegido, tiene muchas semejanzas con Netware, sobretodo en los manejos internos de memoria (file caching) para agilizar el acceso a disco duro.

La versión 2.0 de LAN-Manager está pensada en posicionarse frente a frente con Netware 386, Si bien LAN-Manager no requiere forzosamente correr en un servidor 386, sí lo aprovecha, instalando un sistema de archivos particularmente diseñado para el 386 y 486 : HPFS o "high performance file system". En esta versión se soportarán perfectamente los protocolos típicos: NETBIOS, Named-Pipes, TCP/IP e incluso TP4 de OSI.

Tanto en la parte de seguridad, tolerancia a fallas, como en la de rendimiento, se espera que entre LAN-Manager 2.0 y Netware 386, las situaciones sean muy parejas.

Hasta aquí se termina con toda la teoría que en los siguientes capítulos será utilizada para desarrollar el sistema de activos fijos.

## CAPITULO 2 ANÁLISIS ESTRUCTURADO

En este capítulo se describe el flujo de información que existe entre los departamentos que intervienen en el control y registro de los bienes muebles y equipo.

Con base en la información obtenida se presentará una propuesta de automatización integral de las áreas involucradas, se definirán los alcances del sistema y los recursos de hardware y software seleccionados.

### 2.1 Definición del problema

La relación entre los departamentos que intervienen en el control y registro de los bienes muebles y equipos se muestra en forma gráfica en la *figura 2.1*.

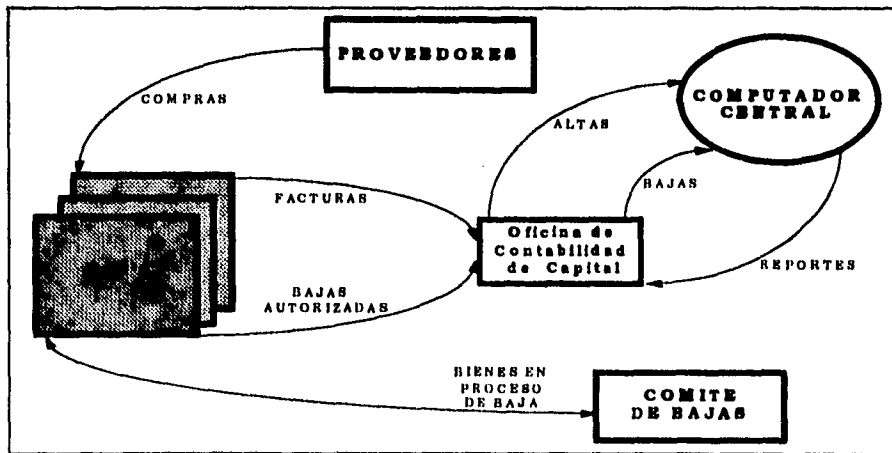


Figura 2.1. Relación de departamentos involucrados

Los departamentos involucrados y sus funciones son los siguientes:

- **Oficina de contabilidad de capital**

Tiene a su cargo la custodia de las facturas de los bienes muebles y equipos que se adquieren y es la encargada de hacer las afectaciones contables correspondientes.

De igual manera, los bienes que causan baja deben tramitarse a través de esta dependencia, ya que el hecho de dar de baja un bien requiere de ajustes contables, y en algunos casos hay recuperación por venta de equipos usados.

El control del activo fijo es responsabilidad directa de esta oficina, por lo cual debe concentrar toda la información requerida para esa tarea. La información llega en forma de facturas de compra que se obtienen a través de las dependencias encargadas de efectuar las compras de mobiliario y equipo; por otro lado, estas dependencias se encargan también de proponer las bajas de bienes obsoletos o dañados, y una vez autorizadas, las propuestas de bajas también son entregadas a la oficina de Contabilidad de Capital.

- **Centros de información presupuestal (CIP's)**

Los Centros de Información Presupuestal que existen actualmente dentro de la Institución son ocho, y tienen como función el ejercicio y control del presupuesto asignado para satisfacer los requerimientos de equipos, necesarios para el desempeño de las funciones en las diferentes áreas de la institución.

Cada uno de los CIP's cumple funciones específicas de acuerdo con el área que debe atender. Los requerimientos de equipos y bienes que los diferentes departamentos de la institución transmiten a los CIP's, son la base para que estos efectúen las compras con los diferentes proveedores que existen en el mercado. Los CIP's que actualmente existen dentro de la institución son los siguientes:

*CIP de Informática.* Tiene como función la de atender los requerimientos de equipos de microcomputación, periféricos, insumos, comunicaciones, telecomunicaciones, así como el Centro de Cómputo de la Institución.

*CIP de Seguridad.* Su función se centra en los aspectos de seguridad dentro de las instalaciones de la institución y tiene a su cargo los equipos contra incendio, armamento, sistemas de circuito cerrado, diverso equipo de laboratorio, transporte blindado y sistemas de comunicación interpersonal.

*CIP de Administración Inmobiliaria.* Tiene a su cargo el mobiliario de la institución, tanto en su compra y distribución como en su mantenimiento.

*CIP de Fábrica.* Tiene a su cargo la compra de maquinaria, insumos, refacciones, equipos y lo necesario para la fabricación de billetes.

*CIP de Caja.* Tiene a su cargo el suministro y mantenimiento de maquinaria y equipos para la distribución, recolección, recuento y destrucción de billetes. Su campo de acción se limita a las áreas de caja de la institución.

*CIP de Administración.* Su función se enfoca a satisfacer las necesidades de la institución en cuanto a equipos de comedor, equipo médico, equipos para la capacitación del personal, y diversos insumos.

*CIP de Inmuebles.* Tiene a su cargo la supervisión de obras en construcción, mantenimiento y acondicionamiento a inmuebles del banco.

*CIP de Autotransportes.* La compra, mantenimiento y acondicionamiento del equipo de transporte terrestre (automóviles, camionetas, motocicletas) son responsabilidad de este CIP.

Todos los CIP's anteriormente mencionados tienen a su cargo el hacer las convocatorias para la licitación y compras de bienes y equipos. Inicialmente, al hacer las compras, se concentra el pedido en los almacenes de los CIP's para su posterior distribución.

Es obligación del CIP hacer la entrega de las facturas de compra correspondientes a la oficina de Contabilidad de Capital, la cual asigna a los CIP's los bienes comprados en el sistema de registro de mobiliario y equipo en el computador central.

También es tarea de los CIP's concentrar en sus almacenes los bienes y equipos dañados u obsoletos para proponer las bajas que deben efectuarse. Para ello, debe turnarse la propuesta al comité de bajas de la institución, la cual se encargará de dar la autorización correspondiente. Una vez hecha la autorización, el CIP debe turnarla a la oficina de Contabilidad de Capital para efectuar los movimientos y afectaciones correspondientes.

#### • **Computador central**

El Computador Central de la Institución es un equipo Unisys 2200 que se encuentra físicamente ubicado en las instalaciones remotas. Ofrece a los usuarios sesiones para trabajar en demanda o en Batch. Los recursos con los que cuenta en cuanto a capacidad de almacenamiento y procesamiento son limitados, ya que la gran cantidad de usuarios que se conectan y el número de procesos que se ejecutan muchas veces provocan que haya sobrecarga en el computador central, lo que trae como consecuencia que se lleguen a bloquear algunos de los trabajos activos.

Debido a que los enlaces con el computador central se dan en su mayoría a través de líneas telefónicas, es frecuente encontrar fallas en la comunicación debido al mal estado en que se encuentran esas líneas.

En la actualidad, la oficina de contabilidad de capital está registrada como un usuario en el computador central y en él puede efectuar el registro del mobiliario y equipo de la institución, así como el registro de la contabilidad de la institución, entre otras aplicaciones.

#### • **Comité de bajas**

El comité de bajas de la Institución tiene como función la de dictaminar si proceden las peticiones de los CIP's, para dar de baja los bienes que ya se encuentran obsoletos o maltrechos, así como determinar los precios de venta de esos bienes en caso de que se pueda recuperar alguna cantidad de ellos.

Esto se da principalmente en el caso de automóviles, que se subastan, y de equipo electrónico, que se subasta en lotes. Finalmente, el comité de bajas notifica a la oficina de contabilidad de capital para que ésta efectúe las bajas correspondientes, tanto en el sistema de registro de mobiliario y equipo como en la contabilidad.

#### **2.1.1 Procedimientos existentes**

La oficina de Contabilidad de Capital es la encargada del registro y control de los activos fijos de la Institución. Los medios con los que cuenta para realizar esta tarea son:

- ◊ Tiene bajo su custodia las facturas de los bienes y equipos que se adquieren por parte de los CIP's.
- ◊ Posee un sistema para el registro de los bienes adquiridos en el computador central de la Institución. El registro de altas se da en base a las facturas que les son entregadas por los CIP's.
- ◊ Del sistema de registro obtiene reportes para checar que la información capturada sea correcta.
- ◊ Recibe del Comité de Bajas la información para dar de baja artículos del sistema de registro.
- ◊ Contabiliza las altas y las bajas.

## 2.1.2 Identificación del problema

En la Actualidad se tienen graves problemas en cuanto al control físico de los bienes y equipos propiedad de la institución. Ocurre que los activos que se adquieren simplemente se registran en el computador central de la institución, considerando que la ubicación física de los artículos está en los departamentos que efectuaron las adquisiciones. Además de esto, se hacen las afectaciones contables correspondientes por parte de la oficina de contabilidad de capital. No existe una retroalimentación que permita ubicar físicamente a los artículos, llegando en ocasiones al extremo de no saber si un artículo en particular aún existe dentro de la institución.

Por otro lado, el sistema informático de registro de mobiliario y equipo utilizado en la actualidad resulta ya inadecuado para las necesidades de la oficina de contabilidad de capital, ya que se trata de una captura que se accesa a través de una terminal remota y que ofrece poca flexibilidad en el registro de la información, además de que la validación de la misma es mínima y de que la salida del sistema está hecha en base a reportes.

Otro inconveniente del sistema de registro usado actualmente es que tiene una documentación prácticamente nula, ya que sólo se cuenta con el código fuente (COBOL) de los programas. Esto provoca que el mantenimiento se convierta en una tarea extenuante para el área de informática, de manera que se considera más conveniente desarrollar un nuevo sistema de control y registro del mobiliario y equipo de la Institución.

## 2.2 Información recabada

Para efectuar este nuevo sistema se procedió a efectuar un levantamiento de información, donde el usuario manifestó durante las entrevistas los siguientes requerimientos:

- El sistema deberá correr en un ambiente multiusuario, en una red Novell y se deberá apegar al software homologado por la Institución Bancaria.
- Acceso a la información desde pantalla o a través de reportes.
- La necesidad de validar la información que entre al sistema. Para ello se le propone al usuario el uso de catálogos para aceptar solamente artículos catalogados.
- Ayuda para la captura de la información con teclas de función. Por ejemplo, ayudar a la codificación de los artículos, ya que se tiene la idea de trabajar directamente con las facturas de los bienes como documentos fuente para la

captura. De esta manera se evitaría un paso previo de codificación y llenado de formatos de captura. Esta ayuda consistiría en oprimir una tecla para abrir una ventana con los posibles valores que pueda tener el campo de captura a llenar.

- Al seleccionar un valor de los que se muestren, el sistema debe ser capaz de llenar el campo con el valor seleccionado en la ventana de ayuda.
- También se plantea la necesidad de parte del usuario de tener flexibilidad en el manejo de los reportes, por lo cual se propone que existan reportes fijos (los usados con mayor frecuencia) y reportes que el propio usuario pueda armar según sus necesidades.
- Se requiere que la actualización de los catálogos que se usen en el sistema se pueda dar en línea desde las pantallas de captura. Esto haría posible que, por ejemplo, se diera una alta a un catálogo a través de una ventana sin tener que abandonar la pantalla de captura y cancelar el movimiento que se estaba registrando.
- En algunos casos sería necesario buscar la codificación de un artículo en base a su nombre, por lo que se requiere que el sistema permita la búsqueda de cadenas de caracteres dentro de un catálogo.
- Se requieren reportes mensuales de bajas, altas y cambios al archivo maestro en un formato fijo.
- También se requiere de reportes fijos de los catálogos a usar en el sistema.
- En todo momento deben poder visualizarse en pantalla a través de ventanas los diferentes catálogos del sistema, pulsando teclas de función específicas para cada uno de ellos.
- El sistema debe ser capaz de aceptar altas de artículos individuales o bien dar la opción de registrar una factura que contenga cualquier cantidad de artículos, en cuyo caso, el importe de los artículos individuales sería el importe total de la factura dividido por el total de artículos.
- En el caso de bajas y cambios, sería conveniente capturar ciertas características de artículos a dar de baja, y en base a ello crear listas de artículos que cumplan con esas características para poder elegir de esa lista el artículo a dar de baja o a modificar, esto debido a que pueden existir artículos con características exactamente iguales.
- Se tiene también la necesidad de trabajar el sistema en ambiente de red para facilitar las labores de captura entre varios usuarios a la vez.

- Es necesario tener una bitácora en papel de todos los movimientos que se generen desde la captura y actualización, por lo cual se debe contar con una impresora por cada uno de los nodos que se encuentren en sesión de trabajo de captura.
- El sistema debe ser capaz de identificar cuántos artículos se han adicionado o dado de baja, así como su importe, en una sesión de captura.
- Para ello se debe contar con una opción que permita validar estas cifras que servirán como control de lo capturado en una sesión de trabajo.
- El sistema debe permitir respaldar la información en discos flexibles.
- También se debe tener una opción para la emisión de resguardos de mobiliario y equipo por departamento para tener un control en cuanto a los movimientos que puedan tener las diferentes dependencias de la institución. Estos resguardos actuarían como documentos oficiales amparando los bienes asignados a cada departamento, los cuales además estarían sujetos a auditoría. Esto obligaría a reportar a la Oficina de Contabilidad de Capital los movimientos ocurridos en la asignación de bienes en un período que se fijaría en un mes.

De acuerdo con los requerimientos del usuario se necesita desarrollar un sistema con manejo de menús, ventanas, pantallas de captura y manejo de archivos. De acuerdo con los planteamientos, es conveniente que el sistema se desarrolle en un ambiente de computadoras personales en red, debido a la flexibilidad que ofrecen en cuanto a disponibilidad de productos para desarrollo de programación y de manejadores de archivos.

Por otro lado, se debe contar con los equipos necesarios para el desarrollo de la programación y sus pruebas, y para la posterior instalación y liberación del sistema, los cuales comprenden computadoras personales, impresoras de matriz para la bitácora en línea y los reportes, impresoras láser para la emisión de resguardos de mobiliario y equipo, acceso a una red de área local para trabajo multiusuario y paquetería para el desarrollo del sistema.

### **2.3 Análisis de costos**

La estimación de costos de un producto de programación es una de las tareas más difíciles de la ingeniería de software; es difícil hacer estimaciones exactas durante la etapa de planeación de un desarrollo debido a la gran cantidad de factores desconocidos en ese momento, esto se debe en gran medida a que:



- El código fuente no es un artículo uniforme, ni es la esencia del producto deseado.
- El software requiere de la creatividad y cooperación del personal involucrado en el sistema, cuyo comportamiento individual y en grupo es generalmente difícil de predecir.
- Históricamente el software tiene una base pequeña en lo concerniente a la experiencia cuantitativa y es difícil formar parte de esta base desarrollando pequeños experimentos controlados.

Los principales factores que influyen en los costos de un producto de programación son :

- |                              |                             |
|------------------------------|-----------------------------|
| - Capacidad del programador. | - Complejidad del producto. |
| - Tamaño del programa.       | - Tiempo disponible.        |
| - Confiabilidad requerida.   | - Nivel tecnológico.        |

### **Capacidad del programador**

La producción y mantenimiento de productos de programación son tareas laboriosas, por lo que la productividad y calidad son funciones directas de la capacidad y esfuerzo individuales. Existen dos aspectos en la capacidad: la competencia global del individuo y su familiaridad con el área particular de aplicación; hay programadores que se muestran competentes en el procesamiento de datos y suelen no serlo en áreas científicas, y de igual forma, un buen programador científico no es, forzosamente, un buen programador de sistemas. La falta de familiaridad con el área de aplicación puede implicar baja productividad y poca calidad.

### **Complejidad del producto**

Existen tres categorías para los productos de programación: programas de aplicación, en los que se incluyen procesamiento de datos y programas científicos; programas de apoyo, como compiladores, ligadores y sistemas de inventario, y programas de sistemas como sistemas de base de datos, sistemas operativos y sistemas operando en tiempo real.

Barry Boehm utiliza tres niveles de complejidad de un producto y proporciona ecuaciones para predecir el esfuerzo total en meses de programador requerido en su desarrollo, considerando como variable independiente al número de millares de instrucciones de código fuente entregadas con el producto.

## **Tamaño del programa**

Un proyecto grande de programación es obviamente más caro en su desarrollo que uno pequeño, las ecuaciones de Boehm indican que la tasa de crecimiento en cuanto al esfuerzo requerido aumenta con el número de instrucciones de código fuente que tengan un exponente un poco mayor de uno.

## **Tiempo disponible**

El esfuerzo total del proyecto se relaciona con el calendario de trabajo asignado para la terminación de proyectos, varios investigadores han estudiado la cuestión del tiempo óptimo de desarrollo, la mayoría concuerdan que los proyectos de programación requieren más esfuerzos si el tiempo de desarrollo se reduce.

## **Confiabilidad requerida**

Todo producto de programación debe poseer un nivel elemental de confiabilidad; sin embargo, la alta Confiabilidad se logrará teniendo gran cuidado en los aspectos:

De análisis, diseño, desarrollo, pruebas y mantenimiento del producto de programación. Se requiere tanto recursos humanos como equipo para obtener un aumento en la confiabilidad; lo anterior conduce a una reducción en la productividad, medida en términos de líneas de código producido durante un mes.

## **Nivel tecnológico**

El nivel tecnológico utilizado en un proyecto de programación incluye aspectos como selección de lenguaje, ambiente computacional, prácticas de programación y herramientas de programación disponibles. Los lenguajes de programación modernos proveen características mejoradas para la definición y manejo de datos, estructuras de construcción para la definición de flujos de control, mejores facilidades de modularización, manejo eficiente de condiciones y facilidades para la programación concurrente.

El ambiente computacional se refiere al conjunto de características del equipo y los programas disponibles para el desarrollo, uso y mantenimiento del producto. La estabilidad y disponibilidad del ambiente computacional influye notablemente en la productividad y calidad del producto.

Las técnicas modernas de programación comprenden el uso de un análisis sistemático y técnicas de diseño, nomenclatura apropiada, codificación estructurada, técnicas de depuración y estudio de documentos y código fuente.

Las herramientas de programación van desde las herramientas más elementales como ensambladores y depuradores sencillos hasta ambiente totales de programación que incorporan herramientas para la administración y el control del desarrollo del proceso.

### **Técnicas de estimación de costos de software**

Dentro de la mayor parte de las organizaciones la estimación de costo de la programación se basan en las experiencias pasadas. La estimación de costos puede llevarse a cabo en forma jerárquica hacia abajo o en forma jerárquica hacia arriba, llamada BOTTOM-UP. La estimación jerárquica hacia abajo se enfoca a los costos del nivel del sistema, así como los costos del manejo de la configuración, del control de calidad, de la integración del sistema, del entrenamiento y de las publicaciones de documentación. Los costos del personal relacionado se estiman mediante el examen de costo de proyectos anteriores que resulten similares.

En la estimación jerárquica hacia arriba, primero se estima el costo de desarrollo de cada módulo o subsistema, tales costos se integran para tener un costo total.

### **Juicio experto**

La técnica más utilizada para la estimación de costos es el uso de juicio experto que además es una técnica de tipo jerárquica hacia abajo. El juicio experto se basa en la experiencia, en el conocimiento anterior y en el sentido comercial de uno o más individuos dentro de la organización.

La mayor ventaja de juicio experto, que es la experiencia, puede llegar a ser su debilidad; el experto puede confiarse de que el proyecto sea similar al anterior, pero bien puede suceder que haya olvidado algunos factores que ocasionen que el sistema nuevo sea diferente, o quizás que el experto que realiza la estimación no tenga experiencia en ese tipo de proyectos.

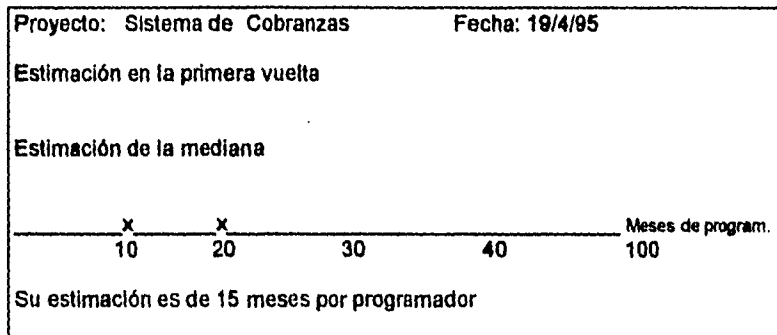
La mayor desventaja de la estimación en grupo es el efecto que la dinámica interpersonal del grupo pueda tener en cada uno de los individuos; los miembros de un grupo pueden ser inocentes con respecto a factores de tipo político, a la presencia de alguna autoridad dentro del grupo, o al dominio de un miembro del grupo con una fuerte personalidad.

### **Estimación del costo por la técnica DELFI**

La técnica DELFI fue desarrollada en la corporación RAND en 1948 con el fin de obtener el consenso de un grupo de expertos sin contar con los efectos negativos de las reuniones de grupo. La Técnica puede adaptarse a la estimación de costo de la siguiente manera:

1. Un coordinador proporciona a cada experto la documentación con la *definición del sistema* y una papeleta para que escriba su estimación.
2. Cada experto estudia la definición y determina su estimación en forma anónima; los expertos pueden consultar con el coordinador, pero no entre ellos.
3. El coordinador prepara y distribuye un resumen de las estimaciones efectuadas, incluyendo cualquier razonamiento extraño efectuado por algunos de los expertos.
4. Los expertos realizan una segunda ronda de estimaciones, otra vez anónimamente, utilizando los resultados de la estimación anterior. En los casos que una estimación difiera mucho de las demás, se podrá solicitar que también en forma anónima el experto justifique su estimación.
5. El proceso se repite tantas veces como se juzgue necesario, impidiendo una discusión grupal durante el mismo.

Un ejemplo de papeleta para la estimación de costos mediante la técnica DELFI se presenta en la **figura 2.2**.



**Figura 2.2. Ejemplo técnica Delphi.**

Razones para su estimación:

- Parece un sistema normal de aplicación.
- Nuestra gente a tenido gran experiencia en proyectos similares.
- No se espera que haya problemas con este proyecto.

Este proyecto lo realizó la empresa Sistemas Automatizados de Información para Seguros Atlas. La tarea fue satisfacer sus necesidades de cobranzas en primas de seguro.

El siguiente enfoque es una variación de la técnica DELFI tradicional que aumenta la comunicación conservando el anonimato:

1. El coordinador proporciona a cada experto la documentación con la *definición del sistema* y una papeleta para que escriba su estimación.
2. Cada experto estudia la definición y el coordinador llama a una reunión del grupo con el fin de que los expertos puedan analizar los aspectos de la estimación con él y entre ellos mismos.
3. Los expertos terminan su estimación en forma anónima.
4. El coordinador prepara un resumen de las estimaciones efectuadas sin influir los razonamientos realizados por algunos de los expertos.
5. El coordinador solicita una reunión del grupo para discutir los puntos donde las estimaciones varían más.
6. Los expertos efectúan una segunda ronda de estimaciones, otra vez en forma anónima. El proceso se repita tantas veces como se juzgue necesario.

Es posible que después de varias rondas de estimaciones no se llegue a un consenso; en ese caso, el coordinador deberá analizar los aspectos relacionados con cada experto para determinar las causas de tales diferencias.

Puede ser que el coordinador tenga que recabar información adicional y presentárselas a los expertos con el fin de resolver las diferencias en los puntos de vista.

### **Técnica COCOMO**

El modelo COCOMO está clasificado como un modelo algorítmico, se considera a su vez un modelo compuesto que incorpora funciones lineales, multiplicativas, analíticas y tabulares para estimar el esfuerzo del software en función de ciertos atributos.

Los modelos algorítmicos están provistos de uno o más algoritmos los cuales producen una estimación del costo del software en función de ciertos atributos, tales como, características del hardware, software del sistema y personal involucrado en el desarrollo del mismo.

Comparados con otros métodos de estimación, los modelos algorítmicos tienen varias ventajas: son objetivos, y no se ven influenciados por factores tales como deseos de trabajar, deseos de terminar el proyecto, etc; son repetitivos, esto es, se obtienen resultados iguales para peticiones iguales en fechas diferentes;

son eficientes y se pueden ajustar de manera objetiva debido a experiencias en proyectos anteriores.

Barry Boehm introduce una jerarquía de modelos de estimación de software relacionado al nombre COCOMO, para modelo de Costo Constructivo. La jerarquía de Boehm lleva la siguiente forma:

*Modelo 1. COCOMO Básico.* Es un modelo estático único-valorado, que valúa el esfuerzo de desarrollo de software como una función de tamaño de programa expresada en líneas estimadas de código.

*Modelo 2. COCOMO Intermedio.* En este modelo se evalúan los esfuerzos de desarrollo de software como una función de tamaño de programa y un conjunto de manejadores de costo que incluyen evaluaciones subjetivas del producto, hardware, personal y atributos del proyecto.

*Modelo 3. COCOMO Avanzado.* Incorpora todas las características de una versión intermedia con una evaluación del impacto del manejador de costos en cada paso, como son análisis, diseño etc., de los procesos de ingeniería de sistemas.

COCOMO puede ser aplicado a tres clases de proyectos de software, éstas son: (1) modo orgánico- relativamente pequeño, proyecto de software simple en la cual pequeños equipos con buena experiencia de aplicación trabajan para poner requerimientos menos rígidos. (2) Modo semi-separado, es un proyecto de software intermedio en tamaño y complejidad, en la cual equipos con niveles de experiencia mezclada deben encontrar una mezcla de requerimientos rígidos y menos rígidos. (3) Modo incrustado, es un proyecto de software que debe ser desarrollado dentro de un conjunto de hardware, software complicado. Las ecuaciones básicas de COCOMO llevan la forma:

$$E = a_b (KLOC) \exp(b_b)$$

$$D = c_b (E) \exp(d_b)$$

Donde **E** es el esfuerzo aplicado en meses-persona, **D** es el tiempo desarrollo en meses, y **KLOC** es el número estimado de líneas entregadas de código para el proyecto, expresadas en miles. Los coeficientes  $a_b$  y  $c_b$  y los exponentes  $b_b$  y  $d_b$  se obtienen de la **Tabla 2.3.**(COCOMO modelo básico).

Proyecto de software	$a_b$	$b_b$	$c_b$	$d_b$
Orgánico	2.4	1.05	2.5	0.38
Semi-separada	3.0	1.12	2.5	0.35
Incrustada	3.6	1.20	2.5	0.32

**Tabla 2.3. COCOMO Básico.**

Cabe hacer la aclaración que en la versión del modelo COCOMO descrito por Boehm una persona mes se define como 152 horas de tiempo de trabajo efectivo. Esta cifra tiene en cuenta el tiempo mensual promedio de días no laborables, de capacitación y de ausencia por enfermedad. El factor toma en cuenta la citación laboral de E.U., en México, las horas tiempo de trabajo efectivo disminuyen. Este valor debe ser tomado en cuenta a la hora de realizar el calculo del costo en pesos.

El modelo básico es extendido para considerar un conjunto de 'atributos de manejadores de costo" que pueden ser agrupados dentro de cuatro categorías:

- |                            |   |
|----------------------------|---|
| 1. Atributos de producto.  | <ul style="list-style-type: none"> <li>◇ Confiabilidad del software requerido.</li> <li>◇ Tamaño de la base de datos de aplicación.</li> <li>◇ Complejidad del producto.</li> </ul>   |
| 2. Atributos del Hardware. | <ul style="list-style-type: none"> <li>◇ Ejecución del tiempo de corrida.</li> <li>◇ Memoria.</li> <li>◇ Volatilidad del ambiente de la máquina virtual.</li> </ul>   |
| 3. Atributos personales.   | <ul style="list-style-type: none"> <li>◇ Requerimientos alrededor del tiempo</li> <li>◇ Capacidad del analista.</li> <li>◇ Capacidad del ingeniero de sistemas.</li> <li>◇ Experiencia en aplicaciones.</li> <li>◇ Experiencia en máquinas virtuales.</li> <li>◇ Experiencia en lenguajes de programación.</li> </ul> |
| 4. Atributos del proyecto. | <ul style="list-style-type: none"> <li>◇ Uso de herramientas de software.</li> <li>◇ Aplicación de métodos de Ingeniería de Software.</li> <li>◇ Horario de desarrollo requerido.</li> </ul>  |

Cada uno de los quince atributos es clasificada en una escala de seis puntos, con rangos de muy bajo a muy alto. Basado en la clasificación, un esfuerzo multiplicador es determinado de las tablas publicadas por Boehm y el producto de todo resultado de esfuerzo multiplicador es un factor de ajuste de esfuerzo llamado **EAF**. Los valores típicos para **EAF** está en el rango de 0.9 a 1.4.

La ecuación del Cocomo Intermedio lleva la forma:

$$E = a_i (\text{KLOC}) \exp(b_i) * \text{EAF}$$

Donde **E** es el esfuerzo aplicado en personas-meses y **KLOC** es el número estimado de líneas entregadas de código para el proyecto. Los coeficientes  $a_i$  y el exponencial  $b_i$  son dados por la **Tabla 2.4.** (Cocomo Intermedio).

Proyecto de Software	$a_i$	$b_i$
Orgánico	3.2	1.05
Semi-separada	3.0	1.12
Incrustado	2.8	1.20

Tabla 2.4. COCOMO Intermedio.

El modelo Cocomo Avanzado incorpora todas las características de la versión intermedia ( ecuaciones y tabla de valores), considerando una evaluación del impacto del manejador de costos en cada paso como análisis, diseño, desarrollo etc., que se toman en cuenta en el proceso de ingeniería de sistemas.

### Aplicación de COCOMO al Sistema de Control y Actualización del Activo Fijo

El sistema de Control y Actualización para el Activo fijo constará del orden de 10,000 líneas de código por lo que la variable **KLOC** toma el valor de 10.

De las tres clases de proyectos de software mencionados, y en función de la complejidad de nuestro sistema, este se ubica dentro del nivel orgánico, cuyas variables y ecuaciones para nuestro sistema son:

De la tabla Cocomo Básico nuestras variables toman los siguientes valores:

$$a_b = 2.4 \quad c_b = 2.5$$

$$b_b = 1.05 \quad d_b = 0.38$$

Desarrollando las ecuaciones:

Descripción del Proceso	Desarrollo de Ecuación	Unidades
El esfuerzo estimado:	$E = a_b (KLOC) \exp(b_b)$	[meses-programador]
Tiempo de desarrollo:	$D = c_b (E) \exp(d_b)$	[meses]
Número de programadores:	$N = E / D$	[programadores]
El esfuerzo estimado:	$E = a_b (KLOC) \exp(b_b)$ $E = 2.4 \times (KLOC) \exp(1.05)$ $E = 2.4 \times 10 \exp(1.05)$ $E = 26.9$	[meses-programador]
Tiempo de desarrollo:	$D = c_b (E) \exp(d_b)$ $D = 2.5 \times E \exp(0.38)$ $D = 2.5 \times 26.9 \exp(0.38)$ $D = 8.7$	[meses]
Número de programadores:	$N = E / D$ $N = 26.9 / 8.7$ $N = 3$	[programadores]
Tomando en cuenta que en este caso trabajarán cinco programadores:	$D = E / N$ $D = 26.9 / 5$ $D = 5.38$	[meses de desarrollo]



Con los resultados que acabamos de obtener, desarrollaremos un breve costeo del sistema, tomando como parámetros, por un lado el sueldo por programador de la Institución y por otro el promedio de sueldo de un programador, basándonos en tres consultorías independientes.

En la Institución Bancaria el sueldo mensual promedio de un programador oscila entre N\$ 8,000.00 y N\$ 9,000.00, en nuestro caso tomaremos un sueldo de N\$ 7,500.00 para ser atractiva la oferta.

En el caso de las consultorías investigadas, estas consideran varios factores al momento de determinar el sueldo (incluyen, sueldos, prestaciones, uso de hardware y software, luz, instalaciones, transportes, etc.), por lo cual los sueldos son más elevados, tomando un promedio, determinamos que una consultoría valúa el sueldo mensual por programador en N\$ 12,000.00.

A partir de estos datos procederemos a realizar una tabla comparativa entre lo que costaría realizar el producto en el área de sistemas y en una consultoría externa.

<b>EXPLICACIÓN</b>	<b>DESARROLLO EN ÁREA DE SISTEMAS</b>	<b>DESARROLLO POR CONSULTORÍA</b>
Tiempo de Desarrollo (meses).	5.38	8.70
Tiempo de Desarrollo (horas). Según Boehm cada mes se trabajan 152 horas efectivas.	817.76	1,322.40
Número de programadores.	5	3
Sueldo mensual por Programador.	N\$ 7,500.00	N\$12,000.00
Sueldo diario (8 horas).	N\$ 250.00	N\$ 400.00
Sueldo por hora trabajada.	N\$ 31.25	N\$ 50.00
Costo por programador.	N\$ 25,555.00	N\$ 66,120.00
<b>Costo Total del Proyecto.</b>	<b>N\$ 127,775.00</b>	<b>N\$ 198,360.00</b>

Como se puede ver claramente en los resultados, la opción propuestas por sistemas resulta ser menos costosa, y el monto de la inversión es un 35% menos que la consultoría.

## 2.4 Recursos disponibles para la realización del sistema

Es fundamental a la hora de realizar o proyectar un sistema el saber los recursos con los cuales se contará para efectuar esta labor, es por ello que en esta parte del capítulo nos enfocaremos al estudio y análisis de los recursos disponibles, partiremos de los dos bloques importantes para todo sistema: hardware y software, a partir de esto, describiremos todos los componentes que existen y puedan ser agrupados en estos bloques.

Debemos dejar bien claro que, como se definió en el inciso 2.2. *información recabada*, una de las premisas fundamentales es que el sistema funcione en una red Novell (ya instalada y probada), con sistema operativo Netware y con un manejador de bases de datos homologado (en este caso FoxPro).

Apegándonos a estos requerimientos, mostraremos de manera breve que estas herramientas son lo suficientemente buenas para el desarrollo que nosotros pretendemos. Por una parte Novell como ya se vio en el *Capítulo 1*, es uno de los mejores sistemas de redes locales, en cuanto al hardware donde se instaló la red, daremos sus características y estructura de la red, para finalmente mostrar un estudio comparativo de FoxPro contra otros manejadores de bases de datos con una estructura similar.

### Hardware

En cuanto a equipos, se cuenta con computadoras personales con capacidad suficiente para el desarrollo y la posterior operación del sistema de aplicación a desarrollar, redes locales de computadoras personales a las que tiene acceso el área usuaria (Contabilidad de Capital), e impresoras asignadas al usuario final. Los equipos con los que se cuenta son los siguientes:

EQUIPO	MODELO	DISCO	PROC.	MEMORIA	USO
SERVER	M400-60	2 Gb	80486	16Mb RAM	Desarrollo/Producción
PC OLIVETTI	M6 400	160 Mb	80486	8Mb RAM	Desarrollo
PC OLIVETTI	M6 400	160 Mb	80486	8Mb RAM	Desarrollo
PC OLIVETTI	M300-30	160 Mb	80486	4Mb RAM	Producción
PC OLIVETTI	M300-30	160 Mb	80486	4Mb RAM	Producción
PC OLIVETTI	MSX 20	80 Mb	80386	2Mb RAM	Producción
PC OLIVETTI	MSX 20	80 Mb	80386	2Mb RAM	Producción

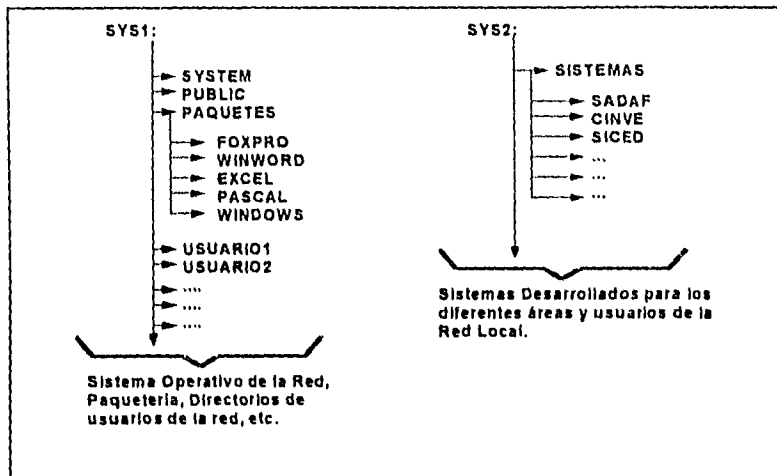
En cuanto a impresoras se tienen:

IMPRESORA	TIPO	RESOLUC.	EN RED	USO
HP LaserJet 4	Laser	600 dpi	Si	Desarrollo/Producción
Epson FX 1050	Matricial	240 dpi	No	Producción
Epson FX 1050	Matricial	240 dpi	No	Producción
Epson FX 1050	Matricial	240 dpi	No	Producción
ATI MT640	Matricial Alta Veloc.	240 dpi	Si	Desarrollo/Producción

Como podemos ver en la tabla de equipos, se menciona un server y hay que aclarar que uno de los requerimientos del sistema es que pueda operar en un ambiente de red local, es por ello que a continuación tocamos algunos puntos relacionados a redes, orientándonos a los componentes y productos que se manejan en la institución.

### Especificaciones de la red local

Servidor de red:                    ◊ Disco duro de 2 Gb particionado en dos volúmenes de igual tamaño con la siguiente estructura de directorios:



- ◆ Estaciones de trabajo:    ◊ Un total de 19 estaciones, de las cuales sólo 5 tendrán acceso al sistema.
- ◆ Topología de la red:     ◊ Tipo árbol con repetidores activos a través de conmutadores.
- ◆ Tipo de cableado:        ◊ Cable telefónico de 8 hilos.
- ◆ Tipo de tarjeta:         ◊ Ethernet con protocolo CSMA/CD.
  - ◊ Velocidad de transmisión de 10 Mb/seg.
  - ◊ En redes de pocos nodos tiene máximo rendimiento.
  - ◊ Trabaja por segmentos, cada segmento es un tramo de cable.
  - ◊ La longitud máxima de cada segmento, depende del fabricante, varía entre 500 y 1000 metros.

- ◆ Tipo de tarjeta:
  - ◇ Forma parte de los 3 estándares de hardware de red fijados por la IEEE.
  - ◇ Es el estándar que permite la conexión más amplia entre equipos.
  - ◇ Sus principales fabricantes son 3Com Corp., Ungermann-Bass, Micron-Interlan y Excelan.
- ◆ Sistema Operativo: ◇ MS-DOS ver 6.0.
- ◆ Sistema Operativo de Red. ◇ Netware 3.11.

## Software

En la actualidad en la institución bancaria se cuenta con paquetes homologados como FoxPro y Turbo Pascal para desarrollo de aplicaciones a nivel computadora personal. También se cuenta con el manejador de archivos Btrieve, que puede usarse desde paquetes como Pascal, BASIC o Cobol. Todos estos paquetes se corren en redes locales con sistema Netware 3.11, del cual a continuación mencionaremos las características más importantes desde nuestro punto de vista.

### Características de Novell Netware

El Advanced Netware es un sistema operativo de red independiente del hardware, por lo cual puede correr en una gran variedad de redes. Ha estado en el mercado desde 1983 y es el sistema operativo más usado. Novell desarrolló originalmente el Netware como sistema operativo para el equipo Novell-S Net, una red que utiliza una topología de estrella y un servidor propietario basado en un procesador Motorola MC 68000.

Debido a que este procesador no tenía ningún sistema operativo, Novell decidió desarrollar el suyo partiendo de cero, y lo optimizó para redes, diseñando de paso todas sus características, alrededor de la funcionalidad de la red.

Debido a que el ROOM BIOS de la IBM PC XT fue diseñado para un sistema operativo (DOS) de un solo usuario, con la deficiencia para ambiente multiusuario y a que Netware es particularmente multiusuario, los programadores de Netware decidieron ignorar el ROOM BIOS y comunicarse directamente con el hardware. La única desventaja de esta forma de operar es la imposibilidad de Netware de utilizar las interfaces (drivers) del DOS, para disco duro. Novell surte estas interfaces para discos compatibles con IBM y muchos fabricantes surten sus propios drivers para Netware.

Netware utiliza cuatro componentes mayores de software:

- El sistema operativo huésped (DOS).
- La interface "SHELL" con DOS.
- El software del servidor de archivos.
- Las utilerías de la red.

El sistema operativo huésped que corre en la estación de trabajo puede ser cualquier versión de MS-DOS, a partir de la 2.0 en adelante.

El 'SHELL' de Netware provee también en las estaciones de trabajo una liga de comunicación entre las estaciones y/o aplicaciones y el software del servidor de archivos. Este SHELL ofrece además la compatibilidad de Netware con DOS y utiliza todas las interfaces estándar establecidas por MS-DOS y NETBIOS. Si la aplicación requiere un servicio de NETBIOS, se puede comunicar directamente con el emulador de éste, mismo que pasará la información al IPX. Por otro lado, si la llamada es local, el SHELL transfiere al DOS para que ejecute, de lo contrario la llamada es pasada a alguna de las implementaciones de MS-NET. El software de servidor de archivos de Netware no se corre como una aplicación de DOS en la máquina servidora de la red.

El software de servidor de archivos de Netware está diseñado específicamente para redes y es multiusuario, lo que significa que una tarea no tiene que esperar hasta terminar para iniciar otra. A las diferentes tareas se les asignan distintos niveles de prioridad, a efecto de realizar rápidamente lo de mayor importancia. En el caso de la operación del servidor de archivos NO DEDICADO, el DOS corre en el server, como una tarea del sistema operativo de red.

Netware también provee un 100% de compatibilidad con NETBIOS mediante el emulador que posee. Esto garantiza compatibilidad con los adaptadores de red, tales como Token-Ring y Pc Network. El emulador de NETBIOS brinda operación en modo protegido. Netware adicionalmente viene con 14 drivers para diferentes adaptadores de red, estos drivers permiten hasta 35 diferentes configuraciones de hardware.

**Interconexión de redes:** Netware provee capacidades extensivas para la formación de Inter-redes. El punto interno de Netware que viene incluido le permite al server soportar simultáneamente hasta 4 diferentes topologías, sin tener que dejarlo como dedicado, es decir, que éste podrá procesar otros trabajos simultáneamente.

**Seguridad:** La seguridad de Netware se basa en el manejo de usuarios. EL supervisor establece derechos a un usuario autorizado y le asigna recursos

específicos, los privilegios de acceso a la área de directorios son asignados de acuerdo a ocho diferentes derechos.

La seguridad a nivel de archivos también puede definirse con atributos, ya que los archivos pueden marcarse como compartidos, no compartidos, de lectura solamente o de Lectura/Escritura. Usando este tipo de seguridad, una persona hace un LOGIN a la red mediante un nombre de usuario.

**Operación en modo protegido:** Se puede hacer uso de toda la capacidad del microprocesador Intel del 80286 o posteriores cuando se utiliza una IBM AT o compatible, permitiéndole al servidor de la red, utilizar hasta 16 MB de memoria y hasta 2 GB de almacenamiento en disco.

**Sistema de archivos:** Esta diseñado específicamente para la administración de redes y no tiene las limitaciones de los archivos del DOS, aunque es completamente compatible con él. Netware tiene su propia estructura de manejo de archivos, que se conserva en memoria y en disco, e implementa cuatro características adicionales, que hacen que funcione rápidamente en el manejo de archivos.

- a) Directory Caching. Técnica que consiste en tener en RAM prácticamente todos los directorios del disco duro.
- b) Directory Hashing. Esta técnica emplea un algoritmo de búsqueda que además de llevar el directorio a memoria principal, lo ordena en función de un algoritmo "hash", y de acuerdo con esto, accesa un archivo.
- c) File Caching. No sólo los directorios se mantienen en memoria, sino también los bloques de disco más utilizados.
- d) Elevator Seeking. Los requerimientos de lectura o escritura demandados al servidor de archivos, se atienden al paso de la cabeza de Lectura/Escritura en el orden en que la información esta físicamente en el disco, y no en el orden en que ésta llegó.

**Integridad de datos:** El sistema de archivos de Netware utiliza varias medidas preventivas para asegurar la disponibilidad e integridad de los datos, a saber:

- a) Verificación de lectura después de escritura.
- b) Duplicidad de directorios.
- c) Tablas duplicadas de alojamiento de archivos (FAT).

**SFT Netware:** Novell ofrece una actualización para el Advance Netware que brinda protección adicional a los datos.

Esta versión mejorada del sistema operativo se conoce como System Fault Tolerant Netware (SFT) (Sistema Tolerante a Fallas). Y tiene tres niveles de implementación:

- 1) Permite detectar blocks dañados de disco en la operación normal del sistema.
- 2) Permite mantener discos espejo, es decir, se tiene en todo momento discos de respaldo.
- 3) Se tienen dos servidores de archivos con respaldo uno del otro, conectados entre sí con un bus de transferencia de alta velocidad.

Lo descrito hasta el momento son las características más importantes de Novel Netware, sin ser todas, son las más representativas del sistema operativo.

En cuanto a paquetes de software para desarrollo de sistemas se cuenta con las siguientes herramientas:

PAQUETE	MANEJADOR DE ARCHIVOS QUE UTILIZA	VERSIÓN PARA RED	EXPERIENCIA EN SU USO	COMPLEJIDAD EN PROGRAMACIÓN
FoxPro/LAN v2.5	Propio	Si	Muy Buena	Regular
Turbo Pascal v7.0	Btrieve	Si	Buena	Alta
QuickBasic	Btrieve	Si	Buena	Alta

Como se puede observar, de los paquetes mencionados, sin lugar a dudas, el que más se orienta hacia el desarrollo que realizaremos es el FoxPro, en los siguientes párrafos abundaremos un poco en las ventajas y características principales del paquete, sin dejar a un lado el requerimiento establecido de usar software homologado por la Institución.

## 2.5 Justificación de hardware y software

Una de las principales premisas, si no es que la más importante, que se mencionó al momento de realizar las entrevistas con el personal de la Institución, fue que en el desarrollo y aplicación del sistema, deberíamos de usar las instalaciones, el equipo y software con que cuenta la misma, por tanto, tenemos que apegarnos y trabajar con lo que se tiene.

En cuanto a hardware (como se puede ver en los párrafos donde se detallaron las características de éste) podemos decir que es realmente el ideal para el sistema; tanto los equipos como la red local son los adecuados y satisfacen plenamente los requerimientos que el sistema necesitará.

Analizando el software podemos ver claramente que el mejor paquete que se adapta para el desarrollo de nuestra aplicación es el FoxPro; sin embargo, también podríamos usar Turbo Pascal, pero necesitaríamos recurrir a un manejador de archivos externo (Btrieve).

Además sería necesario un paquete externo para la creación y manejo de pantallas de captura (Screen Sculptur), lo cual resultaría mucho más difícil en cuanto a programación y generaría un retraso en cuanto a tiempo de desarrollo pues se tendría que aprender a utilizar estos paquetes externos.

Consideramos que para desarrollar nuestro sistema necesitamos un manejador de bases de datos intermedio, como es FoxPro, debido a que ofrece una gran facilidad en la definición y manejo de menús, ventanas, creación de pantallas de captura, y sobre todo cuenta con un manejador de archivos propio muy eficiente. También el mantenimiento se facilita al usar este paquete, debido a la flexibilidad en la definición y creación de archivos y a la compatibilidad con otros paquetes para exportar e importar datos. Sin embargo, podría pensarse que existen manejadores del mismo nivel que pudieran ser mejores que FoxPro, es por ello que decidimos investigar que tan bueno es el paquete.

En esta investigación se encontró un análisis comparativo que se realizó en 1994 en Estados Unidos, donde se realizaron pruebas a 10 manejadores de bases de datos intermedios con arquitectura archivo/servidor, incluidos en estos el FoxPro para DOS y la versión para Windows, a continuación detallamos estas pruebas.

### **Pruebas de eficiencia de manejadores de bases de datos**

El objetivo de la evaluación fue presentar las herramientas de bases de datos que mejor se adaptaran para crear aplicaciones específicas en un entorno corporativo. Estas bases de datos, para el uso profesional, tenían que ofrecer un entorno de programación que permitiera que el usuario creara aplicaciones con una funcionalidad modificada. Los productos también debían tener herramientas interactivas para usuarios finales.

Como la operación en redes se ha convertido en un estándar para el entorno corporativo, se requirió que todos los productos fueran una base de datos de multiusuarios verdadera, con reservación de archivos y controles de simultaneidad. Los productos evaluados son:

- Dvanced Revelation.
- KnowledgeMan.
- Microsoft FoxPro for DOS.
- Paradox for DOS.
- Rbase.
- DataEase
- Microsoft Access.
- Microsoft FoxPro for Windows.
- Paradox for Windows.
- Superbase.



Para checar el rendimiento se desarrolló un grupo de pruebas para evaluar los límites de cada producto sobre un amplio rango de operaciones y medir rendimiento de los productos en un ambiente multiusuario sumamente exigente. Se optó por usar un subconjunto del AS3AP (ANSI SQL escalable estándar y portátil) pruebas de ejecución para los sistemas de bases de datos por relación, desarrollados por Carolyn Turbyfill, Dina Bitton y Cyril Orji de la Universidad de Cornell.

La plataforma de prueba usa una Compaq Deskpro 486/33Mhz con 8 Mb de memoria y disco duro de 310 Mb. El servidor es una Compaq Systempro 486/33 Mhz con 32 Mb de RAM y un disco duro de 1.6 GB de alto rendimiento con controlador IDA.

Para aumentar la carga en las pruebas de red al tiempo que se evitan los cuellos de botella en el hardware, se probó hasta 36 estaciones de trabajo y distribuyó simétricamente el tránsito de la red usando dos adaptadores EISA Token Ring de Madge de 32 bits en el servidor de archivos, las estaciones de trabajo son Compaq Deskpro 386N basadas en 386SX/16 Mhz con 5 Mb de RAM, realizándose todas las pruebas usando Netware 3.11 (todas las características son muy similares al equipo que nosotros usaremos en el desarrollo del sistema).

### Prueba de carga e índice

La prueba de carga e índice mide cuán rápidamente cada paquete de bases de datos puede importar y poner índices en una base de datos de cuatro tablas con 100,000 filas por tabla. Dos tablas contienen dos índices cada una, una contiene cuatro y la restante contiene cinco, los resultados de esta prueba se pueden ver en la gráfica de la **figura 2.5**.

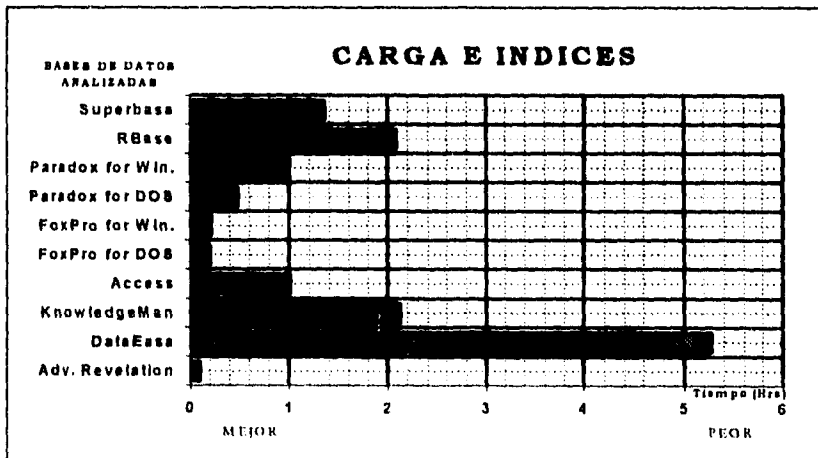


Figura 2.5. Resultados de la prueba de carga e índice.

Como podemos ver, tanto FoxPro para DOS como para Windows sobresalieron en la prueba de carga e índice, completando ambos pasos en 21 minutos, menos de la mitad de tiempo del más cercano producto Paradox para DOS.

### Prueba de actualización

Para la prueba de actualización, se realizaron actualizaciones, inserciones y eliminaciones de grupo. Primero se guardó un subconjunto de 1,000 filas de la tabla en una tabla provisional. Después, se actualizó el rango correspondiente en medio de la tabla de prueba y se cambió el valor de la clave primaria, que fuerza a la bases de datos a actualizar su índice. Los resultados de esta prueba se muestran en la **figura 2.6**.

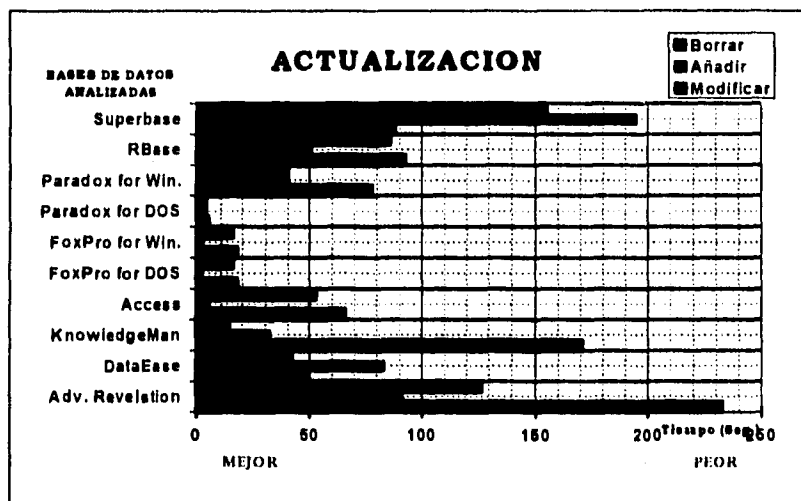


Figura 2.6. Resultados de la prueba de actualización.

Paradox para DOS probó ser muy eficiente al realizar las tres operaciones, completando cada una en menos de 5 segundos. Las dos versiones de FoxPro también probaron ser relativamente rápidas al manejar la tarea Modify.

### Prueba de selección

La prueba de selección muestra cuán rápidamente cada paquete puede realizar indagaciones en tablas únicas donde se regresa con el 10 %. Se realizaron dos tipos de indagaciones. La selección numérica usa la clave primaria de la tabla y un rango secuencial. La indagación alfanumérica desempeña una indagación de comparación exacta de un campo de texto. La respuesta de los paquetes a esta prueba se muestra en la **figura 2.7**.

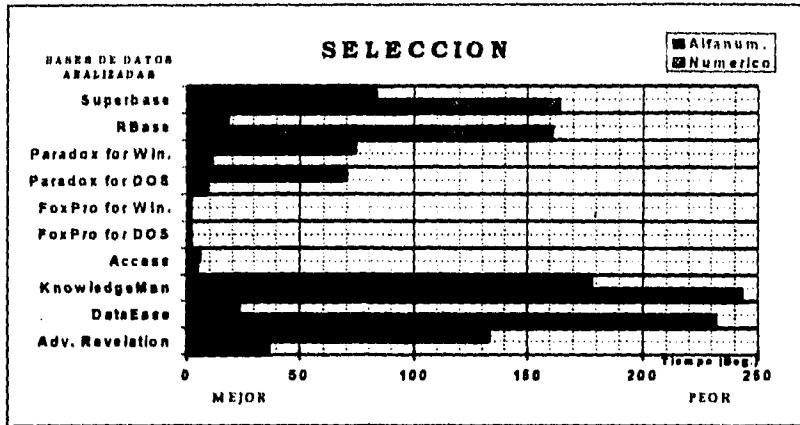


Figura 2.7. Resultados de la prueba de selección.

Tres de los productos, Access de Microsoft y las dos versiones de FoxPro, retornaron una lista de punteros a los datos seleccionados en la memoria y no el valor real de las columnas especificadas. Aunque estos productos ejecutan la indagación casi instantáneamente, deben gastar tiempo adicional para recuperar la información real.

### Prueba de enlace

Para la prueba de enlace se ejecutaron varios de los enlaces especificados en la batería de AS3AP. Aunque los enlaces que retornaron conjuntos pequeños de resultados en su mayoría se procesaron rápidamente, el enlace de dos tablas de una a muchos produjo un amplio rango de rendimiento, dependiendo del plan de optimización que se usó. El resultado de los enlaces se puede observar en la figura 2.7.

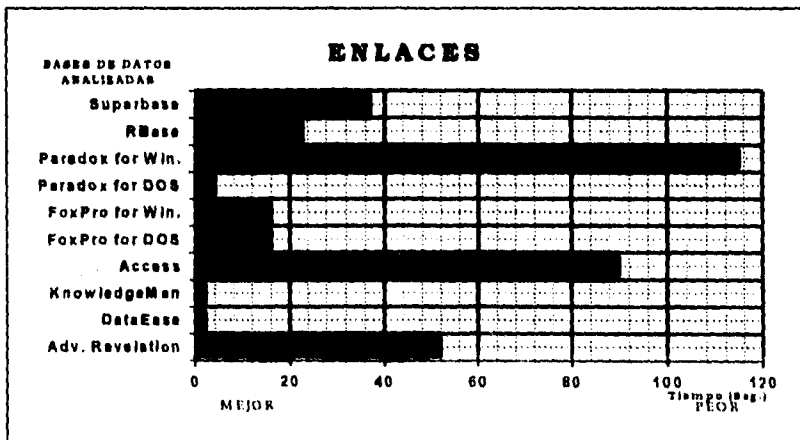


Figura 2.7. Resultados de la prueba de enlaces

DataEase, KnowledgeMan y Paradox para DOS optimizaron excepcionalmente bien la indagación usada en este enlace. En contraste Paradox para Windows y Access produjeron los tiempos más lentos.

### Sumario de características

A continuación presentaremos un sumario de las características más importantes de los productos que sobresalieron en las cuatro pruebas, los productos están listados en orden alfabético, para indicar que sí o no se utilizan los siguientes símbolos (■ - Si □ - No).

Concepto	DataEase	Microsoft FoxPro for DOS	Microsoft FoxPro for Windows	Paradox for DOS	Paradox for Window	Superbase
Precio de Lista.	US\$795	US\$495	US\$495	US\$795	US\$795	US\$795
Entorno Operativo	DOS	DOS	Windows	DOS	Windows	Windows
Versiones Disponibles	Windows OS/2	Macintosh Unix FoxBase+	Macintosh Unix FoxBase+	Windows OS/2	DOS OS/2	Ninguna
Apoyo para memoria Expandida	■	■	□	■	■	■
Apoyo para memoria Extendida	■	■	■	■	■	■
Mínimo requerido	3Mb	No mínimo	3Mb	0.5Mb	3.5MB	1.3Mb
<b>LIMITES DE MANIPULACIÓN DE DATOS</b>						
Máximo de Campos	255 Caract.	256 Caract.	256 Caract.	255 Caract.	255 Caract.	4K
Máximo de registros	4K	64K	64K	4K	4K	Limitado
Número máximo de:						
Campos x registro	255	255	255	255	255	Limitado
Campos índice x reg.	255	255	255	255	255	999
Registros por tablas	16 millones	1 millón	1 millón	255Mb	255Mb	16 millones
Tablas x Base Datos	2000	Ilimitado	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Índice x Base Datos	508,000	Ilimitado	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Archivos abiertos	60	255	255	Ilimitado	Ilimitado	Ilimitado
Vistas x Base Datos	2,000	Ilimitado	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Tablas x vistas	1	255	255	9	9	Ilimitado
Tablas enlazadas	255	224	224	24	24	Ilimitado
<b>TIPOS DE DATOS Y LIMITES</b>						
Matrices:						
Con dimensión	□	■	■	■	■	■
Longitud variable	□	■	■	■	■	■

Concepto	DataEase	Microsoft FoxPro for DOS	Microsoft FoxPro for Windows	Paradox for DOS	Paradox for Window	Superbase
Carácter fijo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Carácter variable	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Enteras	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Decimales fijos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Dec. Punto Flotante	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fecha	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Hora	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Booleana	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Virtual (calculada)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>INDICES</b>						
Indices automáticos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Puede activ. y Desact.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Indice campo compto.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Puede dar ind.a claves	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Puede def. list.de excl.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Apoya únicos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Apoya índices y métodos de acceso:						
B-tree	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
B-tree en racimos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Con pases (hashed)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Tomando como base todos los análisis anteriores podemos concluir que el manejador de base de datos FoxPro es lo suficientemente bueno para efectuar el desarrollo de nuestro Sistema de Control y Actualización de Activos Fijos, y además contamos con la plataforma de hardware adecuada para estos fines..

**ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**

## CAPÍTULO 3 DISEÑO DEL SISTEMA

El propósito de este capítulo es presentar una propuesta de diseño, la cual plantea una solución que pretende resolver la problemática existente en la Institución Bancaria, además, generaremos la información que representa los aspectos más importantes de la documentación requerida para el desarrollo del sistema de control y actualización de activos fijos.

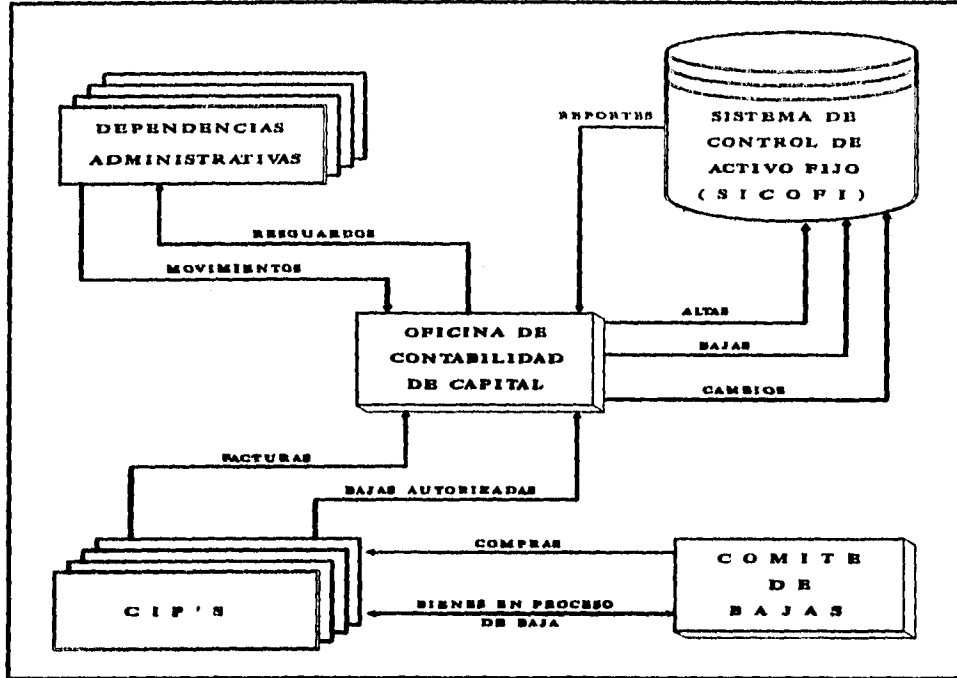
Iniciamos el capítulo definiendo el diseño conceptual del sistema, basándonos íntegramente en el funcionamiento al cual se pretende llegar, por lo cual incluimos una definición de las principales áreas y dependencias que intervienen en el proceso. A partir de esto, identificamos y desarrollamos todos los módulos que deben ser incluidos en el sistema.

Posteriormente definimos la estructura de directorios, bases de datos y campos, especificando para estos su tipo y longitud. Después se presenta el diagrama entida-relación, con la finalidad de llegar a las referencias cruzadas, para identificar archivos programas y campos. Para concluir el capítulo se presenta el diccionario de datos del sistema.

La relevancia de este capítulo, radica en el hecho de que este será la base para el desarrollo del mismo, basta mencionar que un buen diseño facilita y agiliza el desarrollo de cualquier sistema.

### 3.1 Diseño conceptual del sistema

Como se estableció en el Capítulo 1, en el diseño de un sistema deben establecerse los subsistemas que integrarán el sistema de programación final. Partiendo de esta premisa y basándonos en la definición del problema realizado en el capítulo anterior, podemos definir el diseño conceptual del sistema como se muestra en el diagrama de la *figura 3.1*.



**Figura 3.1. Diseño conceptual del sistema**

En el diagrama podemos identificar claramente cuatro entidades funcionando alrededor del sistema de control de activos fijos: las Dependencias Administrativas, la Oficina de Contabilidad de Capital, los Comités de Información Presupuestal (CIP'S) y el Comité de Bajas. Cada uno de estos bloques cumple un papel de importancia para el sistema, a continuación describimos su funcionamiento o participación.

### **Dependencias administrativas**

Son agrupaciones predeterminadas dentro del Banco, éstas pueden catalogarse como gerencias, áreas, secciones, departamentos, etc. Por mencionarlo, el banco cuenta con 9 áreas, 15 gerencias y 450 departamentos.

Esto nos da una idea del volumen de dependencias que existen, las cuales son la base del sistema, pues, es aquí donde se genera la necesidad de catalogar e identificar donde se encuentran ubicados cada uno de los activos de la empresa. Además, estas entidades están continuamente generando nuevas necesidades de equipo y mobiliario.

## **Centros de información presupuestal (CIP'S)**

Los centros de información presupuestal son las entidades encargadas de valorar las necesidades reales de todas las dependencias administrativas, a fin de realizar la adquisición de activos de acuerdo con las prioridades del banco y apegándose a un presupuesto.

Es decir, aquí se determinan las altas, bajas y cambios de todos los activos fijos de la empresa. Cabe señalar que en el banco existen ocho CIP'S, los cuales se agrupan en la siguiente forma:

1. CIP de Inmuebles. Controla el flujo de bienes inmuebles.
2. CIP de Informática. Se encarga de todo lo relacionado con equipo de cómputo, software, hardware y redes.
3. CIP de Fábrica. Controla todo lo relacionado con maquinaria, desde refacciones hasta máquinas completas.
4. CIP de Seguridad. Tiene a su cargo el control de armamento, equipo de seguridad (extintores), hasta equipos de intercomunicación.
5. CIP de Autotransportes. Controla todo el parque vehicular de la institución, incluyendo refacciones.
6. CIP de Administración. Controla todo lo que es papelería, enseres necesarios en los comedores y víveres para los mismos.
7. CIP de Administración Inmobiliaria. Maneja todo lo relacionado con el mantenimiento a instalaciones, mejoras y nuevas construcciones en el banco.
8. CIP de Caja. Se encarga de controlar toda la maquinaria necesaria en el manejo de dinero (billetes y monedas), cajeros, etc.

Como podemos ver, desde estas entidades existe una agrupación muy enfocada a lo que es el manejo y división de los activos fijos.

## **Comité de bajas**

Este comité se encarga de autorizar la baja de algún activo de acuerdo con estudios realizados que verifican la utilidad o inutilidad de éstos. Además, se encargan de determinar el precio de venta de los artículos que son dados de baja y que posteriormente se venderán a otras empresas en una licitación pública. También observa el cumplimiento de contratos de compra-venta que realiza el banco.

## **Oficina de contabilidad de capital**

Es aquí a donde llega toda la información de altas, bajas y cambios realizados en los activos de la institución, podemos decir de antemano que esta área controlará el sistema de activos fijos, observando el buen funcionamiento de éste, y verificando el cumplimiento en los procesos de registro y eliminación de registros.



Esta dependencia controla el sistema porque en ella se determina la actualización de los activos y se registra y controla el monto en dinero que posee el banco en este rubro.

Como podemos ver en el diagrama y en la definición de cada entidad, todas ellas son parte importante en el esquema conceptual del sistema que desarrollaremos.

### 3.2 Diseño modular del sistema (menú principal)

De acuerdo a las investigaciones realizadas y partiendo del diseño conceptual del sistema, hemos podido observar que éste puede subdividirse en siete módulos principales:

- |                         |                             |
|-------------------------|-----------------------------|
| 1. <b>Altas</b>         | 5. <b>Reportes</b>          |
| 2. <b>Bajas</b>         | 6. <b>Respaldos</b>         |
| 3. <b>Cambios</b>       | 7. <b>Cifras de control</b> |
| 4. <b>Mantenimiento</b> |                             |

Cada uno de estos bloques estará relacionado a un archivo maestro que se encargara de activarlos y controlarlos, para ejemplificar este proceso definimos el diagrama de la **figura 3.2**. Como se puede ver todos los módulos del sistema están relacionados con el archivo maestro quien coordina la actividad de cada uno de ellos.

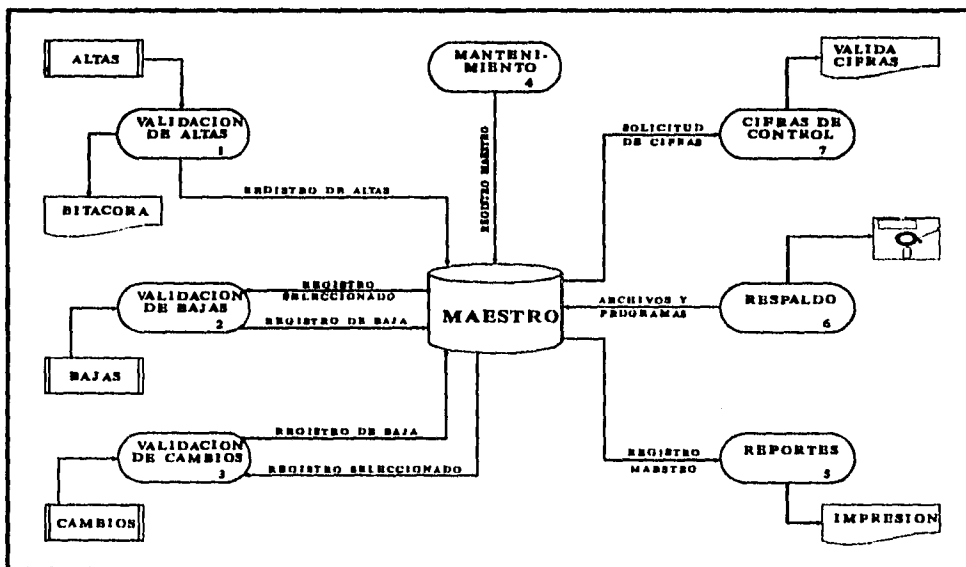


Figura 3.2. Diseño conceptual del menú principal

A continuación damos una pequeña introducción de las funciones que realiza cada módulo, para que más adelante en el siguiente punto se amplíe esta información viendo a detalle cada uno de estos bloques.

### **Altas**

Este proceso permitirá registrar uno o un grupo de artículos en el sistema de activos fijos, realiza la validación de varios datos introducidos.

### **Bajas**

Este proceso permitirá eliminar uno o un grupo de artículos del sistema de activos fijos, realiza la validación de varios datos antes de borrarlos de los archivos.

### **Cambios**

En este módulo podremos realizar la modificación de ciertos datos de uno o varios artículos en el sistema, cuenta también con una validación de ciertos parámetros.

### **Mantenimiento**

En esta opción, nosotros podremos realizar el mantenimiento de todos nuestros catálogos y también actualizaremos las fechas de control del sistema.

### **Reportes**

Con este módulo tendremos la oportunidad de generar resultados de todos nuestros procesos en papel, identificando diferentes alternativas y dejando una opción para que el usuario genere sus propios reportes con la información que se guarda en los archivos.

### **Cifras de control**

Esta opción nos permitirá identificar el número de artículos y su importe adicionado o eliminado en cada sesión de trabajo.

### **Respaldos**

Este bloque del sistema es muy importante, pues nos permite resguardar toda la información en archivos de respaldo, y en caso de cualquier problema, nosotros contaremos con archivos de respaldo que contienen todas las operaciones realizadas.

Es claro que con todas estas opciones estamos abarcando la totalidad de necesidades que se presentan para manejar el sistema de activos fijos. En el siguiente inciso estudiaremos más a fondo cada uno de estos bloques, identificando las validaciones y los archivos que son afectados en cada uno de estos procesos.

### 3.3 Diagrama de módulos del sistema

En esta parte del documento se abordan todos los módulos descritos en el menú principal, tratando de identificar cada uno de ellos como entidades independientes, pero que, al mismo tiempo, están íntimamente relacionadas entre sí.

Uno de los aspectos más relevantes en esta parte del diseño es la determinación de los puntos de control y validación que deben existir en el diseño, esto nos permite asegurar la consistencia e integridad de la información, evitando la redundancia o duplicidad de los datos.

Desde nuestro punto de vista trataremos que una de las principales premisas del sistema sea la validación de la información, es por ello que en el sistema trataremos en particular de validar los siguientes datos:

- **Fecha:** La cual estará conformada por el mes y el año, ésta deberá ser siempre menor o igual al mes en curso.
- **Grupo:** Existirá un catálogo de grupos conformado por una cuenta de mayor y tres niveles de dependencia, donde asociaremos cada uno de los activos. El grupo debe existir y además, debe ser un grupo sin niveles inferiores, es decir, debe ser un grupo afectable, de lo contrario el sistema no actualizará.
- **Número de serie:** Es el número de identificación de un artículo, el cual debe ser único. Además, como sabemos, no todos los activos tienen este número, por ejemplo, las sillas no lo tienen, algunas veces se asigna uno de acuerdo con algún criterio, pero en sí el grupo no necesita de él, en este caso el sistema debe checar en el catálogo de grupos si el artículo requiere número de serie o no. Esta última operación nos indica que el catálogo de grupos deberá contener un registro que indique si el grupo lleva o no número de serie.
- **Pertenece a:** Hay artículos que por su naturaleza son parte de otro activo, por ejemplo, las tarjetas de red son activos, que a su vez pertenecen a otro activo más general, la computadora. En este caso el activo debe verificar si pertenece a otro, entonces deberá de identificarse el número de serie del activo al cual pertenece. Nuevamente, esta operación implica que debe existir un registro en el catálogo de grupos que indique la pertenencia.
- **Referencia:** Se trata de una clave que identifica al departamento, sección, o área en la cual se localizará el activo. Para esto debemos tener un archivo de referencias, con las claves y nombres de cada área de la institución, y en cada operación se deberá verificar que la referencia exista.

- **Tipo de receptor:** Este dato nos dará información del tipo de asignación que se hace al activo, es decir, indicará si se recibe como definitivo, distribuidor, préstamo, rentado, etc. Nosotros tendremos un archivo con la clave y tipo de receptor, y el sistema valida la recepción de un activo por medio de una clave.
- **Importe:** Valida que el importe o costo del activo sea mayor o igual a cero.

Después de analizar las validaciones que se efectuarán en todos los módulos, a continuación procedemos a definir los diagramas esquemáticos de cada bloque, identificando y explicando los puntos más relevantes de cada uno de ellos.

### 3.3.1 Módulo de altas

Este bloque tiene como objetivo fundamental el registro de cualquier activo fijo que ingrese a una determinada área del banco, la información podrá ser captada por registro o por factura. Cuando realicemos una alta por registro, el sistema actualiza la información de cada activo en forma individual; al realizar un registro por factura, tomamos en cuenta que en este documento generalmente se especifica un "n" números de artículos iguales (100 sillars), el sistema captará la cantidad, descripción y valor general de la factura, para posteriormente asignar un valor a cada activo. En cualquiera de estos casos, el sistema realizará todas las operaciones que se ilustran en la **figura 3.3**.

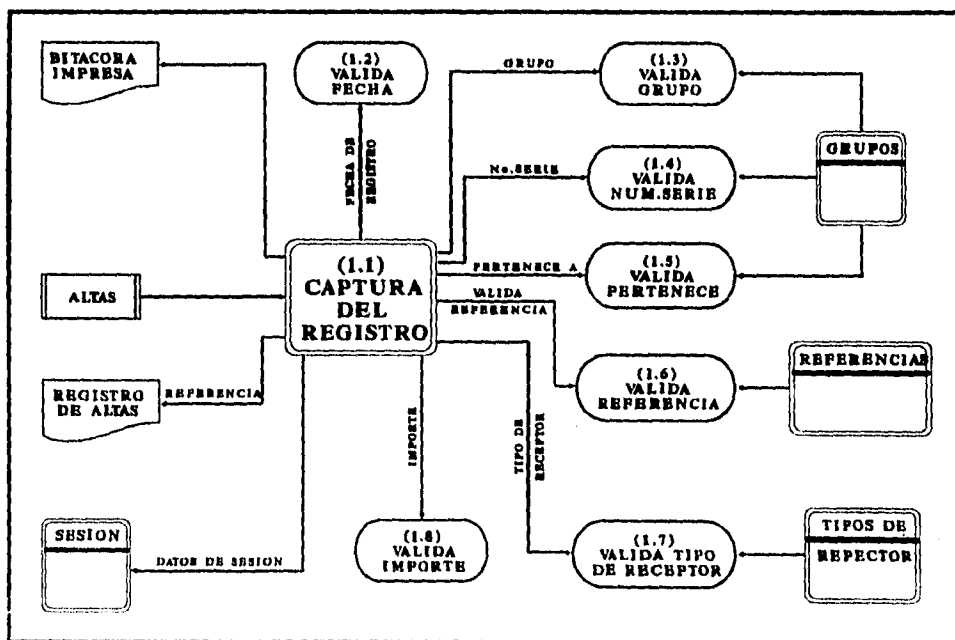


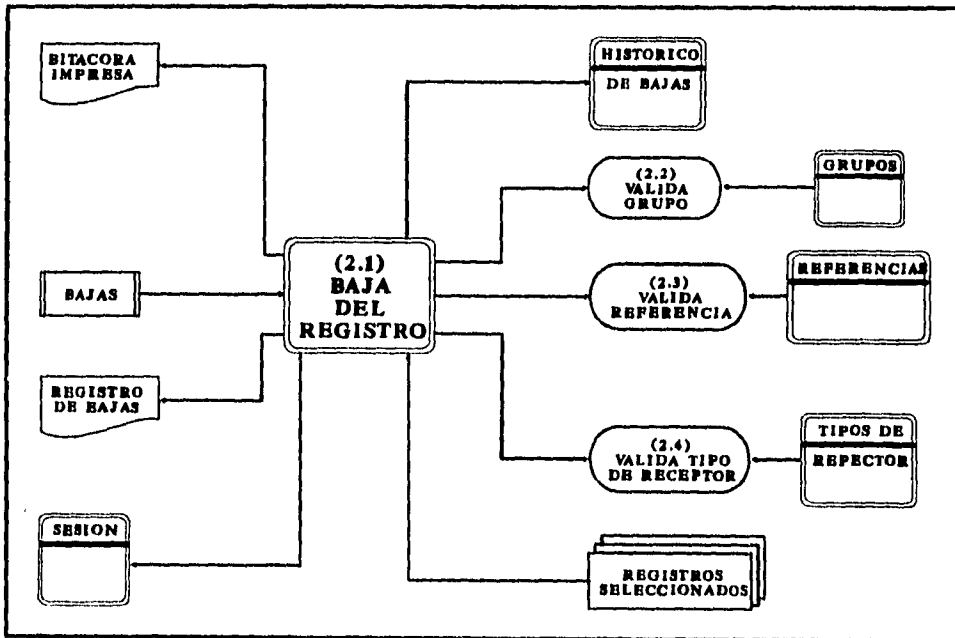
Figura 3.3. Diseño conceptual del módulo de altas

Como se puede observar, este módulo realiza algunas de las validaciones que describimos con anterioridad, después de validar la información, ésta es almacenada en un registro de altas, por medio del cual nosotros tendremos un registro cronológico, y a través de éste podremos capturar datos de períodos futuros, sin haber afectado un período determinado.

Adicionalmente contaremos con una bitácora impresa donde se registrarán todas las altas que se efectúen en una sesión de trabajo, ya sea registros individuales o por factura. Esta forma es un respaldo adicional de la información que se ha procesado en el sistema.

### 3.3.2 Módulo de bajas

En esta parte del sistema el usuario podrá eliminar cualquier registro o grupos de ellos pertenecientes a un activo fijo, cabe aclarar que este módulo no borra físicamente la información, sólo la transferirá de un archivo a otro. Las bajas pueden ser de un solo artículo, o de un grupo de artículos que cumplan con una determinada selección, por referencia, pertenencia, tipo, número de serie, mes o año. El sistema realizará la validación de estos campos y entonces efectuará las operaciones que se muestran en la **figura 3.4**.



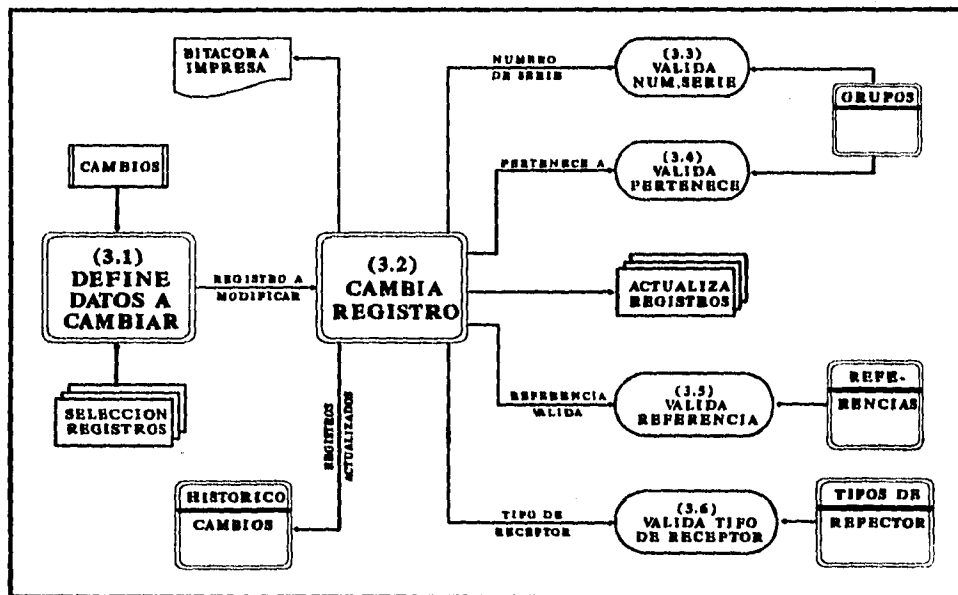
**Figura 3.4. Diseño conceptual módulo de bajas**

Como ya mencionamos, las bajas no implican una eliminación de la información, esto se puede observar claramente en el diagrama, donde notamos que existe un archivo histórico de bajas, en el cual se deposita la información.

Esta es una medida para evitar que en caso de algún error o una reactivación de activos fijos, los usuarios tengan que estar capturando nuevamente los datos. Además contaremos como en el caso de las altas con un registro de bajas y una bitácora impresa donde se registrarán todas las bajas que se efectúen en una sesión de trabajo, ya sea registros individuales o por grupo. Como ya mencionamos esto es una forma de respaldo.

### 3.3.3 Módulos de cambios

El proceso de cambios será creado para modificar datos de un o un grupo de registros, tendrá la misma filosofía que el de bajas, es decir, podemos modificar un grupo de activos que cumplan con ciertas características específicas en sus campos. Esto se ilustra claramente el diagrama de la **figura 3.5**.



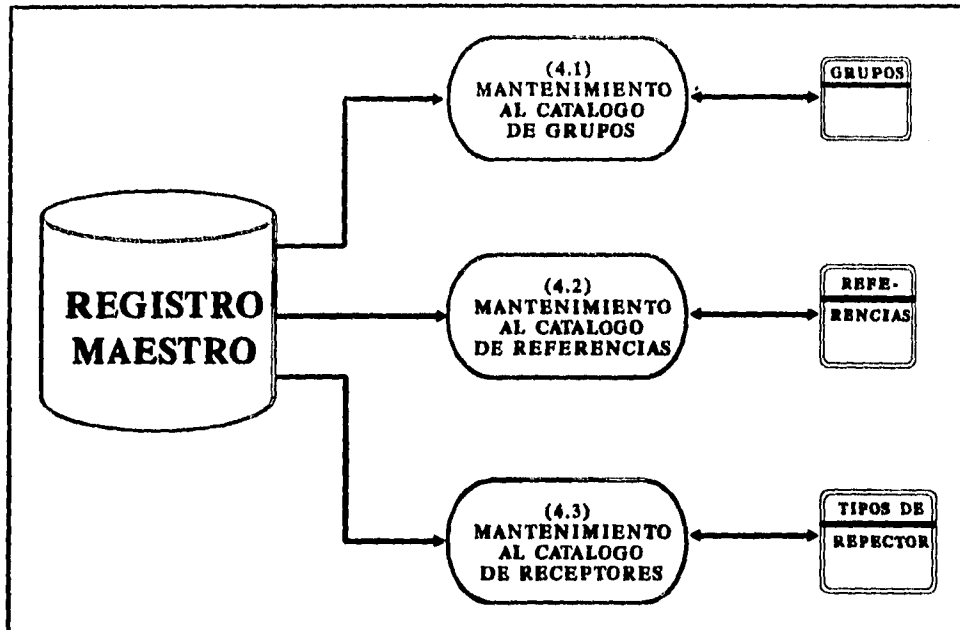
**Figura 3.5. Diseño conceptual del módulo de cambios**

Podemos observar todas las validaciones que efectúa el módulo, adicionalmente tenemos un archivo histórico de cambios que nos muestra todos los movimientos efectuados y la ya conocida bitácora de impresión que registra en papel todas las actividades realizadas.

### 3.3.4 Módulo de mantenimiento

Las actividades de mantenimiento han sido pensadas para que el usuario pueda actualizar los tres archivos más importantes con que cuenta el sistema: catálogo de grupos, catálogo de referencias y tipos de receptores.

También en este bloque se encontrará una opción para actualizar las fechas de control. Todas estas operaciones se pueden analizar en el diagrama de la **figura 3.6.** que se muestra a continuación.



**Figura 3.6. Diseño conceptual del módulo de mantenimiento**

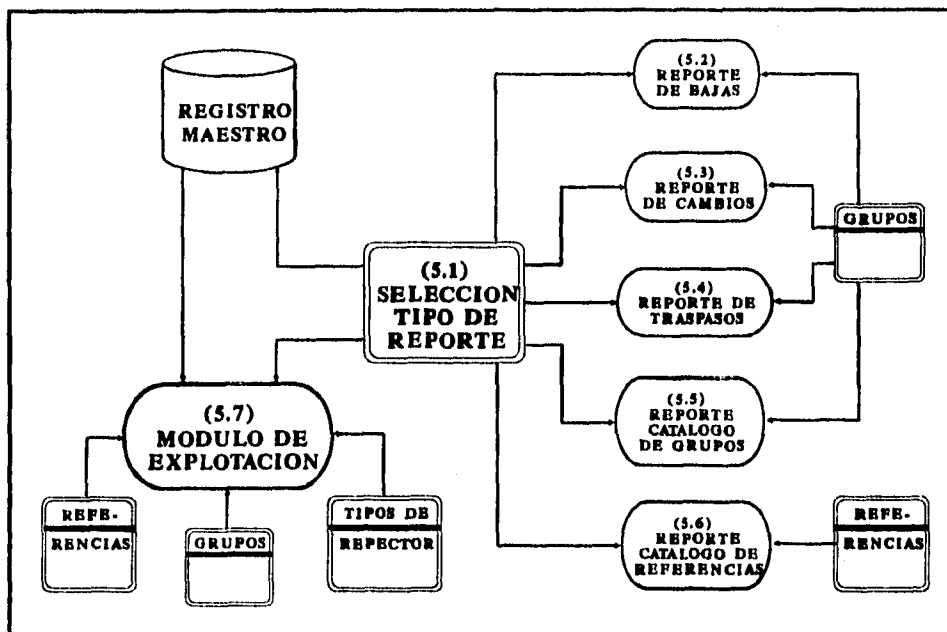
Para cada uno de los tres catálogos mencionados, el sistema deberá contar con las opciones básicas de altas, bajas y modificaciones. Estas funciones deberán ser aplicables a los archivos, validando en cada uno de ellos que dichas operaciones no provoquen redundancia o inconsistencia de la información asociada a cada registros de estos catálogos. Dicho de otra forma, no se puede borrar un grupo al cual esté asociado algún activo fijo, ni se pueden modificar las referencias que tengan asociación, ni tampoco podemos registrar una referencia que ya existe, etc. Todas estas validaciones serán previstas en el sistema.

Otra función de gran importancia que incluiremos en este bloque será el control de las fechas, por medio del cual el usuario podrá efectuar cierres de periodos que son muy importantes para otras áreas, ya que con estos datos la empresa determinará los montos acumulados en sus cuentas de pasivo.

### **3.3.5 Módulo de reportes**

Uno de los propósitos fundamentales del manejo de la información en el sistema de activos fijos es, sin duda, la extracción de resultados en forma lógica y compacta.

Por medio de éstos la empresa puede tomar decisiones o acciones bien fundamentadas. Es por ello que un módulo de reportes es fundamental en el sistema, el diagrama de la **figura 3.7**. nos muestra los principales reportes que debe generar el sistema.



**Figura 3.7. Diseño conceptual del módulo de reportes**

Cabe aclarar que cada uno de estos reportes debe contar con dos alternativas de salida: a papel o a disco, y también debe considerar ciertos rangos de fechas o períodos. Los reportes marcados en el diagrama son los que se consideran fundamentales, sin embargo, pueden adicionarse nuevos reportes, de acuerdo con las necesidades futuras.

### 3.3.6 Módulo de cifras de control

Todo sistema bien fundamentado debe contar con controles para las entradas de información al mismo, nuestro sistema no será la excepción, pues contará con un módulo que controle las altas y bajas del sistema, el diagrama del módulo de cifras de control lo podemos ver en la **figura 3.8**.

Con este módulo nosotros pretendemos validar todas las altas y bajas de activos fijos, la idea es que cuando alguna persona registre o borre información, al final de su sesión deberá informar el total de altas, bajas y los montos de cada uno de estos conceptos, entonces el sistema validará las cifras de control contra lo que se capturó, al final deberá indicar si la captura fue correcta o no.



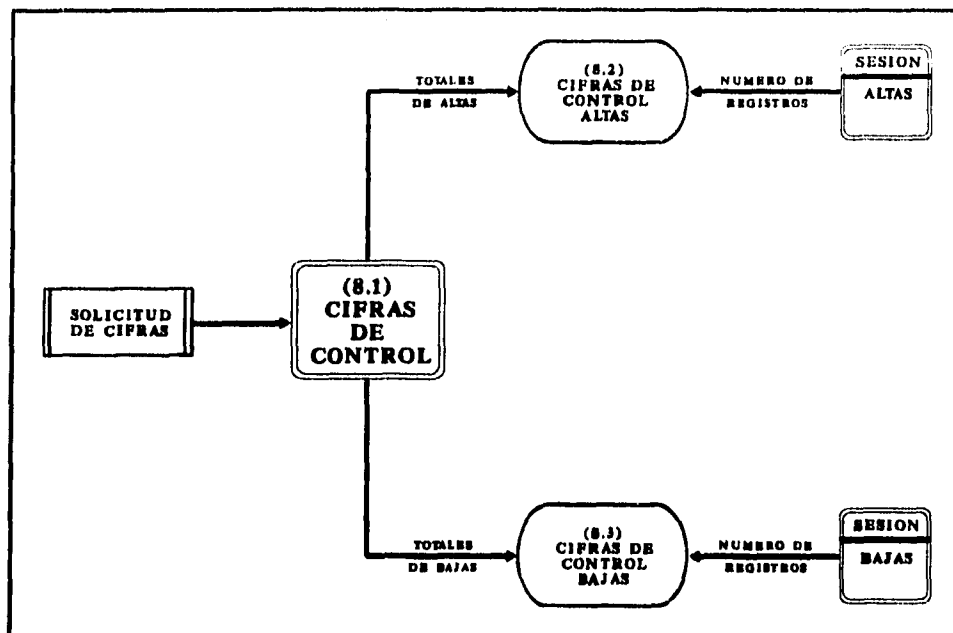


Figura 3.8. Diseño conceptual de módulo de cifras de control.

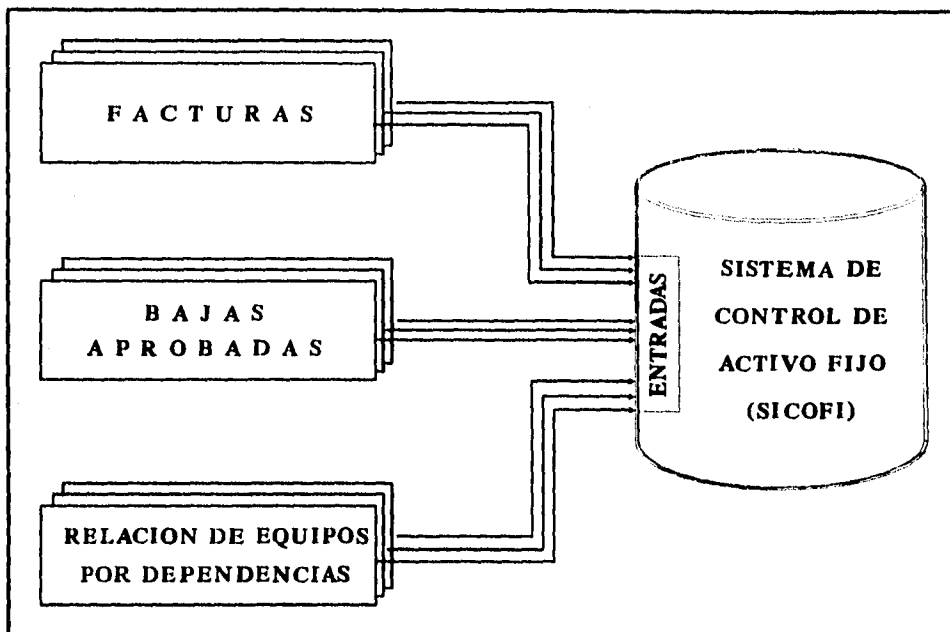
### 3.4 Diseño de entradas al sistema

Consideramos como entradas al sistema a todos los datos que afectan al mismo, éstos serán agrupados y transformados en resultados para su futuro aprovechamiento.

En base al análisis previo que se realizó para definir el problema, podemos detectar tres grandes flujos de información de entrada al sistema, los cuales pueden ser observados en forma esquemática en la *figura 3.9*. que se muestra a continuación.

Como se aprecia en el diagrama, la información de entrada al sistema son todos los documentos que determinan los movimientos de los activos dentro de la institución, los cuales son:

**Facturas:** Estos documentos serán la fuente principal para el registro de altas de activos fijos. Dichas facturas son generadas y proporcionadas por los Centros de Información Presupuestal (CIP'S), quienes se encargan de realizar todas las compras de activos en la institución.



*Figura 3.9. Diseño conceptual de entradas al sistema*

**Bajas aprobadas:** En la institución bancaria se tiene la siguiente política: si un departamento desea dar de baja algún activo, éste deberá de informar dicha decisión a un Comité de Bajas, el cual se encargará de aprobar dicha baja, para posteriormente enviar un documento de baja cuando el activo es considerado como inservible u obsoleto.

También puede darse una reasignación del mismo, si se considera adecuado. Cabe mencionar que el comité de bajas también se encarga de vender o desechar todos los activos dados de baja.

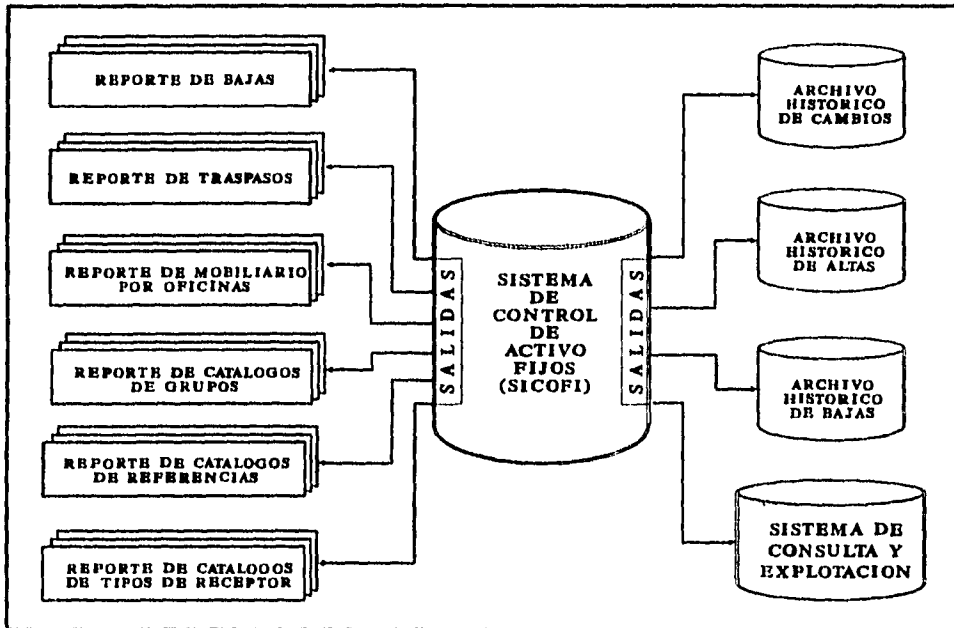
**Relación de equipo por dependencia:** Otra política dentro de la institución es: todas y cada una de las Dependencias Administrativas deberán informar periódicamente a la Oficina de Contabilidad de Capital acerca de los movimientos que hubieran tenido en la asignación de los activos fijos.

Estos informes son especialmente importantes en el caso de los CIP'S, ya que en base a esto son distribuidas las nuevas adquisiciones en las diferentes áreas del banco.

Después de analizar las diferentes entradas que fluyen por el sistema, ahora estudiaremos las salidas del mismo.

### 3.5 Diseño de salidas del sistema

Serán consideradas como salidas del sistema toda la información que después de ser recibida y procesada por el mismo, genere un archivo de bases de datos o un reporte en papel o disco, estas salidas son ilustradas en la **figura 3.10**.



**Figura 3.10. Diseño conceptual de salidas del sistema**

De la figura se desprende que la mayoría de las salidas serán reportes impresos en papel o registrados en disco. El sistema generará también tres bases de datos, y en adición se contará con la alternativa de conectarse a un subsistema de consulta y explotación.

- Archivo histórico de bajas: Contendrá los registros que se den de baja y la fecha en que esto ocurra.
- Archivo histórico de cambios: Contendrá los registros que se modifiquen, así como la fecha del cambio y la información del registro antes de la modificación.
- Archivo de altas: Contendrá los registros adicionados el último mes.
- Reporte de bajas: Será un reporte selectivo por mes en el cual ocurra una baja.
- Reporte de traspasos: Reflejará los movimientos del activo fijo entre las diferentes dependencias administrativas, también será selectivo por mes.

- Reporte de Activos por Oficina: Reportará el activo asignado a una dependencia previamente seleccionada.
- Reporte de Catálogos de Grupos: Informará la clave y descripción de los artículos existentes en la institución.
- Reporte de Catálogo de Referencias: Mostrará las claves, nombres, titulares y ubicación de las diferentes dependencias de la institución.
- Reporte de Catálogo de Tipos de Receptor: Se tendrá los diferentes tipos de asignaciones que puede tener cada artículo en las dependencias.
- Sistema de Consulta y Explotación: Este subsistema será generado para obtener reportes definidos por el propio usuario, dependiendo de sus necesidades en particular. La liga entre los sistemas se dará a través del archivo maestro y los diferentes catálogos.

### 3.6 Definición de archivos

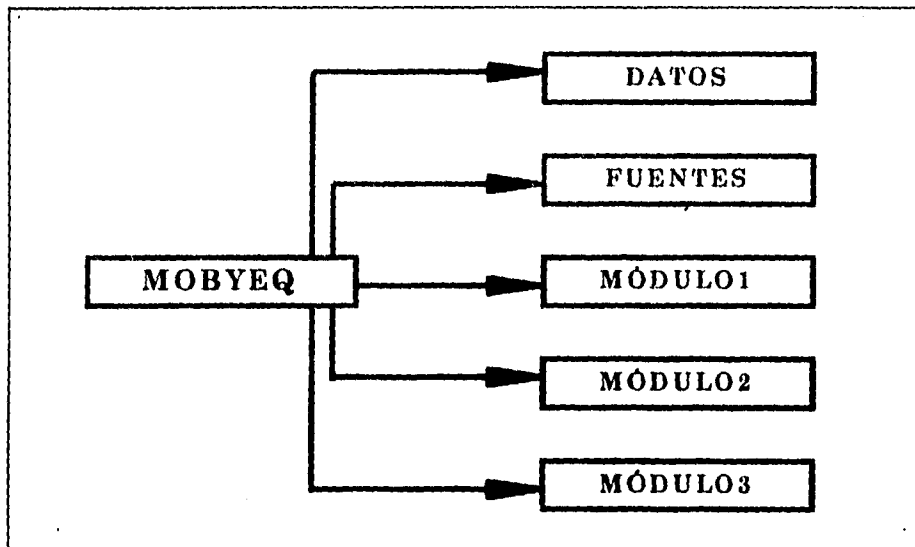
Antes de entrar a la definición de los archivos del Sistema de Control y Actualización del Activo Fijo para una Institución Bancaria, explicaremos la distribución de directorios que nos parece adecuada para agrupar las diferentes partes que integran nuestro sistema. Contaremos con un directorio principal que llamaremos **MOBYEQ (Mobillario y Equipo)**, éste a su vez se subdivide en los siguientes subdirectorios:

- **DATOS.** Contendrá todos los archivos de bases de datos (DBF) e índices (IDX) que van a ser usados por el sistema..
- **FUENTES.** En este subdirectorio colocaremos todo los archivos de código fuente necesarios para la Captura y Actualización de Registros.
- **MODULO1.** En esta parte colocaremos el código fuente de Consulta y Explotación correspondiente a Generación de Archivos.
- **MODULO2.** En esta parte colocaremos el código fuente de Consulta y Explotación correspondiente a Creación de Filtros y Ordenamientos.
- **MODULO3.** En esta parte colocaremos el código fuente de Consulta y Explotación correspondiente a Proceso de Generación de Reportes.

A continuación de muestra en la **figura 3.11**, el directorio principal con sus respectivos subdirectorios.

Para fines de nuestro sistema consideramos que una base de datos es un conjunto de datos relacionados, el almacenamiento de datos se logra empleando uno o más archivos.

Es posible tener acceso a todos los archivos de una base de datos desde una computadora o desde cualquier computadora interconectada si la base de datos está distribuida sobre varias computadoras.



*Figura 3.11. Distribución de directorios.*

Se usan procedimientos en estas computadoras para introducir datos, almacenar datos en los archivos y procesar datos en respuesta a solicitudes de información.

Un *archivo* se define como un conjunto de registros semejantes conservado en dispositivos de computadora de almacenamiento secundario. Nuestro sistema estará conformado de 19 archivos los cuales describimos a continuación:

**MOBYEQ.** Archivo Maestro de mobiliario y equipo. Contendrá los datos de los artículos que integran el activo fijo de la Institución: Fecha de adquisición (mes y año), Clave del artículo (grupo, conformado por cuatro niveles), Número de serie, Número de serie del artículo del que forma parte (si existe), Referencia, Sector e Importe. El archivo deberá ser indexado por tres criterios diferentes:

- Clave de Grupo (MOBY\_IDX).
- Número de Serie (MOBY\_NSE).
- Referencia (MOBY\_REF).

**CATGRUPO.** Archivo que contendrá el catálogo de artículos: Clave, Descripción, Indicativo de que requiere número de serie (OPC1), Indicativo de corte en manejo de texto (OPC2, para reportes y despliegues), y el indicativo de que pertenece a (OPC3). El archivo debe ser indexado por:

- Clave de grupo (GRUP\_IDX).

**CATREFER.** Archivo que contiene el catálogo de las dependencias administrativas del Banco. Contiene claves de Referencia, Descripción, Titular, Ubicación y Área. Deberá ser indexado por:

- Clave de referencia (REFE\_IDX).

**TABRECEP.** Archivo que contendrá los tipos de receptor (asignación) que puede tener un artículo (definitivo, préstamo, almacén, distribuidor, etc.). Está indexado por:

- Clave de tipo de Receptor (RECE\_IDX).

**ALTAS.** Archivo que contiene las adiciones de mobiliario y equipo durante el mes actual. El formato de registro es igual al del archivo maestro.

**BAJAS.** Contiene todos los artículos que se han dado de baja durante el último año. Adicionalmente contiene la descripción del artículo al momento en que se dio la baja, así como la fecha.

**CAMBIOS.** Contiene los cambios ocurridos a los artículos durante el último año, y la fecha en la que ocurrió el cambio. Se tiene también la información del artículo antes de aplicar el cambio.

**SESION.** Contendrá los números de registro e importes de los artículos que se dan de alta en una sesión de captura.

**SESBAJ.** Contendrá los números de registro e importes de los artículos que se dan de baja en una sesión de captura.

**MOBAUX.** Será un archivo auxiliar que se utiliza cuando necesitemos filtrar información del archivo maestro para efectuar bajas o actualizaciones de artículos.

**GRUPAUX.** Será un archivo auxiliar para ayudar en el registro de artículos cuando no se conozca la clave del grupo al cuál corresponde.

**CUENTAS.** Contendrá la relación entre números de cuentas contables y grupos de mobiliario y equipo a utilizar en el sistema.

**ARCHIVO.** Contendrá la relación de todos los reportes generados en el Módulo de Consulta y Explotación (MCE).

**PROCESOS.** Este archivo contendrá los criterios para ordenar, agrupar y filtrar información de cualquier archivo de la base de datos para generar un reporte en el MCE.

**COLUMP.** Contendrá las características de los campos, de la longitud, del orden de precedencia en un reporte generado en el MCE.

**ORDENES.** Guarda el criterio de ordenación de un archivo generado en el Módulo de Consulta y Explotación.

**NOMFILT.** Guarda los filtros creados para un reporte en el Módulo de Consulta y Explotación.

**AFILTROS.** Es un archivo de área de trabajo al cual se cargan unos filtros previamente catalogados en NOMFILT para generar un reporte en el MCE.

### 3.7 Definición de campos

Una vez analizada la estructura de archivos con la cual contará nuestro sistema, el siguiente paso a seguir es la definición de los campos de cada base de datos, esto es fundamental para nuestros fines, ya que una adecuada estructura de campos permite entre otras cosas un fácil acceso a cualquier información y reduce el código de programación.

Nosotros consideramos para nuestros procesos que un *campo almacenado* es la unidad de datos más pequeña que se halla almacenada en la base de datos.

Un *registro almacenado* es un conjunto de campos almacenados agrupados y asociados entre sí. La relación entre ellos consiste en que todos representan propiedades de alguna parte específica. En la mayoría de los sistemas la ocurrencia de registro almacenado es la unidad de acceso a la base de datos.

En la mayoría de los sistemas actuales, un registro lógico de una aplicación es idéntico a un registro almacenado; sin embargo, esto no siempre se observa en un sistema de base de datos por que el Administrador de Base de Datos puede efectuar cambios en la estructura de almacenamiento, es decir, en los campos y registros almacenados, en tanto que los campos y registros lógicos correspondientes no cambian.

Estructura de la base de datos : **CUENTA.DBF** Alias: **CUENTA**

Field	Field name	Type	Width	Dec	Start	End
1	CUENTA	Numeric	2		1	2
2	MAYOR	Numeric	3		3	5
3	NOMBRECTA	Character	50		6	55

Estructura de la base de datos : **SESBAJ.DBF** Alias: **SESBAJ**

Field	Field name	Type	Width	Dec	Start	End
1	NUMREG	Numeric	6		1	6
2	IMPORTE	Numeric	19	2	7	25

Estructura de la base de datos : **SESION.DBF** Alias: **SESION**

Field	Field name	Type	Width	Dec	Start	End
1	NUMREG	Numeric	6		1	6
2	IMPORTE	Numeric	19		27	25

Estructura de la base de datos : **GRUPAUX.DBF** Alias: **GRUPAUX**

Field	Field name	Type	Width	Dec	Start	End
1	NIVEL	Numeric	2		1	2
2	DESCRIPCIO	Character	50		3	52

Estructura de la base de datos : **CAMBIOS.DBF** Alias: **CAMBIOS**

Field	Field name	Type	Width	Dec	Start	End
1	ANIO_CAMB	Numeric	2		1	2
2	MES_CAMB	Numeric	2		3	4
3	NUM_MOV	Numeric	5		5	9
4	ANT_DESP	Character	1		10	10
5	MES	Numeric	2		11	12
6	ANIO	Numeric	2		13	14
7	MAYOR	Numeric	3		15	17
8	NIVEL_1	Numeric	2		18	19
9	NIVEL_2	Numeric	2		20	21
10	NIVEL_3	Numeric	2		22	23
11	NUM_SERIE	Character	20		24	43
12	PERTENECEA	Character	20		44	63
13	REFERENCIA	Character	3		64	66
14	TIPO_RECEP	Numeric	1		67	67
15	IMPORTE	Numeric	18	2	68	85
16	SECTOR	Character	1		86	86

Estructura de la base de datos : **ALTAS.DBF** Alias: **ALTAS**

Field	Field name	Type	Width	Dec	Start	End
1	MES	Numeric	2		1	2
2	ANIO	Numeric	2		3	4
3	MAYOR	Numeric	3		5	7
4	NIVEL_1	Numeric	2		8	9



Field	Field name	Type	Width	Dec	Start	End
5	NIVEL_2	Numeric	2		10	11
6	NIVEL_3	Numeric	2		12	13
7	NUM_SERIE	Character	20		14	33
8	PERTENECEA	Character	20		34	53
9	REFERENCIA	Character	3		54	56
10	TIPO_RECEP	Numeric	1		57	57
11	IMPORTE	Numeric	18	2	58	75
12	SECTOR	Character	1		76	76

Estructura de la base de datos : **MOBAUX.DBF** Alias: **MOBAUX**

Field	Field name	Type	Width	Dec	Start	End
1	NUMREG	Numeric	6		1	6
2	NUMREGA	Numeric	6		7	12
3	NIVEL_1	Numeric	2		13	14
4	NIVEL_2	Numeric	2		15	16
5	NIVEL_3	Numeric	2		17	18
6	ANIO	Numeric	2		19	20
7	MES	Numeric	2		21	22
8	NUM_SERIE	Character	20		23	42
9	PERTENECEA	Character	20		43	62
10	REFERENCIA	Character	3		63	65
11	TIPO_RECEP	Numeric	1		66	66
12	IMPORTE	Numeric	18	2	67	84
13	SECTOR	Character	1		85	85

Estructura de la base de datos : **BAJAS.DBF** Alias: **BAJAS**

Field	Field name	Type	Width	Dec	Start	End
1	ANIOBAJ	Numeric	2		1	2
2	MESBAJ	Numeric	2		3	4
3	ANIOREG	Numeric	2		5	6
4	MESREG	Numeric	2		7	8
5	MAYOR	Numeric	3		9	11
6	NIVEL_1	Numeric	2		12	13
7	NIVEL_2	Numeric	2		14	15
8	NIVEL_3	Numeric	2		16	17
9	DESCGRUPO	Character	100		18	117
10	NUM_SERIE	Character	20		118	137
11	PERTENECEA	Character	20		138	157
12	REFERENCIA	Character	3		158	160
13	TIPO_RECEP	Numeric	1		161	161
14	IMPORTE	Numeric	18	2	162	179
15	SECTOR	Character	1		180	180

Las bases de datos mencionadas en estos párrafos, son sólo una parte del total que es manejado por el sistema, si desea más información, ver el "Apéndice A" al final de la tesis.

A<sub>1</sub>

### 3.8 Diagrama de entidad-relación

Antes de definir el diagrama de Entidad-Relación para nuestro sistema, describiremos la arquitectura de un sistema de base de datos que nos proporcionará un marco de referencia y que nos servirá de base para el diseño de nuestro diagrama Entidad-Relación.

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

Nosotros consideraremos para nuestro diseño que el modelo conceptual de datos en un diagrama Entidad-Relación permite diseñar un fenómeno real para un sistema de base de datos, a través de diagramas con entidades y sus relaciones que se construye con los elementos siguientes.

- *Rectángulos que representa una ENTIDAD.* Cada rectángulo representa un conjunto de objetos sobre los que queremos guardar datos homogéneos (por ejemplo: empleados, artículos, etc.). Es decir es un objeto que contiene ciertos atributos que lo caracterizan, dado que es el elemento básico que se desea administrar, guardándose información del mismo.
- *Rombos que representa una RELACIÓN.* Cada rombo representa una asociación entre los elementos de dos o más rectángulos que se unen con líneas al rombo.
- *Óvalos que representan los ATRIBUTOS.* Cada óvalo representa un conjunto de valores simples, es decir atributos. Cada óvalo se une con líneas a los rectángulos o rombos para indicar cuáles son los tributos de éstos.

A continuación presentamos los Modelos Entidad-Relación de las diferentes Bases de Datos donde fluye o genera información; y representa la asociación de datos entre ellas.

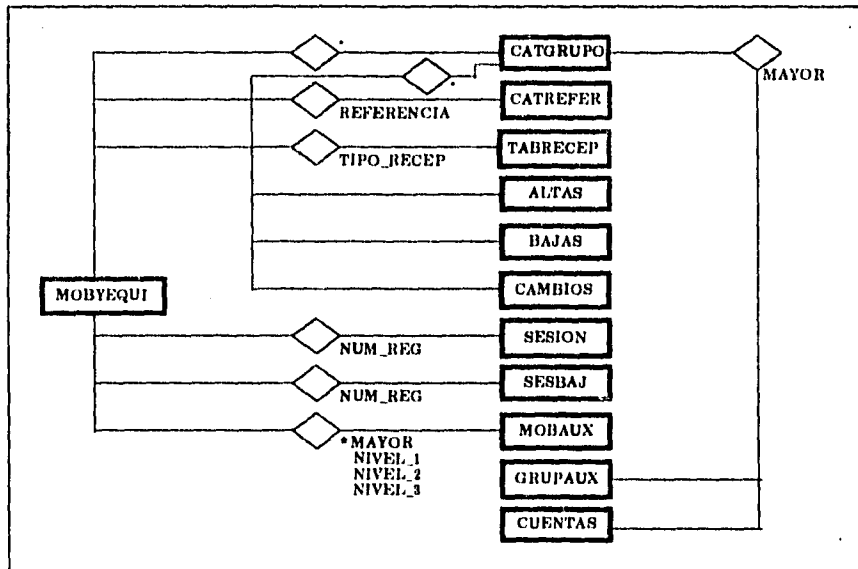
Para los efectos se diseñaron varios modelos donde se consideran los principales archivos del Sistema de Control y Actualización del Activo Fijo.

En primer término observaremos el Modelo General de Entidad-Relación del Sistema de Control y Actualización del Activo Fijo para una Institución Bancaria (SICOFI), y adicionalmente se muestran en forma desglosada las relaciones que existen entre cada una de las Bases de Datos.

Las entidades con las que guarda relación el sistema SICOFI son las siguientes:

CATGRUPO	SESION	CAMBIOS
CATREFER	SESBAJ	BAJAS
CUENTAS	TABRECEP	MOBAUX
ALTAS	GRUPAUX	

A continuación se muestra la **figura 3.12.** el Modelo General de Entidad-Relación del Sistema de Control y Actualización del Activo Fijo.



**Figura 3.12.** Diagrama general del modelo entidad-relación del sistema de SICOFI

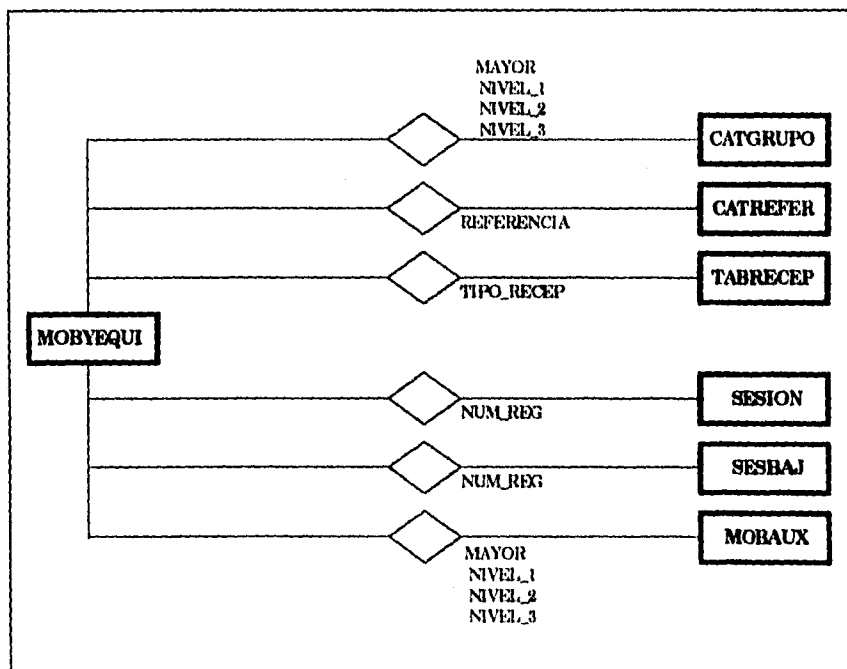
La Base de Datos de MOBYEQ sirve como concentrador de información propia del MOBYEQ y de información asociada al mismo. Las entidades con las que se relaciona son:

- CATGRUPO (Se relaciona con MOBYEQ por los campos *mayor, nivel\_1 al 3*)
- CATREFER (Se relaciona con MOBYEQ por el campo *referencia*)
- TABRECEP (Se relaciona con MOBYEQ por el campo *tipo\_recep*)
- SESION (Se relaciona con MOBYEQ por el campo *num\_reg*)
- SESBAJ (Se relaciona con MOBYEQ por el campo *num\_reg*)
- MOBAUX (Se relaciona con MOBYEQ por los campos *mayor, nivel\_1 al 3*)

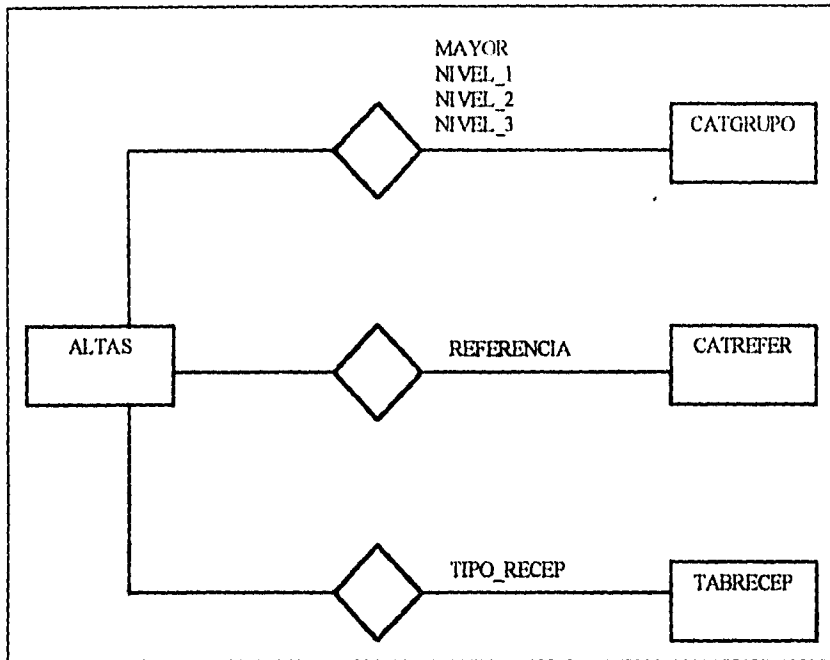
La **figura 3.13** muestra la relación que existe entre el archivo maestro y las entidades.

El modelo para los archivos de ALTAS, BAJAS y CAMBIOS es muy similar, debido a que guardan relación con los archivos CATGRUPO, CATREFER y TABRECEP, a través de los campos: Mayor, Nivel\_1, Nivel\_2, Nivel\_3, Referencia y Tipo\_Recep. En la **figura 3.14**, se muestra el modelo Entidad-Relación para ALTAS, el cual sirve de referencia para el de BAJAS y CAMBIOS.

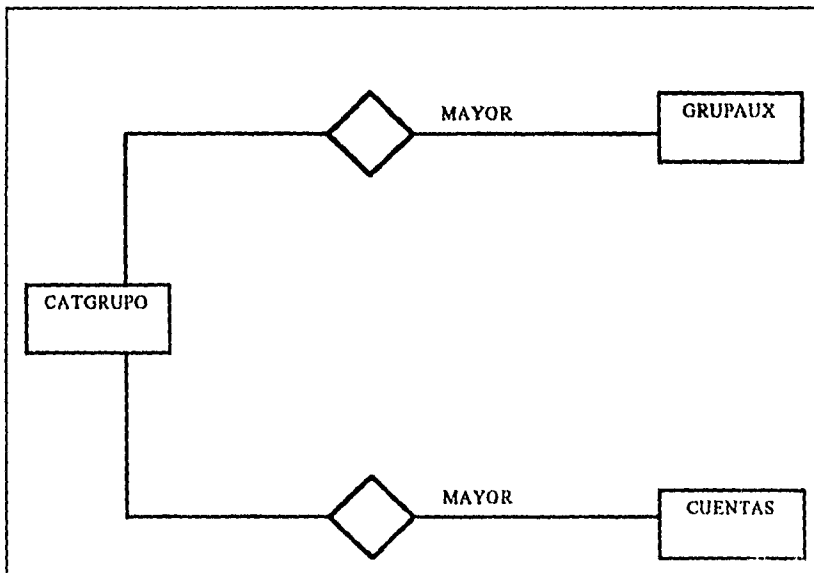
El modelo para el archivo CATGRUPO se muestra en la **figura 3.15**, y guarda relación con los archivos GRUPAUX y CUENTAS a través del campo Mayor.



**Figura 3.13. Modelo Entidad-Relación de MOBYEQ.**



**Figura 3.14 Modelo Entidad-Relación de ALTAS.**



**Figura 3.15. Modelo Entidad-Relación de CATGRUPO.**

### 3.9 Referencias cruzadas (programas-archivos)

Las Referencias Cruzadas (Programas-Archivos) nos sirven para identificar en que programas se utilizan determinado archivo, para que en el momento en que se hagan modificaciones a la base de datos se pueda tener a la mano los programas que pueden salir afectados y poder detectar la falla a tiempo y corregirla.

En el momento en que se hacen modificaciones a la base de datos se tiene que hacer una compilación de todos los programas que usan a la base datos modificada y al mismo tiempo checar si el cambio no afecta a resultados obtenidos en los diferentes programas.

Una parte de las Referencias Cruzadas de Programas-Archivos del Sistema que se va a desarrollar se muestran a continuación, si desea ver todas las referencias cruzadas consulte el "Apéndice A2" que se encuentra al final de la tesis.

ARCHIVOS	PROGRAMAS	ARCHIVOS	PROGRAMAS
ARCHIVOS.DBF	ARCHIVOS.PRG	CUENTA.DBF	MENU.PRG
SESBAJ.DBF	CIFRBAJ.PRG	SESION.DBF	CIRFCTRL.PRG
CAMBIOS.DBF	MENU.PRG	ALTAS.DBF	MENU.PRG
MOBAUX.DBF	MENU.PRG	GRUPAUX.DBF	MENU.PRG
BAJAS.DBF	MENU.PRG	KMOBYEQ.DBF	MENU.PRG
TABRECEP.DBF	MENU.PRG ARCHIVOS.PRG	CATREFER.DBF	ARCHIVOS.PRG CONSINVEN.PRG
CATGRUPO.DBF	MENU.PRG ARCHIVOS.PRG CONSINVEN.PRG EMREPOR.PRG DEPRECIA.PRG MENUPARE.PRG	MOBYEQUI.DBF	MENU.PRG ARCHIVOS.PRG BORRARC.PRG CONSINVEN.PRG CREASALD.PRG DEPRECIA.PRG
PROCESOS.DBF	ARCHIVOS.PRG	OPCIOP.DBF	ARCHIVOS.PRG

### 3.10 Referencias cruzadas (archivos-campos)

Las Referencias Cruzadas (Archivos-Campos) se utilizan para conocer en que archivos se encuentran determinados campos, con el fin de que si se modifica un campo de una base de datos se conozca en que otras bases de datos donde se encuentra el campo y proceder a modificarla para evitar errores o resultados erróneos. Una porción de la Referencia Cruzada Archivos-Campos diseñada para el sistema se muestra a continuación, si desea más detalles vea por favor el "Apéndice A3" al final de la tesis.

NOMBRE DEL CAMPO	BASES DE DATOS
ANCHONOM	COLUMP.DBF
ANCHOPAG	OPCIOP.DBF
ARCHIVO	PROCESOS.DBF
ANIOBAJ	BAJAS.DBF
ANIOREG	BAJAS.DBF
ANIO_CAMB	CAMBIOS.DBF
CONTENIDO	COLUMP.DBF
CUENTA	CUENTA.DBF
DESCGRUPO	BAJAS.DBF
MESBAJ	BAJAS.DBF
MESREG	BAJAS.DBF
NOMBRECTA	CUENTA.DBF
NOREC	COLUMP.DBF
NUMREGA	MOBAUX.DBF
NUM_MOV	CAMBIOS.DBF
NUM_PLACA	MOBYEQUI.DBF
NUM_REGCON	MOBYEQUI.DBF
NUM_REGIST	MOBYEQUI.DBF
NUM_SERIEA	MOBYEQUI.DBF
OPC1	CATGRUPO.DBF
OPC2	CATGRUPO.DBF
OPC3	CATGRUPO.DBF
OPERADOR	AFILTROS.DBF
ORDEN	PROCESOS.DBF
PASSWORD	KMOBYEQ.DBF
TITPAG1	OPCIOP.DBF

NOMBRE DEL CAMPO	BASES DE DATOS
ANT_DESP	CAMBIOS.DBF
BORRADO	MOBYEQUI.DBF
CAMPO	AFILTROS.DBF
COMENTARIO	ARCHIVOS.DBF
COMPRESOR	PROCESOS.DBF
CONECTOR	AFILTROS.DBF
DIRECCION	CATREFER.DBF
DOBLESFACI	OPCIOP.DBF
FECHA	OPCIOP.DBF
MES_CAMB	CAMBIOS.DBF
MILES	OPCIOP.DBF
NOREF	COLUMP.DBF
NUMERAPAG	OPCIOP.DBF
TOTALCOLUM	COLUMP.DBF
UBICACION	CATREFER.DBF
REFE_ANTER	CATREFER.DBF
REPORTE	PROCESOS.DBF
RESPONSAB	CATREFER.DBF
SALTADESP	OPCIOP.DBF
SALTANTES	OPCIOP.DBF
SINOMCTA	COLUMP.DBF
SINOMREC	COLUMP.DBF
SINOMREF	COLUMP.DBF
SINUMYCTA	COLUMP.DBF
SINUMYREC	COLUMP.DBF
SINUMYREF	COLUMP.DBF

### 3.11 Diccionario de datos

El Diccionario de Datos se compone de dos secciones, la primera son la relación de los archivos y la segunda las definiciones de los campos de todos los archivos del sistema. El Diccionario de Datos se utiliza como referencia para futuras modificaciones al sistema, así como para el mantenimiento del mismo, es de gran ayuda para los analistas que no participan en el desarrollo del sistema y que tienen que darle mantenimiento al mismo.

#### Relación de archivos:

BASE DE DATOS	ALIAS
CUENTA.DBF	CUENTA
SESBAJ.DBF	SESBAJ

BASE DE DATOS	ALIAS
CATGRUPO.DBF	CATGRUPO
MOBYEQUI.DBF	MOBYEQUI

BASE DE DATOS	ALIAS
SESION.DBF	SESION
GRUPAUX.DBF	GRUPAUX
CAMBIOS.DBF	CAMBIOS
ALTAS.DBF	ALTAS
MOBAUX.DBF	MOBAUX
BAJAS.DBF	BAJAS
TABRECEP.DBF	TABRECEP
CATREFER.DBF	CATREFER

BASE DE DATOS	ALIAS
ARCHIVOS.DBF	ARCHIVOS
PROCESOS.DBF	PROCESOS
OPCIOP.DBF	OPCIOP
COLUMP.DBF	COLUMP
ORDENES.DBF	ORDENES
NOMFILT.DBF	NOMFILT
AFILTROS.DBF	AFILTROS

### Definición de campos:

NOMBRE DEL CAMPO	TIPO	LONG.	DECIM.
ANCHONOM	Numérico	3	0
ANCHOPAG	Numérico	3	0
ANIO	Numérico	2	0
ANIOBAJ	Numérico	2	0
ANIOREG	Numérico	2	0
ANIO_CAMB	Numérico	2	0
ANT_DESP	Carácter	1	0
ARCHIVO	Carácter	8	0
BORRADO	Carácter	1	0
CAMPO	Carácter	25	0
COMENTARIO	Carácter	40	0
COMENATRIO	Carácter	60	0
COMENTARIO	Carácter	60	0
COMPRESOR	Carácter	8	0
CONECTOR	Carácter	5	0
CONTENIDO	Carácter	10	0
DOBLES PACI	Lógico	1	0
CUENTA	Numérico	2	0
DESCGRUPO	Carácter	100	0
DIRECCION	Carácter	50	0
FECHA	Lógico	1	0
FECHA_CREA	Fecha	8	0
FILTRO	Carácter	8	0
IMPORTE	Numérico	18	2
LINEASXPAG	Numérico	2	0
MAYOR	Numérico	3	0
MES	Numérico	2	0

Hay que aclarar que en esta definición de campos sólo relacionamos algunos, sin embargo, si usted desea verlos todos, puede consultarlos en el "Apéndice A4" que se encuentran al final de la tesis.



## **CAPÍTULO 4 DESARROLLO DEL SISTEMA**

El propósito de este capítulo es presentar parte del código fuente del sistema y explicar éste en forma breve, además, se relaciona el código con las pantallas que conforman el sistema, para que el lector pueda ver la opción que estamos efectuando y la asocie con un conjunto de instrucciones. También incluimos los formatos de los reportes que proporciona el sistema en el apartado reportes.

Debemos aclarar que este capítulo no pretende definir todo el código fuente, es simplemente un ejemplo del desarrollo del mismo sin entrar a grandes detalles.

### **4.1 Programación del menú principal**

El menú principal consta de 7 opciones, como se muestra en la *figura 4.1*.

- |                   |                         |                             |
|-------------------|-------------------------|-----------------------------|
| <b>1. ALTAS</b>   | <b>4. MANTENIMIENTO</b> | <b>6. RESPALDO</b>          |
| <b>2. BAJAS</b>   | <b>5. REPORTES</b>      | <b>7. CIFRAS DE CONTROL</b> |
| <b>3. CAMBIOS</b> |                         |                             |

#### **1. Altas**

Consiste en la adición de nuevos registros al Archivo Maestro, se entra a ella al seleccionar la opción de *Altas* en el menú principal, por lo cual se despliega un menú de dos opciones, éstas son:

- |                                    |                                  |
|------------------------------------|----------------------------------|
| 1. Altas de registros Individuales | 2. Altas de registro por Factura |
|------------------------------------|----------------------------------|

## 2. Bajas

Al seleccionar la opción de bajas en el menú principal, aparece la pantalla para seleccionar registros a dar de baja.

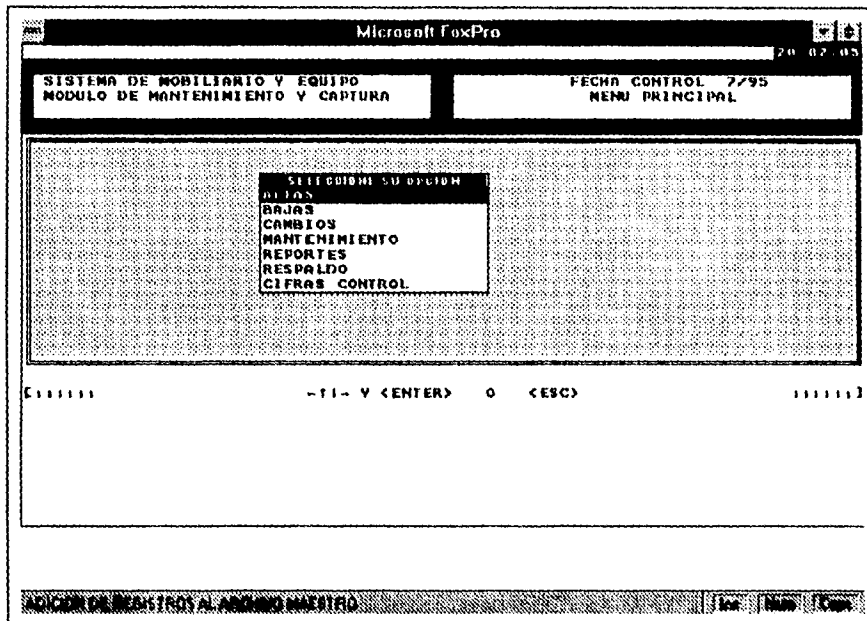


Figura 4.1. Pantalla del menú principal.

## 3. Cambios

La opción de cambios permite modificar y actualizar los registros.

## 4. Mantenimiento

La opción de mantenimiento del *Menú Principal* permite la actualización de los catálogos utilizados en el Sistema de Control de Activo Fijo. Estos son:

- |                                  |                            |
|----------------------------------|----------------------------|
| 1. Catálogo de grupos            | 2. Catálogo de referencias |
| a) altas                         | a) altas                   |
| b) bajas                         | b) bajas                   |
| c) cambios                       | c) cambios                 |
| 3. Catálogo de tipos de receptor | 4. Fecha de control        |
| a) altas                         |                            |
| b) cambios                       |                            |
| c) bajas                         |                            |

## 5. Reportes

La opción de reportes del *Menú Principal* contiene los diferentes tipos de receptores que es posible obtener en el módulo de *Mantenimiento y Captura*. Estos son:

1. Reporte del catálogo de grupos
2. Reporte del catálogo de referencias
3. Reporte de bajas
4. Reporte de cambios
5. Reporte de lo capturado

## 6. Respaldo

La opción de respaldo del *Menú Principal* permite salvar los archivos de datos utilizados por SICOFI, así como la programación en disquetes.

## 7. Cifras de control

Esta opción se utiliza para verificar totales cuando se da de alta o baja un grupo de artículos.

A continuación presentamos un listado parcial del código fuente del programa menu.prg del sistema:

1. \* Programa: MENU.PRG
2. \* Nota.....: Programa de captura y mantenimiento del Activo Fijo.
3. \*
4. \*
5. \*
6. SET TALK OFF
7. SET SAFETY OFF
8. SET CONFIRM OFF
9. SET CONSOLE OFF
10. SET SHADOWS ON
11. SET FULLPATH OFF
12. SET EXCLU OFF
13. SET PATH TO CAMOBYEQDATOS
14. SET PROCEDURE TO RUTINAS  
SET MESSAGE TO 24 CENTER
15. PUBLIC werrgru,windividual,manio,mmes,mmayor,mnivel\_1,mnivel\_2,mnivel\_3;
16. mnum\_serie,mptenecea,mreferencia,mtipo\_recep,mimporte,wCampos;
17. mnum\_articulos,wpad,wsalir,wbar,wOPC1,wOPC2,wOPC3,mSector;
18. mDescgru,mDescref,wMovimiento,wLinea,wFiltro,wCampo,CambAD;
19. wBajGrup,wBajRef,wBajMac,mDesgruM,mDesgru1,mDesgru2,mDesgru3
20. RESTORE FROM PARM\_CTRL ADDITIVE
21. RESTORE FROM MEM\_IMPR ADDITIVE
22. RESTORE FROM CAMOBYEQDATOS/FECH\_CTRL ADDITIVE
23. STORE I TO wbar
24. STORE .T. TO werrgru,windividual
25. STORE .F. TO wOPC1,wOPC2,wOPC3,wBajGrup,wBajRef,wBajMac

26. STORE " " TO wblanco,wMovimiento,wCampo  
 27. STORE 0 TO wLinea  
 28. DO ABRE  
 29. SET COLOR TO  
 30. DEFINE WINDOW win\_msg FROM 10,1 TO 15,60 && Ventana para mensajes  
 31. DEFINE WINDOW win\_dat\_cpl FROM 12,20 TO 24,60 ;  
 32. TITLE "DATOS ADICIONALES POR REGISTRO" && Ventana para datos  
 complementarios  
 33. DEFINE POPUP pop\_repor FROM 12,23 TITLE "MENU DE REPORTES"  
 34. DEFINE BAR 1 OF pop\_repor PROMPT "Reporte del cat logo de Grupos" ;  
 35. MESSAGE "IMPRESION DE TODO EL CATALOGO DE GRUPOS"  
 36. DEFINE BAR 2 OF pop\_repor PROMPT "Reporte del cat logo de Referencias" ;  
 37. MESSAGE "IMPRESION DE TODO EL CATALOGO DE REFERENCIAS"  
 38. DEFINE BAR 3 OF pop\_repor PROMPT "Reporte Bajas de Mobiliario y Equipo" ;  
 39. MESSAGE "IMPRESION DE BAJAS SELECTIVO POR MES"  
 40. DEFINE BAR 4 OF pop\_repor PROMPT "Reporte Cambios de Mobiliario y Equipo" ;  
 41. MESSAGE "IMPRESION DE CAMBIOS SELECTIVO POR MES"  
 42. DEFINE BAR 5 OF pop\_repor PROMPT "Reporte Traspasos de Mobiliario y Equipo" ;  
 43. MESSAGE "IMPRESION DE TRASPASOS SELECTIVO POR MES"  
 44. DEFINE BAR 6 OF pop\_repor PROMPT "Reporte de Mobiliario por Oficina" ;  
 45. MESSAGE "IMPRESION DE MOBILIARIO SELECTIVO POR OFICINA"  
 46. ON SELECTION POPUP pop\_repor DO (popup\_catal WITH BAR)  
 47. DEFINE POPUP pop\_Cifra FROM 12,23 TITLE "SELECCIONE LA CIFRA DE  
 CONTROL"  
 48. DEFINE BAR 1 OF pop\_Cifra PROMPT "Cifras de Control de Altas" ;  
 49. MESSAGE "CHECAR IMPORTE Y ARTICULOS DE ALTA EN LA SESION"  
 50. DEFINE BAR 2 OF pop\_Cifra PROMPT "Cifras de Control de Bajas" ;  
 51. MESSAGE "CHECAR IMPORTE Y ARTICULOS DE BAJA EN LA SESION"  
 52. ON SELECTION POPUP pop\_Cifra DO (popup\_catal WITH BAR)  
 53. DEFINE POPUP pop\_Trsp FROM 12,23 TITLE "SELECCIONE EL TIPO DE REPORTE"  
 54. DEFINE BAR 1 OF pop\_Trsp PROMPT "Por Dependencia Distribuidora" ;  
 55. MESSAGE "REPORTE DE TRASPASOS POR DEPENDENCIA DISTRIBUIDORA"  
 56. DEFINE BAR 2 OF pop\_Trsp PROMPT "Por Dependencia Receptora" ;  
 57. MESSAGE "REPORTE DE TRASPASOS POR DEPENDENCIA RECEPTORA"  
 58. ON SELECTION POPUP pop\_Trsp DO (popup\_catal WITH BAR)  
 59. DEFINE POPUP pop\_mante FROM 10,20 TITLE "MENU DE MANTENIMIENTO"  
 60. DEFINE BAR 1 OF pop\_mante PROMPT "Mantenimiento cat logo de Grupos" ;  
 61. MESSAGE "ALTAS, BAJAS Y CAMBIOS A GRUPOS"  
 62. DEFINE BAR 2 OF pop\_mante PROMPT "Mantenimiento cat logo de Referencias" ;  
 63. MESSAGE "ALTAS, BAJAS Y CAMBIOS A REFERENCIAS"  
 64. DEFINE BAR 3 OF pop\_mante PROMPT "Mantenimiento cat logo de Tipos de Receptor" ;  
 65. MESSAGE "ALTAS, BAJAS Y CAMBIOS A TIPOS DE RECEPTOR"  
 66. DEFINE BAR 4 OF pop\_mante PROMPT "Actualizar la Fecha de Control" ;  
 67. MESSAGE "FECHA DE CONTROL DE MOVIMIENTOS"  
 68. ON SELECTION POPUP pop\_mante DO (popup\_catal WITH BAR)  
 69. DEFINE POPUP MENPRIN FROM 8,30 TITLE "SELECCIONE SU OPCION"  
 70. DEFINE BAR 1 OF MENPRIN PROMPT "ALTAS" ;  
 71. MESSAGE "ADICION DE REGISTROS AL ARCHIVO MAESTRO"  
 72. DEFINE BAR 2 OF MENPRIN PROMPT "BAJAS" ;  
 73. MESSAGE "BORRAR REGISTROS DEL ARCHIVO MAESTRO"  
 74. DEFINE BAR 3 OF MENPRIN PROMPT "CAMBIOS" ;  
 75. MESSAGE "ACTUALIZACION DE REGISTROS DEL ARCHIVO MAESTRO"  
 76. DEFINE BAR 4 OF MENPRIN PROMPT "MANTENIMIENTO" ;  
 77. MESSAGE "MANTENIMIENTO DE CATALOGOS"

```

78. DEFINE BAR 5 OF MENPRIN PROMPT "REPORTES" ;
79. MESSAGE "IMPRESION DE BAJAS, CAMBIOS Y CATALOGOS"
80. DEFINE BAR 6 OF MENPRIN.PROMPT "RESPALDO" ;
81. MESSAGE "RESPALDO DE LOS ARCHIVOS DE DATOS DEL SISTEMA"
82. DEFINE BAR 7 OF MENPRIN PROMPT "CIFRAS CONTROL";
83. MESSAGE "CHECAR CIFRAS DE CONTROL DE ALTAS EN LA SESION"
84. ON SELECTION POPUP menprin DO (popup_catal WITH BAR6)
85. DEFINE POPUP pMayor FROM 12,20 PROMPT FIELD Descripcio ;
86. MESSAGE "GRUPOS DE MAYOR"
87. DEFINE POPUP pNiv1 FROM 12,30 PROMPT FIELD Descripcio ;
88. MESSAGE "GRUPOS A NIVEL 1"
89. DEFINE POPUP pNiv2 FROM 12,40 PROMPT FIELD Descripcio ;
90. MESSAGE "GRUPOS A NIVEL 2"
91. DEFINE POPUP pNiv3 FROM 12,50 PROMPT FIELD Descripcio ;
92. MESSAGE "GRUPOS A NIVEL 3"
93. DEFINE POPUP pop_TipRec FROM 19,34 PROMPT FIELD Descripcio ;
94. MESSAGE "Tipos de Receptor Disponibles"
95. DEFINE POPUP pRefer FROM 17,34 PROMPT FIELD Descripcio ;
96. MESSAGE "Referencias Disponibles"
97. ON KEY LABEL F1 DO SIGUE

```

### **Documentación del código fuente del menú principal**

El programa fuente del *Menú Principal* inicia con algunos comentarios referentes al nombre del programa y notas, estos comentarios se declaran de la línea 1 a la 4, esto nos va a servir para identificar a nuestro programa.

De las líneas 6 a la 15 se especifican las herramientas que utiliza *Fox-Pro*.

De la línea 16 a la 20 tenemos todas las variables globales.

De la línea 21 a la 23 tenemos los parámetros de control.

De la línea 24 a la 28 estamos dando un valor a nuestras variables iniciales.

En la línea 29 se activan los distintos archivos con sus diferentes alias e índices.

De la línea 31 a la 33 se definen ventanas.

De la línea 34 a la 41 se define el *Menú de Reportes*.

De la línea 48 a la 53 se incluye la selección del tipo de cifras de control.

De la línea 54 a la 59 se define el código para seleccionar el tipo de reporte.

De la línea 60 a la 69 se define el *Menú de Mantenimiento*.

De las líneas 70 a la 85 se define el *Menú Principal*.

Las líneas 86 a la 93 contienen menús para ayudas en la captura de claves de artículos.

Las líneas 94 a la 95 contienen menú para ayuda en la captura del tipo receptor.

Las líneas 96 a la 97 contienen el menú para ayuda de la captura de la dependencia.

## **4.2 Programación de módulos**

En esta parte del capítulo se da una descripción de cada uno de los módulos que integran el sistema, definiendo al principio el bloque de subrutinas, que es la parte más importante del sistema dado que en él se encuentran los procedimientos que utilizan todos los módulos del sistema.

### **4.2.1 Descripción de subrutinas**

Para facilitar el desarrollo de la programación, se decidió crear un archivo para almacenar las rutinas que se van a utilizar desde los diferentes módulos del sistema de Altas, Bajas y Cambios. Estas rutinas son de uso general y pueden accederse desde cualquiera de los programas del sistema o desde otras rutinas también contenidas en este archivo.

El archivo en cuestión se ha denominado *Rutinas*, y a continuación describiremos las acciones que efectúan cada una de las funciones y procedimientos, y en algunos casos, transcribiremos el código fuente, cuando la función o el procedimiento sea especialmente importante. Una parte de las funciones y procedimientos que contiene son los siguientes:

- **PROCEDURE: DESPLIEGA\_PANTALLA**

Llamado por:           BORR\_DAT (Procedure en RUTINAS.PRG)  
                          ALTAS.PRG

Descripción : Despliega en la pantalla las etiquetas de los datos a visualizar en altas, bajas o cambios.

- **PROCEDURE: DESPL\_CARAS**

Llamado por:           MENU.PRG  
                          ALTAS.PRG  
                          BAJAS.PRG  
                          CAMBIOS.PRG

Descripción: Dibuja la carátula de la pantalla de captura. Rellena la pantalla con el carácter utilizando la función REPLICATE y después dibuja marcos en la pantalla con las funciones PANEL y DOUBLE. Finalmente pone los encabezados de la pantalla y habilita el reloj.

- **PROCEDURE: RECUPERA\_DATOS**

Llamado por: DATOS\_ADICIONAL(PROCEDURE en RUTINAS.PRG)  
MODI\_DAT (PROCEDURE en RUTINAS.PRG)  
BORR\_DAT (PROCEDURE en RUTINAS.PRG)

Descripción: Mueve los campos del archivo maestro a variables de trabajo. El archivo maestro debe ser seleccionado previamente.

- **PROCEDURE: INICIA\_DATOS**

Llamado por: ALTAS.PRG  
BAJAS.PRG  
CAMBIOS.PRG

Descripción: Inicializa las variables que sirven como puente de captura de los datos del Archivo *Maestro*.

- **PROCEDURE: GUARDA\_DATOS**

Llamado por: DATOS\_ADICIONAL(PROCEDURE en RUTINAS)  
ALTAS.PRG

Descripción: Reemplaza los datos de un registro del Archivo *Maestro* con las variables de trabajo.

- **PROCEDURE: DESPL\_DATOS**

Llamado por: BORR\_DAT (PROCEDURE en RUTINAS.PRG)

Descripción: Despliega en pantalla el contenido de las variables de trabajo que sirven como puente en la captura.

- **PROCEDURE: CAPT\_LLAVE**

Llamado por: BAJAS.PRG y CAMBIOS.PRG  
Llama a: V\_MAYOR() (function en RUTINAS.PRG)  
V\_NIV1() (function en RUTINAS.PRG)  
V\_NIV2() (function en RUTINAS.PRG)  
V\_NIV3() (function en RUTINAS.PRG)

Descripción: Proporciona la mascarilla en pantalla para especificar las características de los artículos a dar de baja o a modificar. Llama a funciones de validación para ayudar en el llenado de los campos que definen la clave del artículo (*Mayor, Nivel\_1, Nivel\_2, Nivel\_3*) con la tecla F1.

- **PROCEDURE: DESP\_LLAVE**

Llamado por: MODI\_DAT (PROCEDURE en RUTINAS.PRG)

Descripción: Despliega en pantalla los datos no modificables de un registro seleccionado para cambios.

- **PROCEDURE: V\_GRUPO**

Llamado por: V\_NIV2() (function en RUTINAS.PRG)

Llama a: V\_NIV3() (function en RUTINAS.PRG)  
NIVEL\_AFECTACION (PROCEDURE en RUTINAS)  
ADI\_GRUPO (PROCEDURE en RUTINAS.PRG)

Descripción: Procedimiento que valida la clave del artículo. Debe existir en el catálogo y debe ser un nivel afectable, lo cual se checa con el procedimiento *Nivel\_Afectacion*. De no existir, se da la opción de adicionarlo al catálogo con el procedimiento *Adi\_Grupo*.

- **PROCEDURE: NIVEL\_AFECTACION**

Llamado por: V\_GRUPO (PROCEDURE en RUTINAS.PRG)

Llama a: DETERMINA\_NIVEL (PROCEDURE en RUTINAS.PRG)

Descripción: Procedimiento que determina si una clave de artículo puede ser afectable, es decir, si puede contener registros en el archivo *Maestro*.

- **PROCEDURE: DETERMINA\_NIVEL**

Llamado por: NIVEL\_AFECTACION (PROCEDURE en RUTINAS.PRG)  
CHECA\_NIVEL (PROCEDURE en RUTINAS.PRG)

Descripción: Procedimiento que determina el nivel del grupo (1, 2, 3 ó 4) para saber si es afectable o no.

- **PROCEDURE: CHECA\_NIVEL**

Llamado por: ADI\_GRUPO (PROCEDURE en RUTINAS.PRG)

Llama a: INS\_GRUPO (PROCEDURE en RUTINAS.PRG)  
DETERMINA\_NIVEL (PROCEDURE en RUTINAS.PRG)  
SIGUE (PROCEDURE en MENU.PRG)

Descripción: Procedimiento que checa si una clave de grupo puede ser dada de alta en el catálogo de grupos

- **Function: V\_REFER()**

Llamado por: CAPTURA\_DATOS (PROCEDURE en RUTINAS.PRG)

DATOS\_ADICIONAL (PROCEDURE en RUTINAS)  
MODI\_DAT (PROCEDURE en RUTINAS.PRG)  
ALTAS.PRG  
Llama a: ADI\_REFERENCIA (PROCEDURE en RUTINAS.PRG)



Descripción: Función que valida la existencia de una clave de oficina en el catálogo de referencias. De no existir, da la opción de añadirla al catálogo. Además, despliega la descripción de la oficina.

- **Function: V\_RECEP()**

Llamado por:           CAPTURA\_DATOS (PROCEDURE en RUTINAS.PRG)  
                          DATOS\_ADICIONAL (PROCEDURE en RUTINAS)  
                          MODI\_DAT (PROCEDURE en RUTINAS.PRG)  
                          ALTAS.PRG  
Llama a:                ADI\_RECEPTOR (PROCEDURE en RUTINAS)

Descripción: Función que valida la existencia en catálogo de la clave de tipo de receptor para el artículo que se esté capturando o modificando. Opcionalmente se permite darlo de alta en el catálogo.

- **Function: V\_NUMSE()**

Llamado por:           CAPTURA\_DATOS (PROCEDURE en RUTINAS.PRG)  
                          DATOS\_ADICIONAL (PROCEDURE en RUTINAS)  
                          MODI\_DAT (PROCEDURE en RUTINAS.PRG)

Descripción: Función booleana que determina si el artículo que se está capturando o modificando debe llevar número de serie, esto en base al campo *OPC1* del catálogo de grupos.

- **Function: V\_PERTE()**

Llamado por:           CAPTURA\_DATOS (PROCEDURE in RUTINAS.PRG)  
                          DATOS\_ADICIONAL (PROCEDURE in RUTINAS.PRG)  
                          MODI\_DAT (PROCEDURE in RUTINAS.PRG)

Descripción: Función booleana que determina si el artículo que se está capturando o modificando debe llevar número de serie del artículo al que pertenece, esto en base al campo *OPC3* del catálogo de grupos.

Si desea obtener más información de los procedimientos y rutinas con que cuenta el sistema, puede consultar el "Apéndice B1" al final de la tesis.

#### 4.2.2 ALTAS

La adición de nuevos registros del archivo *Maestro* se da al seleccionar la opción **Altas** en el *Menú Principal*.

El módulo de Altas consiste de las siguientes opciones:

1. Altas de registros individuales

Las altas de registros individuales se dan al llenar los datos de la mascarilla que presenta la **figura 4.2**.

Los campos que deben ser llenados son los siguientes *año, mes, grupo, nivel1, nivel2, nivel3, num. de serie, pertenece a, referencia, tipo de receptor, importe*.

## 2. Altas de registros por factura

Si la selección de altas es por factura , la pantalla de captura es la mostrada en la **figura 4.3**.

Al igual que en la pantalla de registros individuales, aquí se especifican los datos de los registros, sólo que esos datos se aplican al número de altas en el archivo *Maestro*. El importe se divide entre el número de artículos que se dan de alta, y este resultado es el que se asigna a cada registro individual generado.

Módulo de Mantenimiento y Captura

SISTEMA DE MOBILIARIO Y EQUIPO  
MODULO DE MANTENIMIENTO Y CAPTURA

FECHA CONTROL 7/95  
ALTAS DE REGISTROS

OPERACION DE SISTEMA.- D.C.  
Año: Mes:  
DEFINITIVO   
Clave: Grupo Nivel\_1 Nivel\_2 Nivel\_3  
RONICO/EQUIPO DE MICROCOMPUTACION/UNIDAD BASE DE PROCESAMIENTO/OLIVETTI 486-60  
Num. Serie:  
Pertenece a:  
Referencia:  
Tipo Recep:  
Importe:

SON CORRECTOS TODOS LOS DATOS?  SI NO

Módulo de Mantenimiento y Captura Ver Nuevo Cerrar

**Figura 4.2.-Pantalla de Altas de Registros Individuales.**

Microsoft Windows

SISTEMA DE MOBILIARIO Y EQUIPO  
MODULO DE MANTENIMIENTO Y CAPTURA

FECHA CONTROL: 7/98  
ALTAS DE REGISTROS

OPERACION DE SISTEMAS.- D.C.

Clave:	Grupo	Nivel_1	Nivel_2	Nivel_3
RONICO/EQUIPO DE MICROCOMPUTACION/UNIDAD BASE DE PROCESAMIENTO/OLIUETTI M480-20				
Num articulos				
Referencia				
Tipo Recep				
Importe				

\*\*\*\*\* SON CORRECTOS TODOS LOS DATOS?  SI  NO \*\*\*\*\*

\*\*\*\*\*

Figura 4.3 Pantalla de Altas de Registros por Factura.

Una vez que se han generado todas las altas, se tiene la posibilidad de asignar *número de serie*, *pertenece a*, *referencia*, *tipo de receptor* e *importe* de cada uno de los registros individuales. A continuación se muestra el código fuente correspondiente a la pantalla de **Altas**, posteriormente se da una explicación del mismo.

1. \* Programa: ALTAS.PRG
2. \* Nota....: Programa de captura para mobiliario y equipo.
3. \*
4. \*
5. \*
6. DEACTIVATE MENU MENPRIN
7. DEFINE POPUP pop\_altas FROM 10,10 TO 15,50
8. DEFINE BAR 1 OF pop\_altas PROMPT "Capturar registros individuales"
9. DEFINE BAR 2 OF pop\_altas PROMPT "Capturar registros por factura"
10. ON SELECTION POPUP pop\_altas DO tpopup\_altas WITH BAR()
11. ACTIVATE POPUP pop\_altas
12. CLEAR
13. DO INICIA\_DATOS
14. DO DESPL\_CARA
15. DO DESPLIEGA\_PANTALLA
16. wErrGru = .T.
17. ON KEY LABEL F1 do act\_pops
18. DO WHILE (LASTKEY() <> 27)
19. DO WHILE wErrGru .AND. (LASTKEY() <> 27) && Hasta <ESC>
20. DO CAPTURA\_DATOS
21. IF (LASTKEY() <> 27) && No cancelado
22. DO v\_grupo WITH mMayor, mNivel\_1, mNivel\_2, mNivel\_3
23. IF werrgru THEN

```

24. ACTIVATE WINDOW win_msg
25. @ 2,10 SAY " ERROR EN EL GRUPO"
26. @ 3,10 SAY "PRESIONE UNA TECLA PARA CONTINUAR"
27. STORE INKEY(0) TO wtecla
28. DEACTIVATE WINDOW win_msg
29. ENDIF
30. IF .NOT. v_reter(mReferencia) THEN
31. wErrGru = .T.
32. ACTIVATE WINDOW win_msg
33. @ 2,10 SAY " ERROR EN LA REFERENCIA"
34. @ 3,10 SAY "PRESIONE UNA TECLA PARA CONTINUAR"
35. STORE INKEY(0) TO wtecla
36. DEACTIVATE WINDOW win_msg
37. ENDIF
38. IF .NOT. v_recep(mTipo_receptor) THEN
39. wErrGru = .T.
40. ACTIVATE WINDOW win_msg
41. @ 2,10 SAY " ERROR EN TIPO DE RECEPTOR"
42. @ 3,10 SAY "PRESIONE UNA TECLA PARA CONTINUAR"
43. STORE INKEY(0) TO wtecla
44. DEACTIVATE WINDOW win_msg
45. ENDIF
46. ENDIF
47. ENDDO
48. IF LASTKEY() <> 27 && NO CANCELADO
49. @ 23,10 SAY "SON CORRECTOS TODOS LOS DATOS?"
50. @ 23,50 PROMPT "SI" MESSAGE SPACE(30) + "GRABAR EL REGISTRO"
51. @ 23,54 PROMPT "NO" MESSAGE SPACE(30) + "MODIFICAR EL REGISTRO"
52. MENU TO mprompt
53. DO CASE
54. CASE mprompt = 1
55. IF windividual
56. APPEND BLANK
57. DO GUARDA_DATOS
58. wMovimiento = "**** ALTA ****"
59. DO IMPR_BITACORA
60. SELECT ALTAS
61. APPEND BLANK
62. DO GUARDA_DATOS
63. SELECT SESION
64. APPEND BLANK
65. REPLACE IMPORTE WITH MOBYEQUI->IMPORTE
66. REPLACE NUMREG WITH RECNO('MOBYEQUI')
67. SELECT MOBYEQUI
68. CLEAR GETS
69. ELSE
70. DO DATOS_ADICIONALES
71. ENDIF
72. DO INICIA_DATOS
73. DO DESPL_CARA
74. DO DESPLIEGA_PANTALLA
75. CASE mprompt = 2
76. ENDCASE
77. @ 23,10 SAY SPACE(70)

```

```
78. wErrGru = .T.  
79. ENDIF  
80. ENDDO  
81. ON KEY LABEL F1 DO SIGUE  
82. RETURN  
83. *  
84. * EOF: ALTAS.PRG  
85. □
```

## Documentación del código fuente del programa altas

El programa fuente de **Altas** inicia con algunos comentarios referentes al nombre del programa y notas, estos comentarios se declaran de la línea 1 a la 4, esto nos va a servir para identificar a nuestro programa.

En la línea 6 se activa el *Menú Principal*, que se definió anteriormente con las opciones de Altas, Bajas, Cambios, Mantenimiento, Reportes y Respaldos.

De las líneas 7 a la 10 se define un menú de opciones y pregunta si las altas es por captura de registros individuales ó captura de registros por factura.

En la línea 13 se tiene el procedimiento *Inicia \_Datos*, éste nos sirve para restaurar los campos de trabajo.

En la línea 14 se define el procedimiento *Despliega\_Cara*, éste despliega una parte de la pantalla con el mensaje " Sistema de Mobiliario y Equipo, Módulo de Mantenimiento y Captura".

En la Línea 15 se define el procedimiento *Despliega\_Pantalla*, el cual nos va a mostrar mensajes como el *año, mes, día, grupo, nivel\_1, nivel\_2, nivel\_3*.

De las líneas 19 a la 27 va el procedimiento de captura de datos, si no es cancelado con ESC se va al procedimiento *V\_Grupo*, donde si se encontró el grupo, ve el nivel de afectación, si es falso el nivel entonces manda un mensaje "La Afectación del Grupo no es Válida" y si es verdad entonces lo adiciona al catálogo de grupo.

De las líneas 23 a la 46 se especifica si hay un error de grupo entonces manda un mensaje "Error en el Grupo", si hay un error en la referencia entonces aparece el siguiente mensaje "Error en la Referencia", si hay un error en el receptor aparece el siguiente mensaje "Error en el Tipo de Receptor".

De la línea 48 a la 51 se pregunta si los datos son correctos, si están bien entonces aparece el siguiente mensaje "Grabar el Registro" y con esto se va al procedimiento *Guarda\_Datos* e imprime Bitácora y si los datos no son correctos entonces mandará el siguiente mensaje "Modificar el Registro".

De las líneas 52 en adelante se ejecutan algunas rutinas, las cuales se describen con más detalle en el Apéndice "B1".

#### 4.2.3 BAJAS

El módulo de bajas tiene como principal función la eliminación de activos fijos de nuestro catálogo, ya sea uno por uno o por factura, en cualquiera de los dos casos la información es borrada del archivo principal y actualizada en un archivo histórico de bajas, el cual lleva un registro de estas operaciones, además, permite recuperar datos que no deberían ser eliminados.

Analizando las dos opciones que maneja este módulo, en el proceso de eliminación registro por registro, el usuario especifica todos los datos de un artículo en particular y el sistema lo busca, en caso de encontrarlo pregunta si estamos seguros de la eliminación, si afirmamos, este artículo es borrado. En el caso de eliminación por factura, nosotros podemos seleccionar un grupo de artículos con ciertas características, por ejemplo: todos los activos de cierto grupo que tengan una cierta referencia, entonces el sistema analiza y busca todos los datos que tengan esa característica y pregunta si los elimina, si afirmamos, éstos son borrados.

Al seleccionar del *Menú Principal* la opción de **Bajas**, el sistema nos mostrará la pantalla que se ejemplifica en la **figura 4.4.**, la cual nos muestra las dos opciones que maneja el sistema: registros individuales o por factura.

Una vez seleccionada cualquiera de las dos opciones, inmediatamente después aparecerá en pantalla los datos solicitados por el sistema que pueden verse en la **figura 4.5.**, estos datos aparecerán siempre independientemente de la opción que se elija, la diferencia radicará en la forma en que el sistema interpreta éstos.

SYSTEMA DE MANTENIMIENTO Y EQUIPOS  
MODULO DE MANTENIMIENTO Y CAPTURA

FECHA CONTROL: 9709  
BAJAS DE REGISTROS

ANO: Mes:

Clave: Grupo Nivel\_1 Nivel\_2 Nivel\_3

Man. Serie

Particula

Referencia

Sector

Tipo Recep

Importe

..... DIGITE LA CLAVE DEL REGISTRO .....

Módulo: Recent: 27713/27713 Record: Locked Line: Num: Cap:

Figura 4.4. Pantalla con las 2 opciones del módulo de bajas



```

20. DO CAPT_LLAVE
21. IF LASTKEY() <> 27
22. wErrGru = .F.
23. DO CREA_FILTRO
24. ENDIF
25. ENDDO
26. IF LASTKEY() <> 27 .AND. .NOT. wErrGru  && NO CANCELADO
27. DO BORR_DAT                && Borrar registros
28. KEYBOARD CHR(7)           && Restablece <ESC>
29. DO INICIA_DATOS
30. DO DESPL_CARA
31. @ 3,52 SAY "BAJAS DE REGISTROS "
32. mkey = inkey()
33. ELSE
34. wbajas = .F.
35. ENDIF
36. ENDDO
37. ON KEY LABEL FI DO SIGUE
38. RETURN
39. EOF: BAJAS.PRG

```

Después de observar el listado del programa, ahora daremos una breve explicación de las instrucciones y rutinas que realiza éste:

- Las primeras cinco líneas definen los parámetros generales de pantalla y definen variables privadas del módulo.
- De la línea 6 a la 9 se encuentra la definición de las dos opciones del *Menú de Bajas*.
- En la línea 10 se activa un procedimiento que tiene como propósito fundamental la inicialización de todas las variables que serán leídas en pantalla y manejadas internamente por el proceso.
- La línea 11 activa una pantalla que despliega la información general del sistema, así mismo las líneas 12 a 14 son atributos de pantalla.
- De la línea 15 en adelante se ejecutan algunas rutinas, estas se describen con más detalle en el Apéndice "B1".

#### 4.2.4 CAMBIOS

La opción de cambios abre una pantalla que es una carátula para especificar datos comunes a un grupo de registros a modificar, ésta se puede ver en la *figura 4.6*.



Microsoft Windows

SISTEMA DE MOBILIARIO Y EQUIPO  
MODULO DE MANTENIMIENTO Y CAPTURA

FECHA CONTROL 7/95  
CAMBIOS A REGISTROS

ARU Mes:

Clave: Grupo Nivel\_1 Nivel\_2 Nivel\_3

EQUIPO DE COMPUTO ELECTRONICO/EQUIPO DE MICROCOMPUTACION/UNIDAD BASE DE PROCESA

Min. Serie

Pertenece a

Referencia

Sector

Tipo Recop

Importe

..... DIGITE LA CLAVE DEL REGISTRO .....

Figura 4.6. Pantalla para especificar datos comunes a un grupo de registros a modificar

Una vez que se han especificado estos datos, el sistema selecciona todos los registros que cumplen con el criterio especificado y se muestran en la **figura 4.7.**

Microsoft Windows

FECHA CONTROL 7/95  
CAMBIOS A REGISTROS

Clave	Grupo	Nivel_1	Nivel_2	Nivel_3	Importe
06	7	7113579	R21	1	2109.35
06	7	7113615	R21	1	2060.15
06	7	7113620	R21	1	2060.15
06	7	7113631	R21	1	2060.15
06	11	711388	F03	1	2036.88
06	10	7114104	R21	1	2109.35
07	3	7113541	R24	1	2152.95
07	3	7113566	R24	1	2152.95
07	3	7113607	R24	1	2152.95
07	3	7113621	R24	1	2152.95
07	3	7113630	R24	1	2152.95
07	3	711364	R24	1	2152.95
07	3	7113641	R24	1	2152.95
07	3	7113642	R24	1	2152.95
07	3	7113645	R24	1	2152.95
07	3	7113651	R24	1	2152.95

BASE DE PROCESAMIENTO

.....

Figura 4.7. Pantalla con todos los registros donde pueden modificarse

La ventana puede ser ampliada o disminuida con las teclas <control> <F10>. Los registros a modificar pueden seleccionarse con las teclas de dirección (<← ↑ ↓ →>) e <enter> y en ese caso, aparece otra pantalla (ver **figura 4.8.**) con todos los datos del registro donde pueden modificarse los siguientes datos:

- Núm. de serie
- Pertenece a
- Referencia
- Tipo de receptor
- Importe

Microsoft FoxPro

SISTEMA DE MOBILIARIO Y EQUIPO  
MODULO DE MANTENIMIENTO Y CAPTURA

FECHA CONTROL: 7/95  
CAMBIOS A REGISTROS

Año: 92    Mes: 5  
 Clave:    Grupo:    Nivel\_1:    Nivel\_2:    Nivel\_3:  
 13        2        1        6  
 EQUIPO DE COMPUTO ELECTRONICO - EQUIPO DE MICROCOMPUTACION - UNIDAD BASE DE PRO  
 Num. Serie: 0047397  
 Pertenece a:  
 Referencia: RSE  
 Sector: R  
 Tipo Recep:  
 Importe: 50000.00

-----

-----

Nombre    Puesto    Numero    Fecha    Referencia    Linea    Num    Carga

**Figura 4.8. Pantalla con todos los datos del registro donde pueden modificarse**

A continuación se muestra el código fuente del programa de cambios.

- \* Programa: CAMBIOS.PRG
- \* Nota....: Programa de cambios para mobiliario y equipo.
- 1.PRIVATE mkey
- 2.STORE " " TO mkey
- 3.CLEAR
- 4.DEACTIVATE MENU MENPRIN
- 5.DO INICIA\_DATOS
- 6.DO DESPL\_CARA
- 7.SET COLOR TO 6+
- 8.@ 3,52 SAY "CAMBIOS A REGISTROS"
- 9.SET COLOR TO
- 10.ON KEY LABEL F1 do act\_pops
- 11.STORE .T. TO wCambios
- 12.DO WHILE wcambios && Hasta <ESC>
- 13. werrgru = .T.
- 14. DO WHILE werrgru .AND. (LASTKEY() <> 27) && Mientras error grupo
- 15. DO CAPT\_LLAVE
- 16. IF (LASTKEY() <> 27)
- 17. wErrGru = .F.
- 18. DO CREA\_FILTRO
- 19. ENDIF
- 20. ENDDO
- 21. IF LASTKEY() <> 27 .AND. NOT. wErrGru && NO CANCELADO

```

22. DO MODI_DAT
23. KEYBOARD CHR(7) && Restablece <ESC>
24. DO INICIA_DATOS
25. DO DESPL_CARAS
26. @3,52 SAY "CAMBIOS A REGISTROS"
27. mkey = inkey()
28. ELSE
29. wcambios = .F.
30. ENDIF
31. ENDDO
32. ON KEY LABEL F1 DO SIGUE
33. RETURN
* EOF: CAMBIOS.PRG

```

### Documentación del código fuente del programa cambios

El programa fuente de cambios inicia con algunos comentarios referentes al nombre del programa, fecha de realización y una nota. Lo anterior es con el fin de dar claridad a la programación.

De la línea 1 a la 4 se declara una variable privada, a la cual se le almacena un espacio en blanco. Se limpia la pantalla y a su vez se activa la ventana del *Menú Principal*.

En la línea 5 se define el procedimiento *Inicia\_Datos*, este procedimiento nos sirve para restaurar los campos de trabajo.

En la línea 6 y 7 se define el procedimiento *Despliega\_Cara*, este proceso despliega una parte de la pantalla con el mensaje "Cambios a Registros".

La línea 8 activa la tecla **F1** y a su vez ejecuta el procedimiento *Act\_Pops*.

La línea 9 almacena un valor booleano a la variable *wCambios*.

En la línea 10 se ejecuta el procedimiento mientras sea diferente de **<esc>**.

En la línea 11 la variable booleana *werrgru* tiene asignado el valor de "true".

En la línea 12 se ejecuta un procedimiento mientras la variable *werrgru* sea verdadera y la última tecla oprimida sea diferente de **<esc>**.

En la línea 13 se ejecuta el procedimiento de *Capt\_LLave*.

De la línea 14 a la 17 se ejecuta una estructura de control condicional.

La línea 18 finaliza todo el proceso comprendido desde la líneas 8 a la 17.

De la línea 19 a la 29. Si no se oprime la tecla <esc> se va al procedimiento modifica datos *Mod\_Dat*. Si se oprime <esc> se ejecutan los procedimientos *Inicia\_Datos* y *Des\_Cara* hasta que la variable *wCambios* sea igual a falso, finalizando el proceso de cambios.

La línea 30 y 31 verifica si sigue activada la tecla F1, si es así ejecuta un procedimiento que no realizada nada, y la línea 31 retorna al *Menú Principal*.

#### 4.2.5 MANTENIMIENTO

La opción de mantenimiento del *Menú Principal* permite la actualización de los catálogos utilizados en el Sistema de Control y Actualización del Activo Fijo. La **figura 4.9** muestra la pantalla de mantenimiento con las diferentes opciones de mantenimiento que se permiten realizar en el Módulo de Mantenimiento y Captura.

El Menú de Mantenimiento cuenta con cuatro opciones de mantenimiento al sistema que a continuación se describen:

#### Catálogos de Grupos

En este catálogo se encuentran registrados los diferentes *tipos* de artículos con los que cuenta la institución, clasificados por su naturaleza contable. Una vez seleccionada esta opción del Menú de Mantenimiento a Catálogos, aparecerá una ventana con el catálogo de grupos que se muestra en la **figura 4.10**.

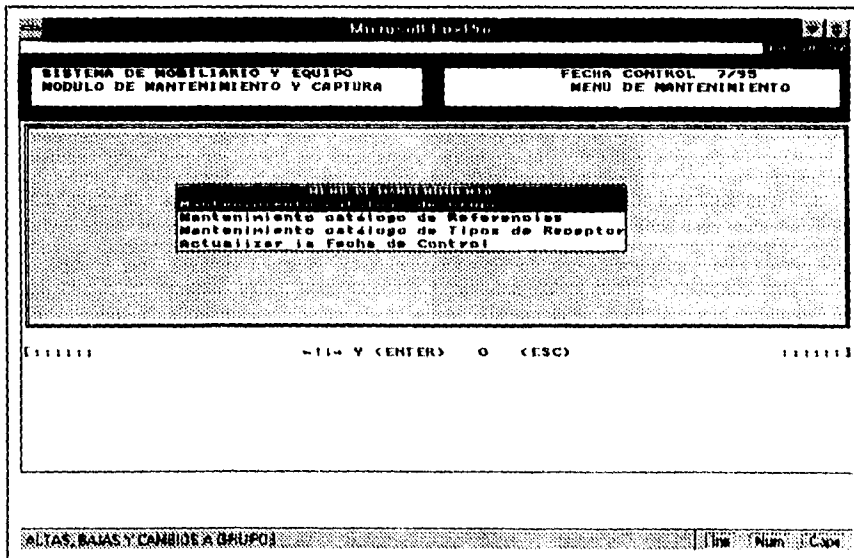


Figura 4.9 Pantalla de mantenimiento del menú principal.

El mantenimiento al catálogo se da con las siguientes acciones:

**Altas.** La tecla <insert> proporciona una ventana para adicionar nuevos registros al Catálogo de Grupos. Los campos a digitar son: clave del grupo, descripción del grupo, indicador de que requiere número de serie, indicador de corte de control de texto y el indicador de requiere pertenece a. Los posibles valores de los indicadores son F - Falso y T - Verdadero. Se hace una validación sobre los siguientes casos:

- El nuevo grupo a ser adicionado no debe existir previamente.
- Si se trata de un nivel analítico, los niveles superiores no deben tener artículos de mobiliario y equipo previamente registrados.
- Si se trata de un nivel analítico, los niveles superiores deben existir.

**Bajas.** La tecla <supr> permite un registro en particular del catálogo de grupos, seleccionándolo previamente con las teclas de dirección. Una vez que se ha borrado un registro, aparecerá con un pequeño rombo en su lado izquierdo. Se hace la validación en los siguientes casos:

- El grupo a borrar no debe tener niveles dependientes.
- El grupo a borrar no debe tener artículos registrados previamente.

GRUPO	Mayor	Nivel_1	Nivel_2	Nivel_3	DESCRIPC
013	02	01	19	T F F	OLIVETTI M488-20
013	02	01	20	T F F	OLIVETTI PCS 86
013	02	01	21	T F F	OLIVETTI PRO 3M20
013	02	01	22	T	UTS 38W DISPLAY 3575/99
013	02	01	24	T	CPU 2462
013	02	01	25	T	CPU 2807
13	2	1	28	T F F	OLIVETTI PCS 206
13	2	1	29	T F F	OLIVETTI PRO 3M20
13	2	1	30	T F F	OLIVETTI 486 -BU486
13	2	1	31	T F F	MICRON
13	2	1	32	T F F	BPM
13	2	1	33	T F F	HEULETT-PACKARD
13	2	1	34	T F F	NEC
13	2	1	35	T F F	TOSHIBA
13	2	1	36	T F F	UNISYS
13	2	1	37	T F F	UVSE

Figura 4.10. Ejemplo de pantalla de mantenimiento a catálogos.

**Cambios.** Al posicionarse sobre un registro con las teclas de dirección y pulsar la tecla <enter> se abre una ventana en la que es posible modificar la descripción del grupo, así como los indicadores de requiere número de serie, corte de control de texto y pertenece a otro artículo.

También puede accesarse al catálogo de grupos desde la pantalla de captura de mobiliario a través de la tecla <F9>. El Catálogo se muestra en una ventana.

Para buscar la clave de un artículo en particular deben teclarse simultáneamente las teclas <control> <F9>. En este momento aparecerá un recuadro en el que se puede digitar la descripción del artículo a buscar ó una parte de ella en el campo **look for**.

Si se quiere buscar por ejemplo, la clave de una silla, se debe digitar la palabra silla. Una vez digitada la descripción del artículo a buscar, desplácese con las teclas de dirección hasta el botón **find** y oprima la tecla <enter>, en ese momento se empezará a hacer la búsqueda del artículo.

Si no se ha encontrado la descripción especificada, aparecerá un mensaje con una leyenda **not found**. Si la descripción especificada se ha encontrado, el cursor se posicionará en ese registro. Basta con desplazarse hacia la derecha o izquierda hasta encontrar la clave del grupo correspondiente. También existe la posibilidad de buscar nuevamente, por si existen otros grupos con una descripción similar, esto se hace con las teclas < control > < G >. Para salir del catálogo de grupos se pulsa la tecla < esc >.

### **Catálogo de referencias**

En este catálogo se encuentran registradas todas las Dependencias Administrativas del Banco. Una vez seleccionada esta opción del menú de Mantenimiento a Catálogos, aparecerá una ventana con el catálogo de referencias. El mantenimiento al catálogo se da con las siguientes acciones:

**Altas.** La tecla <insert> proporciona una ventana para adicionar nuevos registros al Catálogo de Referencias. Los datos a digitar son: clave de referencia, nombre de referencia y sector. Los posibles valores del campo Sector son:

1. Oficina Central
2. Fábrica de Billetes
3. Sucursales

La nueva referencia a ser adicionada no debe existir previamente.

**Bajas.** La tecla <supr> permite borrar un registro en particular del catálogo de referencias, seleccionándolo previamente con las teclas de dirección. Una vez que se borra un registro, aparecerá con un pequeño rombo en el lado izquierdo. Se hace la validación en el siguiente caso:

- La referencia no debe tener artículos registrados previamente.

**Cambios.** Al posicionarse sobre un registro con las teclas de dirección y pulsar la tecla <enter> se abre una ventana, en la que es posible modificar el nombre de referencia, así como el sector al que pertenece.

La tecla <F10> permite el acceso al catálogo de referencias desde la pantalla de captura y permite la búsqueda de descripción con las teclas <control> <F> y <control> <G>.

### **Catálogo de tipos de receptor**

En este catálogo se encuentran registrados todos los tipos de asignación que puede tener un artículo. Una vez seleccionada esta opción del Menú de Mantenimiento a Catálogos, aparecerá una ventana con el catálogo de tipos de receptor. El mantenimiento al catálogo se da con las siguientes acciones:

**Altas.** La tecla <insert> proporciona una ventana para adicionar nuevos registros al catálogo de tipo de receptor. Los cambios a digitar son: la clave del tipo de receptor y la descripción del tipo de receptor. El nuevo tipo de receptor a ser adicionado no debe existir previamente.

**Bajas.** La tecla <supr> permite borrar un registro en particular del catálogo de tipos de receptor, seleccionándolo previamente con las teclas de dirección. Una vez que se ha borrado un registro, aparecerá con un pequeño rombo a su izquierda.

**Cambios.** Al posicionarse sobre un registro con las teclas de dirección y pulsar la tecla <enter>, se abre una ventana en la que es posible modificar el nombre del tipo de receptor.

### **Fecha de control**

Esta opción del Menú de Mantenimiento permite cambiar la fecha de control de registro de artículos de Mobiliario y Equipo. Es conveniente hacer el cambio de fecha, una vez que se ha concluido la captura de un mes y se han generado los reportes de Altas, Bajas y Cambios. Estos reportes son selectivos por mes de registro y son útiles para determinar cifras de control.



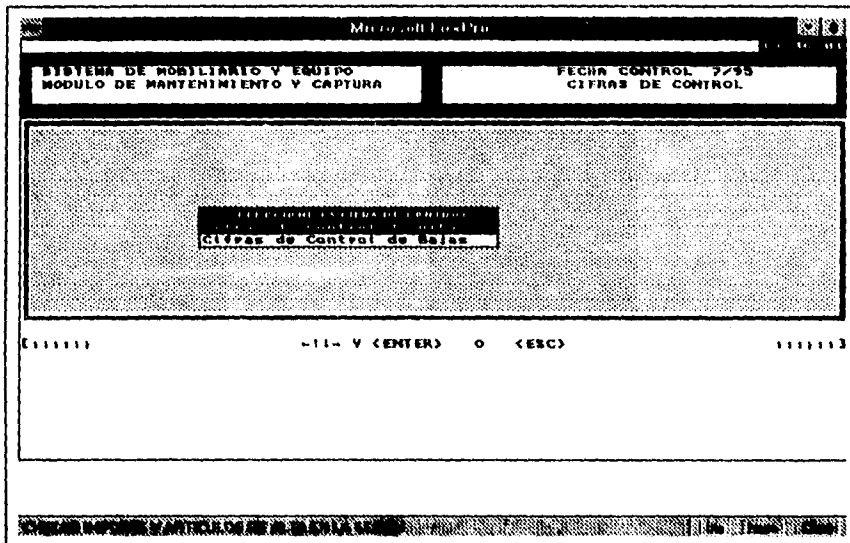


Al final del capítulo se muestran algunos reportes representativos generados por el sistema.

#### 4.2.7 CIFRAS DE CONTROL

En el *Módulo de Cifras de Control* se hacen las validaciones de las cifras de control que se manejan en el sistema.

La **figura 4.12** muestra la pantalla del módulo de cifras de control.



**Figura 4.12.** Pantalla de cifras de control.

El módulo de cifras de control contiene dos opciones que son las siguientes:

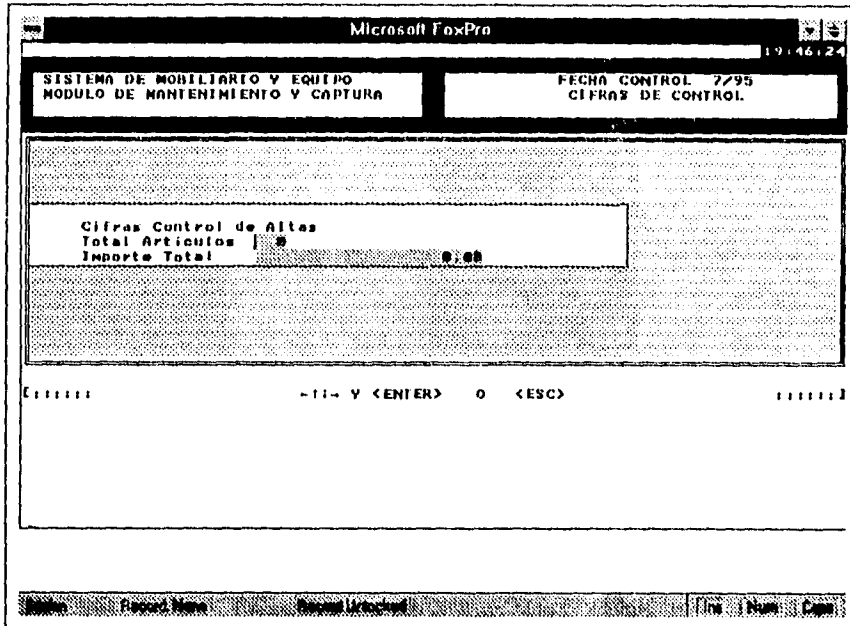
1.- Cifras de control de altas.- Valida las cifras de control de altas

Consiste de dos cifras:

Número de artículos.- El total de artículos adicionados en una sesión de trabajo.

Importe.- El Importe total de los artículos adicionados en una sesión de trabajo.

En la **figura 4.13**, se muestra la pantalla de Cifras de Control de Altas.



**Figura 4.13. Pantalla de cifras de control de altas.**

A continuación se muestra el código del programa de cifras de control de altas.

```

PRIVATE mImpSesi,mArtSesi,mImpCtrl,mArtCtrl
STORE 0 TO mImpSesi,mArtSesi,mImpCtrl,mArtCtrl
STORE .T. TO mErrCtrl
ON ESCAPE DO SIGUE
CLOSE DATA
USE SESION
SET DELETED ON
GO TOP
SUM IMPORTE TO mImpSesi
COUNT ALL TO mArtSesi
ACTIVATE WINDOW Win_Msg
DO WHILE (mErrCtrl .AND. LASTKEY() <> 27)
@ 1,5 SAY "Cifras Control de Altas"
@ 2,5 SAY "Total Articulos " GET mArtCtrl PICTURE "999"
@ 3,5 SAY "Importe Total " GET mImpCtrl PICTURE "999,999,999,999,99"
READ
mErrCtrl = .F.
IF (mArtCtrl <> mArtSesi)
WAIT "ERROR EN NUMERO DE ARTICULOS" WINDOW
mErrCtrl = .T.
ENDIF
IF (mImpCtrl <> mImpSesi)
WAIT "ERROR EN IMPORTE" WINDOW
mErrCtrl = .T.

```

```

ENDIF
IF LASTKEY() = 27
mErrCtrl = F.
ENDIF
ENDDO
WAIT "CIFRAS CORRECTAS" WINDOW
SET DELETED OFF
CLOSE DATA
DEACTIVATE WINDOW Win_Msg
DO ABRE
ON ESCAPE
RETURN

```

En el código se observa que se trae un acumulador para el número de artículos y el importe, en el momento en que se capturan las cifras de control se comparan con los acumuladores y si no son iguales se despliega un mensaje de error, si son iguales se despliega un mensaje de cifras correctas.

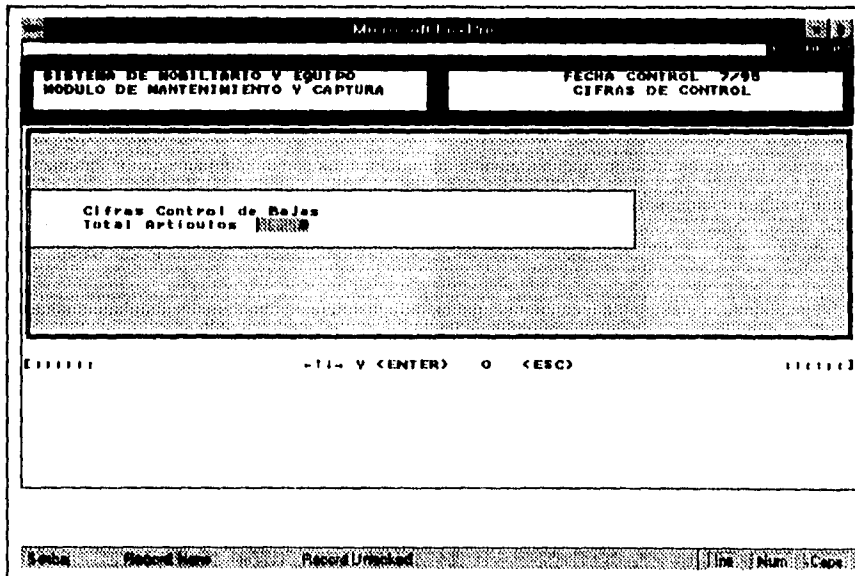
2.- Cifras de control de bajas.- Valida las cifras de control de bajas

Cifras de control de bajas

Número de artículos.- Total de artículos dados de baja en una sesión de trabajo.

Importe.- El importe de los artículos dados de baja en una sesión de trabajo.

En la **figura 4.14.** se muestra la pantalla de Cifras de control de bajas.



**Figura 4.14.** Pantalla de cifras de control de bajas.

Las cifras de control deben darse al finalizar la sesión de trabajo.

A continuación se muestra el código del programa de cifras de control de bajas .

```
PRIVATE mImpSesi,mArtSesi,mImpCtrl,mArtCtrl
STORE 0 TO mImpSesi,mArtSesi,mImpCtrl,mArtCtrl
STORE .T. TO mErrCtrl
ON ESCAPE DO SIGUE
CLOSE DATA
USE SESBAJ
SET DELETED ON
GO TOP
SUM IMPORTE TO mImpSesi
COUNT ALL TO mArtSesi
ACTIVATE WINDOW Win_Msg
DO WHILE (mErrCtrl .AND. LASTKEY() <> 27)
@ 1,5 SAY "Cifras Control de Bajas"
@ 2,5 SAY "Total Articulos " GET mArtCtrl PICTURE "999"
@ 3,5 SAY "Importe Total " GET mImpCtrl PICTURE "999,999,999,999.99"
READ
mErrCtrl = .F.
IF (mArtCtrl <> mArtSesi)
WAIT "ERROR EN NUMERO DE ARTICULOS" WINDOW
mEnCtrl = .T.
ENDIF
IF (mImpCtrl <> mImpSesi)
WAIT "ERROR EN IMPORTE" WINDOW
mErrCtrl = .T.
ENDIF
IF LASTKEY() = 27
mErrCtrl = .F.
ENDIF
ENDDO
WAIT "CIFRAS CORRECTAS" WINDOW
SET DELETED OFF
CLOSE DATA
DEACTIVATE WINDOW Win_Msg
RETURN
```

En el código se observan los acumuladores de importe y de número de artículos que se compararán con el valor capturado en la pantalla, si éstos son iguales se despliega un mensaje de cifras correctas, en caso de ser diferentes se despliega un mensaje de error.

#### 4.3 Reportes generados por el sistema

A continuación se muestran algunos formatos de los reportes más representativos que genera el sistema, estos son ejemplos, si desea ver con más detalle, consulte el Apéndice "C1" donde se muestran ejemplos de las corridas de todos los reportes.

- Ejemplo del formato de impresión de todo el Catálogo de Grupos del sistema:

**B A N C O X X X X X X X X X X**  
**SISTEMA DE CONSULTA Y EXPLOTACION DEL MOBILIARIO Y EQUIPO**  
**CATALOGO DE GRUPOS**  
**FECHA: 25/SEP./1995** **HOJA: 4**

MAYOR	NIVEL 1	NIVEL 2	NIVEL 3	DESCRIPCION	NUM.SERIE	PERTENECE A:
---	---	---	---	-----	-----	-----
---	---	---	---	-----	-----	-----

- Ejemplo del formato de impresión del Catálogo de Referencias del sistema:

**B A N C O X X X X X X X X X X**  
**SISTEMA DE CONSULTA Y EXPLOTACION DEL MOBILIARIO Y EQUIPO**  
**CATALOGO DE REFERENCIAS**  
**FECHA: 25/SEP./1995** **HOJA: 4**

REFER.	DESCRIPCION	TITULAR	UBICACIÓN
---	-----	-----	-----
---	-----	-----	-----

- Ejemplo del formato de impresión de Bajas Selectivas por Mes del sistema:

**B A N C O X X X X X X X X X X**  
**SISTEMA DE CONTROL FISICO DE BIENES MUEBLES**  
**DIRECCION DE CONTRALORIA**  
**REPORTE DE BAJAS DE MOBILIARIO Y EQUIPO**  
**CONTABILIDAD DE CAPITAL Y FIDEICOMISOS**  
**DEL MES DE ENERO DE 1995**  
**FECHA: 26/SEP./1995** **HOJA: 4**

FECHA DE ADQUIS.	DESCRIPCION DEL ARTICULO	REF	NUM. DE SERIE	IMPORTE	SUMA POR MES	SUMA POR AÑO
-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----

- Ejemplo del formato de impresión de Cambios Selectivos por Mes del sistema:

**B A N C O X X X X X X X X X X**  
**SISTEMA DE CONTROL FISICO DE BIENES MUEBLES**  
**DIRECCION DE CONTRALORIA**  
**CONTABILIDAD DE CAPITAL Y FIDEICOMISOS**  
**REPORTE DE CAMBIOS DE MOBILIARIO Y EQUIPO**  
**DEL MES DE ENERO DE 1995**

**FECHA: 25/SEP./1995**

**HOJA: 4**

FECHA DE ADQUIS.	CONCEPTO	NUM. DE SERIE	REF	IMPORTE
-----	-----	-----	-----	-----
-----	-----	-----	-----	-----

- Ejemplo del formato de impresión de Traspasos Selectivos por Mes del sistema:

**B A N C O X X X X X X X X X X**  
**SISTEMA DE CONTROL FISICO DE BIENES MUEBLES**  
**DIRECCION DE CONTRALORIA**  
**REPORTE DE TRASPASOS DE MOBILIARIO Y EQUIPO**  
**CONTABILIDAD DE CAPITAL Y FIDEICOMISOS**  
**DISTRIBUCION DE BIENES REGISTRADOS EN EL MES DE ENERO DE 1995**  
**FECHA: 25/SEP./1995**

**HOJA: 4**

FECHA DE ADQUIS.	DEPENDENCIA DISTRIBUIDORA	DESCRIPCION DEL BIEN	NUM. DE SERIE	IMPORTE	DEPENDENCIA RECEPTORA
-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----

- Ejemplo del formato de impresión de Mobiliario Selectivos por Oficina del sistema:

**B A N C O X X X X X X X X X X**  
**SISTEMA DE CONTROL FISICO DE BIENES MUEBLES**  
**DIRECCION DE CONTRALORIA**  
**CONTABILIDAD DE CAPITAL Y FIDEICOMISOS**  
**RECEPCION DE BIENES REGISTRADOS EN EL MES DE ENERO DE 1995**  
**FECHA: 25/SEP./1995**

**HOJA: 4**

FECHA DE ADQUIS.	DEPENDENCIA RECEPTORA	DESCRIPCION DEL BIEN	NUM. DE SERIE	IMPORTE	DEPENDENCIA DISTRIBUIDORA
-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----

## **CAPÍTULO 5**

# **PRUEBAS, MANTENIMIENTO E INSTALACIÓN**

El propósito de este capítulo es presentar un panorama general de la instalación, pruebas y mantenimiento al Sistema de Control y Actualización del Activo Fijo.

Inicia este capítulo con una breve descripción del proceso de instalación del sistema SICOFI. Posteriormente presentaremos algunas pruebas realizadas al sistema SICOFI, reportando los resultados en una matriz de pruebas.

Finalmente mencionaremos los tipos de mantenimiento que se le puede dar a un sistema, así como las conclusiones finales del capítulo.

### **5.1 Instalación del sistema**

Para la instalación del sistema SICOFI es necesario adecuar los directorios dentro de la red y la preparación de los nodos que tienen acceso al sistema. En primer lugar veamos la configuración de los directorios dentro de la red.

El servidor de archivos maneja el disco duro de la red en dos volúmenes SYS: y SYS2:. En el primer volumen (SYS:) se tiene el sistema operativo de la red, paquetería y directorios de trabajo de los usuarios de la red. Ambos volúmenes tienen una capacidad de 850 Mb.

La estructura del árbol de directorios de este volumen se muestra en la *figura 5.1*.

### **5.2 Descripción de los directorios**

BAT. A este directorio tienen acceso todos los usuarios con derechos de lectura y búsqueda para realizar procesos o acceder paquetes a la red.

SYSTEM. Contiene al sistema operativo de la red.

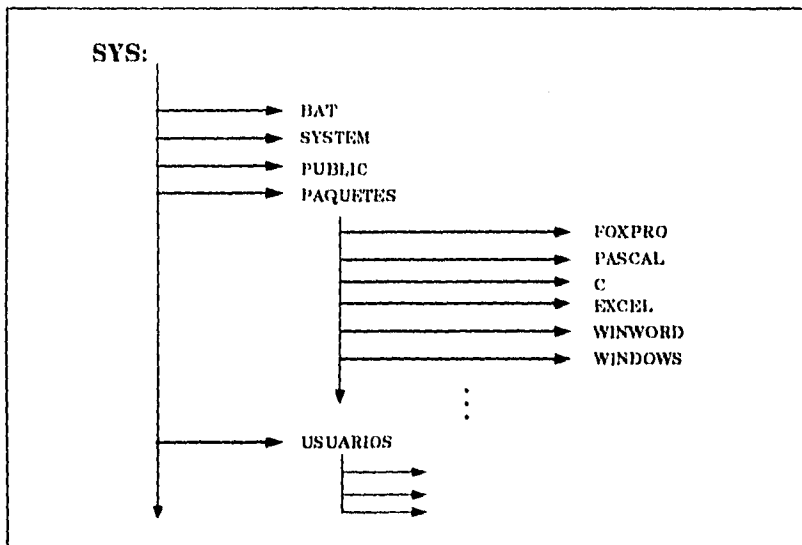


Fig 5.1 Estructura del Árbol de Directorios del volumen SYS de la Red

PUBLIC. Contiene utilerías de la red; los usuarios tienen acceso a este directorio con derechos de lectura y búsqueda de archivos.

PAQUETES. Bajo este directorio se encuentra la paquetería que puede acceder el usuario para satisfacer sus necesidades en cuanto a manejo de procesadores de texto, hojas de cálculo, paquetes de programación, diseño de diagramas, etc. Los usuarios tienen derechos de lectura y búsqueda de archivos sobre este directorio y sus niveles inferiores.

USUARIOS. Bajo este directorio se encuentran localizados los directorios personales de los usuarios de la red, sobre los cuales tienen derechos de lectura, escritura, creación, borrado, búsqueda y actualización de archivos. Los límites en cuanto a espacio varían de acuerdo a las necesidades particulares de cada uno de ellos.

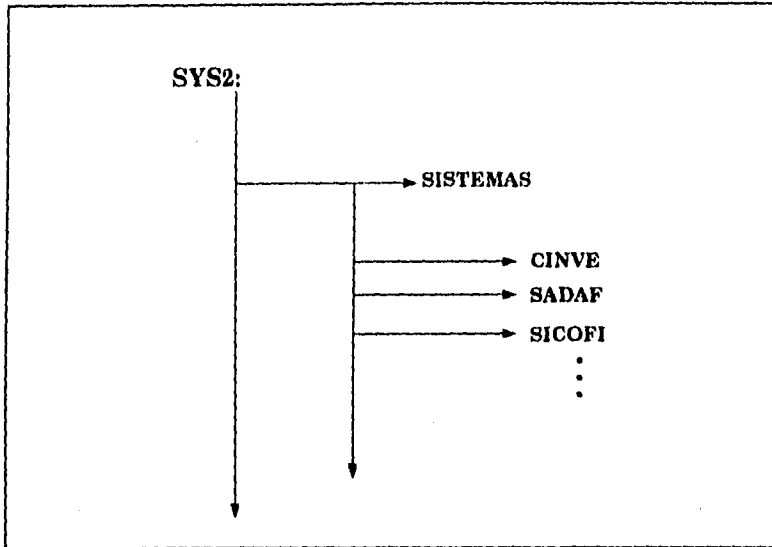
El volumen 2 (SYS2:) contiene los sistemas de aplicación desarrollados por el área de sistemas. La estructura de árbol de este volumen se muestra en la *figura 5.2*.

Para la instalación de SICOFI es necesario la creación del directorio

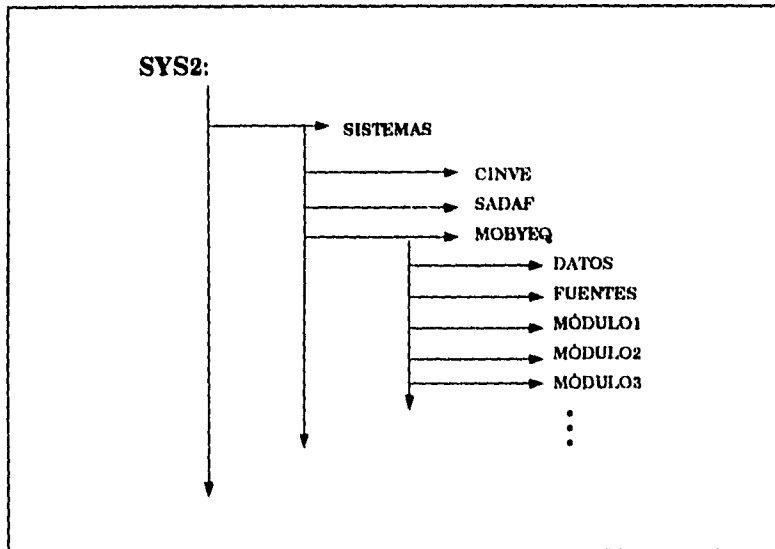
SYS2: SISTEMAS\MOBYEQ

Y la estructura del volumen SYS2: se muestra en la *figura 5.3*.





**Fig. 5.2 Estructura de Árbol del volumen SYS2 de la Red**



**Fig. 5.3 Estructura Final de Árbol del volumen SYS2 de la Red**

También es necesaria la creación del usuario que hará uso del sistema SICOFI. La creación se da a través de la utilidad SYSCON de Netware, la cual hace la asignación de los derechos que el usuario requiere sobre los directorios y archivos de la red. En nuestro caso el usuario SICOFI debe tener los siguientes derechos:

Directorio	Derechos sobre archivos
SYS2: SISTEMAS\MOBYEQ\DATOS	Lectura, escritura, creación borrado y actualización
SYS2: SISTEMAS\MOBYEQ\FUENTES	Lectura y búsqueda.
SYS2: SISTEMAS\MOBYEQ\MODULO1	Lectura y búsqueda.
SYS2: SISTEMAS\MOBYEQ\MODULO2	Lectura y búsqueda.
SYS2: SISTEMAS\MOBYEQ\MODULO3	Lectura y búsqueda.

### 5.3 Pruebas del sistema

El proceso de prueba es la parte en el desarrollo de sistemas que nos indica el nivel de eficiencia de nuestro producto, para ello los sistemas se dividen en módulos y éstos son probados minuciosamente con el fin de detectar errores en los resultados o insuficiencias del sistema.

Con el fin de detectar y corregir los errores en el sistema SICOFI se procedió a analizarlo por etapas, estas etapas son:

- Pruebas de funciones. Se verifica la operación de las funciones en forma individual.
- Prueba de módulos. Se verifica la operación de funciones que actúa en conjunto y forman un módulo.
- Prueba de subsistema. Se verifica la operación entre módulos para que entreguen un resultado correcto entre ellos.
- Prueba del sistema. Se verifica la relación en cuanto a traspaso de información de un subsistema a otro.
- Prueba de aceptación. Se realiza una prueba general del sistema utilizando datos reales y comparando los resultados contra una fuente confiable de la Institución.

En el sistema SICOFI se realizaron pruebas aplicando los enfoques ascendente y descendente. Las pruebas de tipo ascendente (ver **figura 5.4.**) se utilizaron para probar módulos e irlos incorporando al sistema. Para probar los programas que implicaban la relación padre-hijos se utiliza la prueba ascendente, tomando en cuenta que las funciones utilizadas por los módulos no requerían la simulación de módulos para niveles inferiores.

Para el sistema SICOFI se conformó el mismo grupo de programadores que realizaron el programa para llevar a cabo la etapa de pruebas, primero cada programador realizó pruebas a los módulos en el que el participó, después los programadores realizaron pruebas a módulos en los cuales no había participado, detectando errores que no se habían detectado al principio.

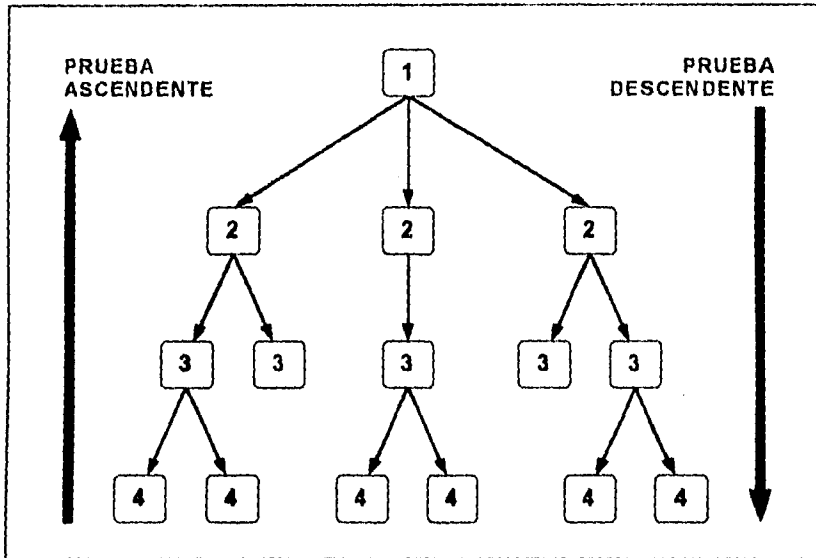


Fig. 5.4 Pruebas Ascendentes y Descendentes

Con los datos obtenidos de las pruebas realizadas al sistema por los programadores se realizó una matriz de prueba, la cual se muestra a continuación.

Objetivo de la Prueba	Módulo	Indica mensaje de error	Diseño de casos de prueba	Resultados observados	Rechazo
Validar que la fecha registrada sea anterior o igual a la fecha de control	Altas	<input type="checkbox"/>	Al introducir una fecha posterior a la fecha definida como de control	Mensaje del sistema "Invalid Input"	<input type="checkbox"/>
Verificar que la clave del artículo sea válida	Altas	<input type="checkbox"/>	Introducir una clave de artículo que no existe	Muestra mensaje en ventana con la indicación de que la clave no existe y tiene la posibilidad de adicionarla al catálogo	<input type="checkbox"/>
Verificar que se digite el número de serie, si el artículo lo requiere, de acuerdo al catálogo	Altas y Cambios	<input type="checkbox"/>	Registrar un artículo que requiera número de serie	Despliegue del mensaje 'Se requiere número de serie'	<input type="checkbox"/>

Objetivo de la Prueba	Módulo	Indica mensaje de error	Diseño de casos de prueba	Resultados observados	Rechazo
Determinar que se valide la oficina a la que pertenece el artículo	Altas y Cambios	<input type="checkbox"/>	Introducir una clave de oficina que no exista	Despliegue del mensaje 'La referencia no existe', y se pueda adicionarla al catálogo en ese momento.	<input type="checkbox"/>
Verificar que se valide la clave de tipo de receptor	Altas y Cambios	<input type="checkbox"/>	Introducir una clave de tipo de receptor que no exista	Despliegue del mensaje con la indicación de que la clave no existe, y la posibilidad de adicionarla al catálogo	<input type="checkbox"/>
Determinar que el importe del artículo a registrar sea diferente de cero	Altas y Cambios	<input type="checkbox"/>	Introducir un importe de cero para el artículo	Despliegue del mensaje de sistema 'Invalid input'	<input type="checkbox"/>
Verificar que no se dupliquen las claves de artículos en el catálogo de grupos	Mantto.	<input type="checkbox"/>	Introducir una clave de artículo que ya exista	Despliegue del mensaje con la indicación de que ya existe la clave del artículo	<input type="checkbox"/>
Determinar que no se pueda borrar una clave de artículo si existe en el archivo maestro al menos un artículo con esa clave	Mantto.	<input type="checkbox"/>	Seleccionar para dar de baja una clave de artículo que esté referenciada en el archivo maestro	Despliegue del mensaje con la indicación de que no es posible dar de baja la clave del grupo	<input type="checkbox"/>
Verificar que las altas al catálogo de grupos sean válidas	Mantto.	<input type="checkbox"/>	Insertar un registro al catálogo de grupos con clave de artículo para la cual no existe nivel superior	Despliegue de mensaje que indica que el nivel del grupo no es válido	<input type="checkbox"/>
Verificar el adecuado funcionamiento de las cifras de control	Cifras de control	<input type="checkbox"/>	Digitar cifras de control correctas e incorrectas respecto a lo capturado	Las cifras de control funcionaron adecuadamente	<input type="checkbox"/>

## 5.4 Mantenimiento

El mantenimiento de Software ha sido caracterizado como un "Iceberg": Lo ideal sería que lo que está visible fuera todo el problema. En la realidad sabemos que una enorme cantidad de problemas potenciales de costo están bajo la superficie.

El mantenimiento del software existente puede representar hasta el 70 por ciento de todo el esfuerzo de una organización que se dedique al desarrollo de software. El porcentaje continúa en aumento conforme se produce más software. En el horizonte podemos prever que no se puede producir más software porque todos los recursos disponibles se destinan al mantenimiento de software instalado previamente.

El cambio es inevitable cuando se desarrolla un sistema basado en computadora; por eso, es necesario desarrollar mecanismos para evaluar, controlar, y hacer modificaciones. El uso de técnicas como el análisis estructurado ayudan a mejorar la facilidad con la cual los cambios se pueden efectuar, y reducir el esfuerzo necesario para dar mantenimiento.

### **Definición de mantenimiento de software**

Se puede definir el mantenimiento describiendo cuatro actividades que se toman después de liberado un sistema:

La primera actividad del mantenimiento ocurre porque no es razonable asumir que las pruebas de software podrán descubrir todos los errores latentes en un sistema grande.

Durante el uso de cualquier programa extenso, ocurrirán errores y se reportarán al encargado del desarrollo. El proceso que incluye el diagnóstico y corrección de uno o más errores se le llama *Mantenimiento Correctivo*.

La segunda actividad que contribuye a una definición de mantenimiento ocurre debido al rápido cambio que se encuentra en cada aspecto de la computación. Las nuevas generaciones de hardware son anunciadas en ciclos de 24 meses; sistemas operativos nuevos, y liberaciones de otros ya existentes, aparecen regularmente; equipo periférico y otros elementos de sistemas con frecuencia son actualizados o modificados.

La vida útil de software de aplicación, por otro lado, puede fácilmente sobrepasar los 10 años, sobreviviendo al ambiente para el cual fue desarrollado originalmente. Por ello, el *Mantenimiento Adaptativo* - una actividad que modifica al software para interactuar adecuadamente con un medio ambiente cambiante - es a la vez necesario y frecuente.

La tercera actividad que se puede aplicar a una definición de mantenimiento ocurre cuando un paquete de software es exitoso. Conforme el software se va usando, se reciben recomendaciones para nuevas capacidades, modificaciones a funciones existentes y mejoras en general. Para satisfacer los requerimientos en esta categoría, se da el *Mantenimiento Perfectivo*. Esta actividad requiere del mayor esfuerzo de todo el mantenimiento del software.

La cuarta actividad ocurre cuando se dan los cambios para mejorar en el futuro el mantenimiento, o para tener una mejor base para futuras mejoras, también llamada *Mejoramiento Preventivo*.

En la actualidad, las tareas que ocurren como parte del mantenimiento Adaptativo y Perfectivo son las mismas que se aplican durante la fase de desarrollo. Para adaptar o perfeccionar debemos determinar nuevos requerimientos, generar código, y probar software existente. Tradicionalmente, estas tareas, cuando se aplican a un programa existente, han recibido el nombre de Mantenimiento.

### **Características del mantenimiento**

Para entender las características del mantenimiento al Software, consideremos tres puntos de vista:

1. Las actividades requeridas para completar la fase de mantenimiento y el impacto de una metodología adecuada en la eficacia de tales actividades.
2. El costo asociado con la fase de mantenimiento.
3. Los problemas que se encuentran frecuentemente cuando se aplica el mantenimiento al software.

A continuación proporcionamos las características del mantenimiento desde estos tres puntos de vista.

### **Mantenimiento estructurado contra no estructurado**

Si el único elemento disponible es el código fuente, las actividades del mantenimiento comienzan con la evaluación del código, que se complica con una pobre documentación interna. Algunas características, como la estructura del programa, estructuras de datos globales, interfaces del sistema, y rendimiento son difíciles de identificar, y con frecuencia se mal interpretan. Las ramificaciones de los cambios hechos al código se identifican con dificultad. Las pruebas regresivas (repetiendo pruebas pasadas para asegurar que las modificaciones no han introducido fallas en el software) son imposibles de conducir porque no existen registros previos de pruebas. En este caso estamos hablando de mantenimiento no estructurado y se paga el precio (en esfuerzo y frustración) que acompaña al software que no ha sido desarrollado usando una metodología bien definida.

Por otro lado, si existe una configuración completa de software, la tarea del mantenimiento comienza con una evaluación de la documentación del diseño. Se determinan importantes características del software en cuanto a estructura,

rendimiento e interface. Se asienta el impacto de las modificaciones o correcciones requeridas y se planea una primera solución.

El diseño se modifica (usando técnicas como el diseño estructurado) y se revisa. El nuevo código se desarrolla y se efectúan las pruebas de regresión, el software se libera de nuevo.

Esta secuencia de eventos constituyen el mantenimiento estructurado y ocurren como el resultado de una aplicación previa desarrollada con una metodología adecuada. Aunque la existencia de una configuración de software no garantiza un mantenimiento sin problemas, la cantidad de esfuerzo se reduce y la calidad se mejora al hacer un cambio o corrección.

### **Costo de mantenimiento**

El costo del mantenimiento al software se ha incrementado notablemente durante los últimos 20 años. Durante los 70's el mantenimiento representaba entre el 35 y el 40 por ciento de los sistemas de información de una organización. Este número brincó hasta aproximadamente el 60 por ciento durante los 80's. Si no se hace nada para mejorar la forma de dar mantenimiento, muchas compañías gastarán cerca del 80 por ciento en mantenimiento para finales de los 90's.

Un costo intangible del mantenimiento del software es que se pospone o pierde la oportunidad de desarrollar nuevos sistemas, porque los recursos disponibles deben canalizarse hacia tareas de mantenimiento. Otros costos intangibles incluyen:

- Reducción en la calidad del software como resultado de los cambios requeridos en el mantenimiento.
- Insatisfacción de los usuarios cuando sus reclamos para reparación o modificación no se pueden direccionar de una manera calendarizada.

El costo final del mantenimiento del software es un dramático descenso en la productividad que se encuentra cuando se inicia el mantenimiento en programas viejos.

### **Problemas**

La mayoría de los problemas asociados con el mantenimiento al software pueden asociarse a deficiencias en la forma en que se planeó y desarrolló el software. Una falta de control y disciplina en las actividades de desarrollo casi siempre se traducen en problemas durante el mantenimiento.

Entre los problemas clásicos asociados con el mantenimiento de software se encuentran los siguientes:

1. Es difícil o imposible rastrear la evolución del software a través de varias versiones o liberaciones.
2. Lo es también el rastreo del proceso mediante el cual se creó el software.
3. Es excepcionalmente difícil entender el programa desarrollado por otros. La dificultad se incrementa conforme el número de elementos se decrementa en una configuración de software. Si sólo existe código sin documentación, se pueden esperar graves problemas.
4. La persona que desarrolló los programas tal vez ya no esté. La movilidad entre personal de software es muy alta.
5. La documentación no existe o tiene carencias. El reconocimiento de que el software debe estar bien documentado es un primer paso, pero la documentación debe ser comprensible y consistente con el código fuente para que sea útil.
6. La mayoría del software se diseña sin considerar los cambios. A menos que un método de diseño acomode los cambios en conceptos como independencia funcional o clases de objetos, las modificaciones al software son difíciles y propensas a errores.

#### **5.4.1 Mantenimiento a la red**

El mantenimiento de una red local es necesario una vez que la red entra en operación.

Para dar el mantenimiento el supervisor de la red deberá supervisar periódicamente los parámetros de seguridad del software de la red, así como el vigilar que el cableado tanto de comunicación como el de suministro eléctrico estén en perfectas condiciones para su uso.

Además de las revisiones periódicas, el supervisor tendrá que adecuar cuando sea necesario las restricciones de uso de los archivos, usuarios y grupos de usuarios. También deberá reportar las fallas en el hardware cuando éstas ocurran, o si es posible repararlas en el momento de su ocurrencia.

Las tareas del supervisor referentes al mantenimiento de la red pueden ser auxiliadas si se llevan controles escritos. La utilización de bitácoras son de gran ayuda, ya que éstas servirán como apoyo al supervisor para poder tener un



perfecto control de la situación actual de la red, así como, los cambios que ha sufrido la red desde que está en operación; asimismo, será posible realizar un análisis de:

- La utilización de la red por usuario.
- El promedio de uso de cada nodo.
- El promedio de fallas en la red, tanto de hardware como de software.

Sobre todo el objetivo primordial del mantenimiento de la red, es el poder medir constantemente el "performance" (rendimiento) de la red para poder mantenerla en optimas condiciones de operación.

Otra recomendación que se puede dar es el de tener una persona que en un momento dado pueda sustituir al supervisor, en caso de que éste no se encuentre.

#### **5.4.2 Mantenimiento al sistema**

El mantenimiento al sistema es la última fase del proceso de ingeniería de software, la cual se lleva la mayor parte de todo el dinero gastado en el software de computadora. A medida que se desarrollan más programas surge una problemática, la cantidad de esfuerzo y de recursos gastados en el mantenimiento de software crece y crece.

Consideramos que nuestro sistema puede recibir los siguientes 4 tipos de mantenimiento, dependiendo de las necesidades futuras de la institución.

1. *Mantenimiento correctivo.* Este tipo de mantenimiento persigue la corrección de errores que no se detectaron antes de poner en uso el software, estos errores se reportarán al encargado del desarrollo para que haga un diagnóstico y la corrección de cada uno de ellos. Este tipo de mantenimiento es el que más recibió nuestro sistema.
2. *Mantenimiento adaptativo.* Este tipo de actividad modifica al software para que interactúe adecuadamente con el medio ambiente cambiante, éste se da debido al cambio rápido en la computación, ya que a cada momento surgen nuevas generaciones de hardware, nuevos sistemas operativos y equipo periférico, es por eso que nuestro software se tiene que adaptar al medio ambiente actual.

Para nuestro sistema el mantenimiento adaptativo fue muy importante, ya que en un principio el registro de los activo fijos se tenía en un sistema de registro a base de tarjetas perforadas, con el tiempo se introdujeron terminales remotas y con esto fue posible implantar el sistema actual de registro de los bienes de la institución.

3. *Mantenimiento perfectivo*. Este tipo de mantenimiento surge en el momento en que nuestro software es exitoso, conforme nuestro software se va utilizando se van dando recomendaciones al encargado del desarrollo para ser mejoras en general al sistema, por ejemplo: crear ventanas de ayuda, mensajes, adición de campos. Es por eso que este tipo de mantenimiento requiere mayor esfuerzo, debido a que debemos determinar nuevos requerimientos como generar más código y probar el software que ya se tiene.
4. *Mantenimiento preventivo*. Este tipo de mantenimiento ocurre cuando se dan los cambios para mejorar en el futuro y con esto tener una mejor base para futuras mejoras, por ejemplo: interfaces con nuevos sistemas, adición de módulos, adición de archivos, etc.

## RESULTADOS Y CONCLUSIONES

### Resultados

En el presente trabajo describimos los pasos necesarios para planear un proyecto de desarrollo de sistemas, también se indicaron algunos factores que deben considerarse cuando se desarrollan el plan del proyecto. Se analizaron los costos del producto de programación así como los factores más importantes que influyen en el costo total del software. Se utilizó el juicio experto que es una de las técnicas más utilizadas para la estimación de costos. También se describió la técnica DELFI por medio de un grupo de expertos, además se presentaron los aspectos más importantes del modelo constructivo COCOMO.

Por otra parte se describieron varias herramientas para los requisitos de programación, la mayor parte de las herramientas para la definición de estos requisitos son herramientas de análisis y diseño, ya que incorporan aspectos estructurales y detallados como el manejador de base de datos FOX-PRO.

Se analizaron los conceptos fundamentales del diseño incluyendo cubrimiento de información, estructura, modularidad, verificación de concurrencia y estética del diseño. Las notaciones del diseño que se analizaron incluyen los diagramas de flujo de datos pseudocódigo y diagramas de flujo estructurados.

Nuestro sistema de información se organizó en archivos para acumular y almacenar datos para su procesamiento. Los archivos contienen registros relacionados de datos y describen entidades de importancia para la empresa. Los principales tipos de archivos utilizados en el sistema de información son: el archivo maestro, de transacciones, de tablas y de informe. El archivo maestro es un archivo permanente que existe durante la vida del sistema pero debe de mantenerse actualizado para ser de utilidad conforme ocurren los hechos que incluyen a la empresa.

Los archivos de transacciones son procesados contra el archivo maestro para actualizarlo. Los archivos de tablas se utilizan con el objeto de almacenar datos de referencias y también cuando se produce la salida o se procesa las transacciones. Los archivos de referencias son los que acumulan la salida almacenada temporalmente en un disco hasta que se pueda imprimir.

El propósito de un sistema de información es el apoyo a las consultas de usuarios, es por eso la necesidad de un sistema de manejo de datos, ya que es un programa que nos sirve entre las estructuras de los archivos que almacenan los datos y las estructuras de datos que representan las necesidades de los usuarios.

Se analizaron las técnicas de programación en la producción de código fuente para que sea fácil de leer y comprender. Se utilizaron las técnicas de codificación estructurados, el cuaderno de notas de cada unidad de programa y la documentación interna del código fuente, se analizó la importancia de documentos de apoyo tales como: la especificación de requisitos, documentación del diseño, planes de prueba y manuales de usuarios. Los cuadernos de notas de cada unidad se utilizan para organizar las actividades de trabajo y para conservar los documentos de unidades de programas individuales. Una unidad de programa es una unidad de código fuente que es desarrollada ó le da un mantenimiento un programador individual. Se analizó el formato y contenido del cuaderno de notas de algunas unidades del programa ya que esto juega un papel importante en el desarrollo de la programación.

Se estudiaron técnicas para evaluar y mejorar la calidad del software, las cuales incluyen procedimientos sistemáticos de depuración y pruebas de unidad, de integración y de aceptación. En la práctica se requiere una combinación de técnicas para evaluar y mejorar la calidad del software, el análisis y prueba de unidad proporcionan información complementaria de estructura y comportamiento del código fuente, los principales objetivos de la verificación y validación son evaluar y mejorar la calidad de los distintos productos de trabajo generados durante el desarrollo del software. La mejor calidad de un producto de software se logra mejor con una atención más cuidadosa en los detalles de planeación sistemática, análisis, diseño e implementación.

También se estudió las diferentes actividades del mantenimiento del software para realizar mejoras al sistema, adaptar el sistema a nuevos ambientes y corregir problemas que pudieran surgir.

Finalmente se dio una descripción de la instalación del sistema, ya que éste se encuentra dentro de una red, esta red tiene como objetivo principal compartir recursos materiales como equipos y periféricos y recursos informáticos como datos y programas.

## Conclusiones

Nosotros consideramos que la principal aportación del sistema radica en el hecho de que fue diseñado para operar en una red local, lo cual permite a todas las áreas o departamentos involucrados en los procesos de registro y actualización de activos fijos contar con una herramienta en línea que facilita un mejor desempeño de sus labores en la institución. Asimismo el sistema permite tener la información actualizada y oportuna sin necesidad de esperar el procesamiento de los activos en el mainframe.

Es importante destacar que la participación activa de los usuarios finales del sistema permitió precisar en su oportunidad los objetivos, los alcances y las características en general del mismo. De las experiencias derivadas en el desarrollo, observamos que una parte fundamental del éxito del sistema fueron las fases de análisis y diseño pues tratamos que estos apartados fueran completos y de alta calidad.

Por otra parte el sistema reduce considerablemente los costos de mantenimiento de software y hardware, es decir, el costo del procesamiento y mantenimiento de hardware de un mainframe es más alto que el de una red local. En lo referente al software, es más costoso y complicado mantener un sistema en COBOL que en FOX-PRO, debido a que FOX-PRO es un manejador de base de datos que cuenta con herramientas prediseñadas que facilitan el manejo de archivos, el mantenimiento de sistemas y la interfaz gráfica entre otras.

Hay que mencionar que en la selección de la red local y de la herramienta de programación fue necesario ajustarnos a las normas establecidas por la institución, nosotros pensamos que se pudieron utilizar otras herramientas para mejorar el funcionamiento y diseño del mismo.

El diseño de este sistema tiene una gran flexibilidad en el manejo de catálogos, lo cual nos da como resultado que el usuario final tenga una mejor clasificación de la información ya que de un activo puede conocerse el grupo, subgrupos, localización física, área de adjudicación, responsable, etc. Lo anterior deriva en un mejor control de sus bienes y ayuda en la toma de decisiones para futuras adquisiciones y bajas de los activos con que contará la institución.

Otro de los aspectos que cabe mencionar dentro del sistema es el manejo que da a las bajas. Pues este nos da la facilidad de realizar bajas a través de un grupo y no artículo por artículo, asimismo mantiene un archivo histórico con todos los movimientos de bajas realizadas durante un período de tiempo.

En una segunda fase, el software debe ser revisado por el personal de sistemas de la institución bancaria, derivando en sugerencias y mejoras para la aplicación.

Probablemente debe fortalecerse la interfaz gráfica con el usuario y las salidas impresas.

Esperamos que esta tesis haya impartido algunos de los métodos, herramientas y técnicas de la ingeniería de software como ayuda al lector para elevar su nivel académico.

## BIBLIOGRAFÍA

- Archer, R.  
The Practical Guide to Local Area Network.  
Mc Graw Hill.
- Brooner, E.G.  
The Local Area Network Book  
Howard Sams & Co.
- Del Rio González Cristobal  
Introducción al Estudio de la Contabilidad de los Costos Industriales  
ECASA
- Fairley, Richard  
Ingeniería de Software  
Mc Graw Hill
- Lehrman, S. R.  
Local Area Networking with Microcomputer (A Guide for the Business Decision-Mak) Prentice Hall.
- Reiss.  
Introduction Local Area Network with Microcomputers Experiments  
Prentice Hall.
- Teanenbaun, A.  
Computer Networks  
Prentice Hall.
- Pyle William, Larson D. Kermit  
Principios Fundamentales de Contabilidad Primer Curso  
CECSA
- Senn, James  
Análisis y Diseño de Sistemas de Información  
Mc Graw Hill
- Date, C.J.  
Introducción a las Bases de Datos  
Adison-Wensley
- Durr Michael.  
Networking IBM PCs a Practical Guide  
Director
- Korth, Henry  
Fundamentos de Base de Datos  
Mc Graw Hill
- Manual de usuario de FOX-PRO  
Relation Database Management System  
for MS-DOS and WINDOWS
- Pressman, Roger  
Software Engineering  
Mc Graw Hill
- Torres Tovar Juan Carlos  
Contabilidad II  
DIANA

**APÉNDICE A1.  
ESTRUCTURA DE BASES DE DATOS DEL SISTEMA**

Estructura de la base de datos : **CUENTA.DBF** Alias: **CUENTA**

Field	Field name	Type	Width	Dec	Start	End
1	CUENTA	Numeric	2		1	2
2	MAYOR	Numeric	3		3	5
3	NOMBRECTA	Character	50		6	55

Estructura de la base de datos : **SESBAJ.DBF** Alias: **SESBAJ**

Field	Field name	Type	Width	Dec	Start	End
1	NUMREG	Numeric	6		1	6
2	IMPORTE	Numeric	19	2	7	25

Estructura de la base de datos : **SESION.DBF** Alias: **SESION**

Field	Field name	Type	Width	Dec	Start	End
1	NUMREG	Numeric	6		1	6
2	IMPORTE	Numeric	19		27	25

Estructura de la base de datos : **GRUPAUX.DBF** Alias: **GRUPAUX**

Field	Field name	Type	Width	Dec	Start	End
1	NIVEL	Numeric	2		1	2
2	DESCRIPCIO	Character	50		3	52



Estructura de la base de datos : **CAMBIOS.DBF** Alias: **CAMBIOS**

Field	Field name	Type	Width	Dec	Start	End
1	ANIO_CAMB	Numeric	2		1	2
2	MES_CAMB	Numeric	2		3	4
3	NUM_MOV	Numeric	5		5	9
4	ANT_DESP	Character	1		10	10
5	MES	Numeric	2		11	12
6	ANIO	Numeric	2		13	14
7	MAYOR	Numeric	3		15	17
8	NIVEL_1	Numeric	2		18	19
9	NIVEL_2	Numeric	2		20	21
10	NIVEL_3	Numeric	2		22	23
11	NUM_SERIE	Character	20		24	43
12	PERTENECEA	Character	20		44	63
13	REFERENCIA	Character	3		64	66
14	TIPO_RECEP	Numeric	1		67	67
15	IMPORTE	Numeric	18	2	68	85
16	SECTOR	Character	1		86	86

Estructura de la base de datos : **ALTAS.DBF** Alias: **ALTAS**

Field	Field name	Type	Width	Dec	Start	End
1	MES	Numeric	2		1	2
2	ANIO	Numeric	2		3	4
3	MAYOR	Numeric	3		5	7
4	NIVEL_1	Numeric	2		8	9
5	NIVEL_2	Numeric	2		10	11
6	NIVEL_3	Numeric	2		12	13
7	NUM_SERIE	Character	20		14	33
8	PERTENECEA	Character	20		34	53
9	REFERENCIA	Character	3		54	56
10	TIPO_RECEP	Numeric	1		57	57
11	IMPORTE	Numeric	18	2	58	75
12	SECTOR	Character	1		76	76

Estructura de la base de datos : **MOBAUX.DBF** Alias: **MOBAUX**

Field	Field name	Type	Width	Dec	Start	End
1	NUMREG	Numeric	6		1	6
2	NUMREGA	Numeric	6		7	12
3	NIVEL_1	Numeric	2		13	14
4	NIVEL_2	Numeric	2		15	16
5	NIVEL_3	Numeric	2		17	18

Field	Field name	Type	Width	Dec	Start	End
6	ANIO	Numeric	2		19	20
7	MES	Numeric	2		21	22
8	NUM_SERIE	Character	20		23	42
9	PERTENECEA	Character	20		43	62
10	REFERENCIA	Character	3		63	65
11	TIPO_RECEP	Numeric	1		66	66
12	IMPORTE	Numeric	18	2	67	84
13	SECTOR	Character	1		85	85

Estructura de la base de datos : **BAJAS.DBF** Alias: **BAJAS**

Field	Field name	Type	Width	Dec	Start	End
1	ANIOBAJ	Numeric	2		1	2
2	MESBAJ	Numeric	2		3	4
3	ANIOREG	Numeric	2		5	6
4	MESREG	Numeric	2		7	8
5	MAYOR	Numeric	3		9	11
6	NIVEL_1	Numeric	2		12	13
7	NIVEL_2	Numeric	2		14	15
8	NIVEL_3	Numeric	2		16	17
9	DESCGRUPO	Character	100		18	117
10	NUM_SERIE	Character	20		118	137
11	PERTENECEA	Character	20		138	157
12	REFERENCIA	Character	3		158	160
13	TIPO_RECEP	Numeric	1		161	161
14	IMPORTE	Numeric	18	2	162	179
15	SECTOR	Character	1		180	180

Estructura de la base de datos : **TABRECEP.DBF** Alias: **TABRECEP**

Field	Field name	Type	Width	Dec	Start	End
1	TIPO_RECEP	Numeric	1		1	1
2	DESCRIPCIO	Character	50		2	51

Estructura de la base de datos : **CATREFER.DBF** Alias: **CATREFER**

Field	Field name	Type	Width	Dec	Start	End
1	REFE_ANTER	Character	3		1	3
2	DESCRIPCIO	Character	50		4	53
3	RESPONSAB	Character	50		54	103
4	UBICACION	Character	50		104	153
5	SECTOR	Character	1		154	154
6	DIRECCION	Character	50		155	204
7	REFERENCIA	Character	3		205	207

Estructura de la base de datos : **CATGRUPO.DBF** Alias: **CATGRUPO**

Field	Field name	Type	Width	Dec	Start	End
1	MAYOR	Numeric	3		1	3
2	NIVEL_1	Numeric	2		4	5
3	NIVEL_2	Numeric	2		6	7
4	NIVEL_3	Numeric	2		8	9
5	OPC1	Logical	1		10	10
6	OPC2	Logical	1		11	11
7	OPC3	Logical	1		12	12
8	DESCRIPCIO	Character	50		13	62

Estructura de la base de datos : **MOBYEQUI.DBF** Alias: **MOBYEQUI**

Field	Field name	Type	Width	Dec	Start	End
1	MES	Numeric	2		1	2
2	ANIO	Numeric	2		3	4
3	MAYOR	Numeric	3		5	7
4	NIVEL_1	Numeric	2		8	9
5	NIVEL_2	Numeric	2		10	11
6	NIVEL_3	Numeric	2		12	13
7	NUM_SERIE	Character	20		14	33
8	NUM_SERIEA	Character	20		34	53
9	NUM_PLACA	Character	10		54	63
10	PERTENECEA	Character	20		64	83
11	REFERENCIA	Character	3		84	86
12	TIPO_RECEP	Numeric	1		87	87
13	IMPORTE	Numeric	18	2	88	105
14	SECTOR	Character	1		106	106
15	NUM_REGIST	Character	6		107	112

Estructura de la base de datos : **ARCHIVOS.DBF** Alias: **ARCHIVOS**

Field	Field name	Type	Width	Dec	Start	End
1	NOMBRE	Character	8		1	8
2	COMENTARIO	Character	40		9	48
3	FECHA_CREA	Date	8		49	56
4	PROTEGIDO	Logical	1		57	57

Estructura de la base de datos : **PROCESOS.DBF** Alias: **PROCESOS**

Field	Field name	Type	Width	Dec	Start	End
1	NOMBRE	Character	8		1	8
2	COMENTARIO	Character	40		9	48

3	FECHA_CREA	Date	8		49	56
4	PROTEGIDO	Logical	1		57	57
5	ARCHIVO	Character	8		58	65
6	ORDEN	Character	12		66	77
7	FILTRO	Character	8		78	85
8	COMPRESOR	Character	8		86	93
9	REPORTE	Character	8		94	101

Estructura de la base de datos : OPCIO.PDBF Alias: OPCIO.P

Field	Field name	Type	Width	Dec	Start	End
1	NOMBRE	Character	8		1	8
2	COMENTARIO	Character	40		9	48
3	TITPAG1	Character	75		49	123
4	TITPAG2	Character	75		124	198
5	TITPAG3	Character	75		199	273
6	ANCHOPAG	Numeric	3		274	276
7	LINEASXPAG	Numeric	2		277	278
8	DOBLES P A C I	Logical	1		279	279
9	MILES	Logical	1		280	280
10	SALTANTES	Logical	1		281	281
11	SALTADESP	Logical	1		282	282
12	NUMERAPAG	Logical	1		283	283
13	FECHA	Logical	1		284	284

Estructura de la base de datos : C O L U M P . D B F Alias: C O L U M P

Field	Field name	Type	Width	Dec	Start	End
1	NOMBRE	Character	8		1	8
2	CONTENIDO	Character	10		9	18
3	TOTALCOLUM	Logical	1		19	19
4	SINOMREF	Logical	1		20	20
5	SINUMYREF	Logical	1		21	21
6	SINOMCTA	Logical	1		22	22
7	SINUMYCTA	Logical	1		23	23
8	SINOMREC	Logical	1		24	24
9	SINUMYREC	Logical	1		25	25
10	NOREF	Logical	1		26	26
11	NOREC	Logical	1		27	27
12	NOCTA	Logical	1		28	28
13	ANCHONOM	Logical	3		29	31

Estructura de la base de datos : **ORDENES.DBF** Alias: **ORDENES**

Field	Field name	Type	Width	Dec	Start	End
1	NOMBRE	Character	8		1	8
2	COMENTARIO	Character	40		9	48

Estructura de la base de datos : **NOMFILT.DBF** Alias: **NOMFILT**

Field	Field name	Type	Width	Dec	Start	End
1	FILTRO	Character	8		1	8
2	COMENTARIO	Character	60		9	68

Estructura de la base de datos : **AFILTROS.DBF** Alias: **AFILTROS**

Field	Field name	Type	Width	Dec	Start	End
1	FILTRO	Character	8		1	8
2	CAMPO	Character	25		9	33
3	OPERADOR	Character	2		34	35
4	TEXTO	Character	20		36	55
5	CONECTOR	Character	5		56	60
6	TIPCAM	Character	1		61	61

**APÉNDICE A2.  
REFERENCIAS CRUZADAS PROGRAMAS-ARCHIVOS**

ARCHIVOS	PROGRAMAS
ARCHIVOS.DBF	ARCHIVOS.PRG
SESB AJ.DBF	CIFRBAJ.PRG
CAMBIOS.DBF	MENU.PRG
MOBAUX.DBF	MENU.PRG
BAJAS.DBF	MENU.PRG
TABRECEP.DBF	MENU.PRG ARCHIVOS.PRG
CATGRUPO.DBF	MENU.PRG ARCHIVOS.PRG CONSINVEN.PRG EMREPOR.PRG DEPRECIA.PRG MENUPARE.PRG
PROCESOS.DBF	ARCHIVOS.PRG
GRUPOP.DBF	ARCHIVOS.PRG
EXPORTA.DBF	EXPORTA.PRG
QUERYBEX.DBF	FILTROS.PRG
COMPRE1.DBF	DEPRECIA.PRG
COMPRE3.DBF	DEPRECIA.PRG
FURNITURE.DBF	RESGUARD.PRG BORRAORDE.PRG CORSOR.PRG ARMAPROC.PRG CORORDEN.PRG
FILTROS.DBF	CORFIL.PRG CORFILTR.PRG
AFILTROS.DBF	BORRAFILT.PRG CARGAFIL.PRG MODIFIL.PRG

ARCHIVOS	PROGRAMAS
CUENTA.DBF	MENU.PRG
SESION.DBF	CIRFCTRL.PRG
ALTAS.DBF	MENU.PRG
GRUPAUX.DBF	MENU.PRG
KMOBYEQ.DBF	MENU.PRG
CATREFER.DBF	ARCHIVOS.PRG CONSINVEN.PRG
MOBYEQUI.DBF	MENU.PRG ARCHIVOS.PRG BORRAR.C.PRG CONSINVEN.PRG CREASALD.PRG DEPRECIA.PRG
OPCIOP.DBF	ARCHIVOS.PRG
COLUMP.DBF	ARCHIVOS.PRG
MAESTRO.DBF	PASASADA.PRG
ORDENES.DBF	ORDENA.PRG
COMPRE2.DBF	DEPRECIA.PRG
WRESULT.DBF	EMREPOR.PRG
NOMFILT.DBF	BORRAFILT.PRG CARGAFIL.PRG MODIFIL.PRG MENU_MAS.PRG ARMAPROC.PRG
COMPRESOR.DBF	SELEPRO.PRG DEPRECIA.PRG
X2.DBF	CREASALDO.PRG COMPRESOR.PRG CREASALD.PRG

**APÉNDICE A3**  
**REFERENCIAS CRUZADAS ARCHIVOS-CAMPOS**

NOMBRE DEL CAMPO	BASES DE DATOS
ANCHONOM	COLUMP.DBF
ANCHOPAG	OPCIOP.DBF
ARCHIVO	PROCESOS.DBF
ANIOBAJ	BAJAS.DBF
ANIOREG	BAJAS.DBF
ANIO_CAMB	CAMBIOS.DBF
CONTENIDO	COLUMP.DBF
CUENTA	CUENTA.DBF
DESCGRUPO	BAJAS.DBF
MESBAJ	BAJAS.DBF
MESREG	BAJAS.DBF
NOMBRECTA	CUENTA.DBF
NOREC	COLUMP.DBF
NUMREGA	MOBAUX.DBF
NUM_MOV	CAMBIOS.DBF
NUM_PLACA	MOBYEQUI.DBF
NUM_REGCON	MOBYEQUI.DBF
NUM_REGIST	MOBYEQUI.DBF
NUM_SERIEA	MOBYEQUI.DBF
OPC1	CATGRUPO.DBF
OPC2	CATGRUPO.DBF
OPC3	CATGRUPO.DBF
OPERADOR	AFILTROS.DBF
ORDEN	PROCESOS.DBF
PASSWORD	KMOBYEQ.DBF
TITPAG1	OPCIOP.DBF
TITPAG2	OPCIOP.DBF
TITPAG3	OPCIOP.DBF
ANIO	MOBAUX.DBF ALTAS.DBF CAMBIOS.DBF MOBYEQUI.DBF ORDENES.DBF
FILTRO	AFILTROS.DBF PROCESOS.DBF NOMFILT.DBF
DESCRIPCIO	GRUPAUX.DBF CATGRUPO.DBF CATREFER.DBF TABRECEP.DBF
NIVEL	GRUPAUX.DBF KMOBYEQ.DBF
MAYOR	CUENTA.DBF BAJAS.DBF ALTAS.DBF

NOMBRE DEL CAMPO	BASES DE DATOS
ANT_DESP	CAMBIOS.DBF
BORRADO	MOBYEQUI.DBF
CAMPO	AFILTROS.DBF
COMENTARIO	ARCHIVOS.DBF
COMPRESOR	PROCESOS.DBF
CONECTOR	AFILTROS.DBF
DIRECCION	CATREFER.DBF
DOBLESFACI	OPCIOP.DBF
FECHA	OPCIOP.DBF
MES_CAMB	CAMBIOS.DBF
MILES	OPCIOP.DBF
NOREF	COLUMP.DBF
NUMERAPAG	OPCIOP.DBF
TOTALCOLUM	COLUMP.DBF
UBICACION	CATREFER.DBF
REFE_ANTER	CATREFER.DBF
REPORTE	PROCESOS.DBF
RESPONSAB	CATREFER.DBF
SALTADESP	OPCIOP.DBF
SALTANTES	OPCIOP.DBF
SINOMCTA	COLUMP.DBF
SINOMREC	COLUMP.DBF
SINOMREF	COLUMP.DBF
SINUMYCTA	COLUMP.DBF
SINUMYREC	COLUMP.DBF
SINUMYREF	COLUMP.DBF
TEXTO	AFILTROS.DBF
TIPCAM	AFILTROS.DBF
PERTENECEA	BAJAS.DBF MOBYEQUI.DBF CAMBIOS.DBF ALTAS.DBF MOBAUX.DBF
COMENTARIO	NOMFILT.DBF OPCIOP.DBF PROCESOS.DBF
MES	MOBAUX.DBF CAMBIOS.DBF MOBYEQUI.DBF ALTAS.DBF
FECHA_CREA	ARCHIVOS.DBF PROCESOS.DBF
NIVEL_1	CAMBIOS.DBF CATGRUPO.DBF MOBYEQUI.DBF

NOMBRE DEL CAMPO	BASES DE DATOS
	CATGRUPO.DBF CAMBIOS.DBF MOBYEQUI.DBF
NIVEL_2	CAMBIOS.DBF MOBYEQUI.DBF MOBAUX.DBF ALTAS.DBF BAJAS.DBF CATGRUPO.DBF
NOMBRE	OPCIOP.DBF PROCESOS.DBF ARCHIVOS.DBF ORDENES.DBF COLUMP.DBF
REFERENCIA	MOBYEQUI.DBF BAJAS.DBF CAMBIOS.DBF MOBAUX.DBF ALTAS.DBF CATREFER.DBF
NUMREG	SESION.DBF SESBAJ.DBF MOBAUX.DBF
IMPORTE	BAJAS.DBF MOBYEQUI.DBF ALTAS.DBF SESION.DBF SESBAJ.DBF MOBAUX.DBF CAMBIOS.DBF

NOMBRE DEL CAMPO	BASES DE DATOS
	BAJAS.DBF MOBAUX.DBF ALTAS.DBF
NIVEL_3	CATGRUPO.DBF BAJAS.DBF CAMBIOS.DBF MOBAUX.DBF MOBYEQUI.DBF ALTAS.DBF
NUM_SERIE	MOBAUX.DBF CAMBIOS.DBF BAJAS.DBF MOBYEQUI.DBF ALTAS.DBF
TIPO_RECEP	ALTAS.DBF CAMBIOS.DBF MOBAUX.DBF MOBYEQUI.DBF TABRECEP.DBF BAJAS.DBF
PROTEGIDO	PROCESOS.DBF ARCHIVOS.DBF
SECTOR	CATREFER.DBF MOBAUX.DBF BAJAS.DBF ALTAS.DBF



**APÉNDICE A4**  
**DEFINICIÓN DE CAMPOS**

<b>NOMBRE DEL CAMPO</b>	<b>TIPO</b>	<b>LONG.</b>	<b>DECIM.</b>
ANCHONOM	Numérico	3	0
ANCHOPAG	Numérico	3	0
ANIO	Numérico	2	0
ANIOBAJ	Numérico	2	0
ANIOREG	Numérico	2	0
ANIO_CAMB	Numérico	2	0
ANT_DESP	Caracter	1	0
ARCHIVO	Caracter	8	0
BORRADO	Caracter	1	0
CAMPO	Caracter	25	0
COMENTARIO	Caracter	40	0
COMENATRIO	Caracter	60	0
COMENTARIO	Caracter	60	0
COMPRESOR	Caracter	8	0
CONECTOR	Caracter	5	0
CONTENIDO	Caracter	10	0
DOBLES P A C I	Lógico	1	0
CUENTA	Numérico	2	0
DESCGRUPO	Caracter	100	0
DESCRIPCIO	Caracter	50	0
DIRECCION	Caracter	50	0
FECHA	Lógico	1	0
FECHA_CREA	Fecha	8	0
FILTRO	Caracter	8	0
IMPORTE	Numérico	18	2
IMPORTE	Numérico	19	2
IMPORTE	Numérico	18	2
LINEASXPAG	Numérico	2	0
MAYOR	Numérico	3	0
MES	Numérico	2	0
MESBAJ	Numérico	2	0
MESREG	Numérico	2	0
MES_CAMB	Numérico	2	0
MILES	Lógico	1	0
NIVEL	Numérico	2	0
NIVEL_1	Numérico	2	0
NIVEL_2	Numérico	2	0
NIVEL_3	Numérico	2	0
NOMBRECTA	Caracter	50	0
NOCTA	Lógico	1	0
NOMBRE	Caracter	8	0
NOREC	Lógico	1	0
NOREF	Lógico	1	0
NUMERAPAG	Lógico	1	0
NUMREG	Numérico	6	0
NUMREGA	Numérico	6	0

NOMBRE DEL CAMPO	TIPO	LONG.	DECIM.
NUM_MOV	Númérico	5	0
NUM_PLACA	Carácter	10	0
NUM_REGIST	Carácter	6	0
NUM_SERIE	Carácter	20	0
NUM_SERIEA	Carácter	20	0
OPC1	Lógico	1	0
OPC2	Lógico	1	0
OPC3	Lógico	1	0
OPERADOR	Carácter	2	0
ORDEN	Carácter	12	0
PASSWORD	Carácter	10	0
PERTENECEA	Carácter	20	0
PROTEGIDO	Lógico	1	0
REFERENCIA	Carácter	3	0
REFE_ANTER	Carácter	3	0
REPORTE	Carácter	8	0
RESPONSAB	Carácter	50	0
SALTADESP	Lógico	1	0
SALTANTES	Lógico	1	0
SECTOR	Carácter	1	0
SINOMCTA	Lógico	1	0
SINOMREC	Lógico	1	0
SINOMREF	Lógico	1	0
SINUMYCTA	Lógico	1	0
SINUMYREC	Lógico	1	0
SINUMYREF	Lógico	1	0
TEXTO	Carácter	20	0
TIPCAM	Carácter	1	0
TIPO_RECEP	Númérico	1	0
TITPAG1	Carácter	75	0
TITPAG2	Carácter	75	0
TITPAG3	Carácter	75	0
TOTALCOLUM	Lógico	1	0
UBICACION	Carácter	50	0

## APÉNDICE B1 RELACIÓN DE PROCEDIMIENTOS

- **PROCEDURE: DESPLIEGA\_PANTALLA**

Llamado por:           BORR\_DAT (Procedure en RUTINAS.PRG)  
                          ALTAS.PRG

Descripción : Despliega en la pantalla las etiquetas de los datos a visualizar en altas, bajas o cambios.

- **PROCEDURE: DESPL\_CARA**

Llamado por:           MENU.PRG  
                          ALTAS.PRG  
                          BAJAS.PRG

Descripción: Dibuja la carátula de la pantalla de captura. Rellena la pantalla con el caracter utilizando la función REPLICATE y después dibuja marcos en la pantalla con las funciones PANEL y DOUBLE. Finalmente pone los encabezados de la pantalla y habilita el reloj.

- **PROCEDURE: RECUPERA\_DATOS**

Llamado por:           DATOS\_ADICIONAL(PROCEDURE en RUTINAS.PRG)  
                          MODI\_DAT (PROCEDURE en RUTINAS.PRG)  
                          BORR\_DAT (PROCEDURE en RUTINAS.PRG)

Descripción: Mueve los campos del archivo maestro a variables de trabajo. El archivo maestro debe ser seleccionado previamente.

- **PROCEDURE: INICIA\_DATOS**

Llamado por:           ALTAS.PRG  
                          BAJAS.PRG  
                          CAMBIOS.PRG

Descripción: Inicializa las variables que sirven como puente de captura de los datos del Archivo *Maestro*.

- **PROCEDURE: GUARDA\_DATOS**

Llamado por:           DATOS\_ADICIONAL(PROCEDURE en RUTINAS.PRG)  
                          ALTAS.PRG

Descripción: Reemplaza los datos de un registro del Archivo *Maestro* con las variables de trabajo.

- **PROCEDURE: DESPL\_DATOS**

Llamado por:           BORR\_DAT (PROCEDURE en RUTINAS.PRG)

Descripción: Despliega en pantalla el contenido de las variables de trabajo que sirven como puente en la captura.

- **PROCEDURE: CAPT\_LLAVE**

Llamado por:           BAJAS.PRG  
                  CAMBIOS.PRG  
Llama a:             V\_MAYOR() (function en RUTINAS.PRG)  
                  V\_NIV1() (function en RUTINAS.PRG)  
                  V\_NIV2() (function en RUTINAS.PRG)  
                  V\_NIV3() (function en RUTINAS.PRG)

Descripción: Proporciona la mascarilla en pantalla para especificar las características de los artículos a dar de baja o a modificar. Llama a funciones de validación para ayudar en el llenado de los campos que definen la clave del artículo (*Mayor, Nivel\_1, Nivel\_2, Nivel\_3*) con la tecla **F1**.

- **PROCEDURE: DESP\_LLAVE**

Llamado por:           MODI\_DAT (PROCEDURE en RUTINAS.PRG)

Descripción: Despliega en pantalla los datos no modificables de un registro seleccionado para cambios.

- **PROCEDURE: V\_GRUPO**

Llamado por:           V\_NIV2() (function en RUTINAS.PRG)  
                  V\_NIV3() (function en RUTINAS.PRG)  
Llama a:             NIVEL\_AFECTACION(PROCEDURE en RUTINAS)

Descripción: Procedimiento que valida la clave del artículo. Debe existir en el catálogo y debe ser un nivel afectable, lo cual se checa con el procedimiento *Nivel\_Afectacion*. De no existir, se da la opción de adicionarlo al catálogo con el procedimiento *Adj\_Grupo*.

- **PROCEDURE: NIVEL\_AFECTACION**

Llamado por:           V\_GRUPO (PROCEDURE en RUTINAS.PRG)  
Llama a:             DETERMINA\_NIVEL(PROCEDURE en RUTINAS.PRG)

Descripción: Procedimiento que determina si una clave de artículo puede ser afectable, es decir, si puede contener registros en el archivo *Maestro*.

- **PROCEDURE: DETERMINA\_NIVEL**

Llamado por:           NIVEL\_AFECTACION(PROCEDURE en RUTINAS.PRG)  
                  CHECA\_NIVEL (PROCEDURE en RUTINAS.PRG)

Descripción: Procedimiento que determina el nivel del grupo (1, 2, 3 ó 4) para saber si es afectable o no.

- **PROCEDURE: CHECA\_NIVEL**

Llamado por: ADI\_GRUPO (PROCEDURE en RUTINAS.PRG)  
INS\_GRUPO (PROCEDURE en RUTINAS.PRG)  
Llama a: DETERMINA\_NIVEL (PROCEDURE en RUTINAS.PRG)  
SIGUE (PROCEDURE en MENU.PRG)

Descripción: Procedimiento que checa si una clave de grupo puede ser dada de alta en el catálogo de grupos

- **Function: V\_REFER()**

Llamado por: CAPTURA\_DATOS (PROCEDURE en RUTINAS.PRG)  
DATOS\_ADICIONAL (PROCEDURE en RUTINAS.PRG)  
MODI\_DAT (PROCEDURE en RUTINAS.PRG)  
ALTAS.PRG  
Llama a: ADI\_REFERENCIA (PROCEDURE en RUTINAS.PRG)

Descripción: Función que valida la existencia de una clave de oficina en el catálogo de referencias. De no existir, da la opción de añadirla al catálogo. Además, despliega la descripción de la oficina.

- **Function: V\_RECEP()**

Llamado por: CAPTURA\_DATOS (PROCEDURE en RUTINAS.PRG)  
DATOS\_ADICIONAL (PROCEDURE en RUTINAS.PRG)  
MODI\_DAT (PROCEDURE en RUTINAS.PRG)  
ALTAS.PRG  
Llama a: ADI\_RECEPTOR (PROCEDURE en RUTINAS.PRG)

Descripción: Función que valida la existencia en catálogo de la clave de tipo de receptor para el artículo que se esté capturando o modificando. Opcionalmente se permite darlo de alta en el catálogo.

- **Function: V\_NUMSE()**

Llamado por: CAPTURA\_DATOS (PROCEDURE en RUTINAS.PRG)  
DATOS\_ADICIONAL (PROCEDURE en RUTINAS.PRG)  
MODI\_DAT (PROCEDURE en RUTINAS.PRG)

Descripción: Función booleana que determina si el artículo que se está capturando o modificando debe llevar número de serie, esto en base al campo *OPC1* del catálogo de grupos.

- **Function: V\_PERTE()**

Llamado por: CAPTURA\_DATOS (PROCEDURE in RUTINAS.PRG)  
DATOS\_ADICIONAL (PROCEDURE in RUTINAS.PRG)  
MODI\_DAT (PROCEDURE in RUTINAS.PRG)

Descripción: Función booleana que determina si el artículo que se está capturando o modificando debe llevar número de serie del artículo al que pertenece, esto en base al campo *OPC3* del catálogo de grupos.

- **Function: V\_PASSW()**

Llamado por: MENU.PRG  
ADI\_GRUPO (PROCEDURE en RUTINAS.PRG)  
ADI\_REFERENCIA (PROCEDURE en RUTINAS.PRG)  
ADI\_RECEPTOR (PROCEDURE en RUTINAS.PRG)

Descripción: Función booleana que pide el password de entrada al sistema, o de acceso a la modificación de catálogos a través de una ventana. Checa que el password digitado se encuentre registrado en el archivo de passwords.

- **PROCEDURE: ADI\_GRUPO**

Llamado por: V\_GRUPO (PROCEDURE en RUTINAS.PRG)  
V\_MAYOR() (function en RUTINAS.PRG)  
V\_NIV1() (function en RUTINAS.PRG)  
V\_NIV2() (function en RUTINAS.PRG)  
V\_NIV3() (function en RUTINAS.PRG)  
Llama a: CHECA\_NIVEL (PROCEDURE en RUTINAS.PRG)

Descripción: Este procedimiento se utiliza para registrar un nuevo grupo de activos, solicitando los niveles, la descripción y la pertenencia a otro grupo de artículos, este proceso activa la función explicada anteriormente, la cual verifica el password antes de realizar cualquier operación.

- **PROCEDURE: ADI\_REFERENCIA**

Llamado por: V\_REFER() (function in RUTINAS.PRG)  
Llama a: V\_PASSW() (function in RUTINAS.PRG)

Descripción: Este procedimiento se utiliza para registrar o adicionar una nueva referencia para los grupos de activos, solicitando el número de la referencia, su descripción y el sector.

- **PROCEDURE: ADI\_RECEPTOR**

Llamado por: V\_RECEP() (function in RUTINAS.PRG)  
Llama a: V\_PASSW() (function in RUTINAS.PRG)  
STR() (function in ?)

Descripción: Este procedimiento se utiliza para registrar o adicionar un nuevo tipo de receptor, solicitando el tipo de receptor y su descripción.

- **PROCEDURE: TPOPUP\_ALTAS**

Llamado por: ALTAS.PRG  
BAJAS.PRG

Descripción: Este es un procedimiento de tipo booleano, el cual sólo se encarga de verificar si se realizará una captura individual o por factura, asignando a una variable el valor F ó T.

- **PROCEDURE: TITULO**

Llamado por: IMPR\_BITACORA (PROCEDURE in RUTINAS.PRG)  
REP\_CAPT.PRG  
Llama a: CHR() (function in ?)  
REPLICATE() (function in ?)

Descripción: Este es activado al realizarse una impresión de algún movimiento de activos, basicamente genera la carátula del reporte, con letras normales y comprimidas.

- **PROCEDURE: IMPR\_BITACORA**

Llamado por: DATOS\_ADICIONAL (PROCEDURE in RUTINAS.PRG)  
MODI\_DAT (PROCEDURE in RUTINAS.PRG)  
BORR\_DAT (PROCEDURE in RUTINAS.PRG)  
ALTAS.PRG  
Llama a: TITULO (PROCEDURE in RUTINAS.PRG)

Descripción: Este es activado al realizarse una impresión de algún movimiento de activos, y es el complemento del proceso explicado anteriormente, éste genera los datos que se imprimen en el cuerpo del reporte.

- **PROCEDURE: CAPTURA\_DATOS**

Llamado por: ALTAS.PRG  
Llama a: V\_ANIO() (function in RUTINAS.PRG)  
V\_MES() (function in RUTINAS.PRG)  
ACT\_POPS (PROCEDURE in RUTINAS.PRG)  
V\_MAYOR() (function in RUTINAS.PRG)  
V\_NIV1() (function in RUTINAS.PRG)  
V\_NIV2() (function in RUTINAS.PRG)  
V\_NIV3() (function in RUTINAS.PRG)  
V\_NUMSE() (function in RUTINAS.PRG)  
V\_PERTE() (function in RUTINAS.PRG)  
V\_IMPORTE() (function in RUTINAS.PRG)

Descripción: Este proceso es un de los más importantes en el sistema, su tarea consiste en leer todos los datos de cada artículo que ingresa al activo fijo de la institución, hay que aclarar que no solo lee y capta los datos, sino que también valida el contenido de cada campo, con la finalidad de checar que sólo existan datos correctos.

**APÉNDICE C1.  
LISTADOS Y CORRIDAS DEL SISTEMA**



E A N C O S S I X T T T T  
 SISTEMA DE CONSULTA Y EXPLOTACION DEL MODELO Y EQUIPO  
 CATALOGO DE SP-RTE      FECHAS: 02/07/87  
 MODAL 1

MAJOR	NIVEL 1	NIVEL 2	NIVEL 3	DESCRIPCION	NumSerie?	Factorize??	Orden Text?
0	0	0	0		F	F	F
1	0	0	0	TRANSPORTE AEREO	F	F	F
1	0	0	0	AVION	F	F	T
1	1	1	0	DOUGLAS DC-9	T	F	F
2	0	0	0	TRANSPORTE TERRESTRE	F	F	F
2	1	0	0	AUTOMOVIL	F	F	T
2	1	1	0	CHEVLEK	F	F	F
2	1	1	3	DART 88	T	F	F
2	1	1	4	DART 89	T	F	F
2	1	1	5	DART K 84	T	F	F
2	1	1	7	NEW YORKER 91	T	F	F
2	1	1	8	RAM CHARGER 91	T	F	F
2	1	1	9	SPIRIT 90	T	F	F
2	1	1	10	SPIRIT 91	T	F	F
2	1	1	12	RAM CHARGER 92	T	F	F
2	1	1	13	SPIRIT 92	T	F	F
2	1	1	14	SHADOW 89	T	F	F
2	1	1	15	SHADOW 90	T	F	F
2	1	1	16	SPIRIT 93	T	F	F
2	1	1	17	DART 81	T	F	F
2	1	1	18	DART 85	T	F	T
2	1	1	19	RAM CHARGER 94	T	F	F
2	1	1	20	SPIRIT 94	T	F	F
2	1	1	21	SPIRIT 95	T	F	F
2	1	1	22	LE BARDON 95	F	F	F
2	1	2	0	FORD	F	F	F
2	1	2	1	FAIRMONT 82	T	F	F
2	1	2	2	FAIRMONT COUNTRY 81	T	F	F
2	1	2	3	GRAND MARQUIS 82	T	F	F
2	1	2	4	TOPAI 91	T	F	F
2	1	2	5	TOPAI 92	T	F	F
2	1	2	6	GRAND MARQUIS 94	T	F	F
2	1	2	7	MERCURY SABLE 95	T	F	F
2	1	2	8	ESCORT 95	T	F	F
2	1	2	9	MYSTIQUE 95	T	F	F
2	1	3	0	GENERAL MOTORS	F	F	F
2	1	3	2	CELEBRITY 89	T	F	F
2	1	3	3	CENTURY 91	T	F	F
2	1	3	4	CUTLASS 88	T	F	F
2	1	3	5	CUTLASS 89	T	F	F
2	1	3	6	CUTLASS 91	T	F	F
2	1	3	7	MALIBU 81	T	F	F
2	1	3	8	CUTLASS 92	T	F	F
2	1	3	9	CAVALIER 92	T	F	F
2	1	3	10	CENTURY LIMITED 93	T	F	F
2	1	3	11	CAVALIER 93	T	F	F
2	1	3	12	CUTLASS 90	T	F	F
2	1	3	13	MALIBU 80	T	F	F
2	1	3	14	SCURBEAM 95	T	F	F
2	1	3	15	CENTURY 95	T	F	F
2	1	3	16	CAVALIER 95	T	F	F
2	1	4	0	NISSAN	F	F	F
2	1	4	1	ICHI-VAN 90	T	F	F
2	1	4	2	ICHI-VAN 91	T	F	F
2	1	4	3	TSURU II 89	T	F	F

SISTEMA DE CONSULTA Y EXPLORACION DEL CODIGO DE CLASIFICACION  
CATALOGO DE REFERENCIAS

REF	DESCRIPCION
L95	CONTROL ADMVO. DE OTROS SERVICIOS AL PERSONAL
L96	SERVICIO MEDICO EXTERNO
L97	CONSULTORIO CENTRAL
L98	COMEDORES
L99	MOBILIARIO A CARGO DE SU DIRECCION-OFICINAS NO IDE
M01	DIRECCION DE CONTRALORIA
M03	SUBGERENCIA DE INFORMATICA
M04	DESARROLLO Y MANTENIMIENTO DE SISTEMAS.- D.C.
M05	OPERACION DE SISTEMAS.- D.C.
M10	GERENCIA DE PRESUPUESTOS Y CONTABILIDAD
M11	SUBGCCIA. DE PRESUPUESTOS E INFORMACION FINANCIERA
M13	PRESUPUESTOS
M14	INFORMACION FINANCIERA
M21	SUBGERENCIA DE CONTABILIDAD
M23	CONTADURIA
M24	CONTABILIDAD DE CAPITAL Y FIDEICOMISOS EN ADMON.
M25	CONTABILIDAD DE COSTOS
M30	GERENCIA DE CONTROL
M31	SUBGERENCIA DE REGULACION ADMINISTRATIVA
M33	INSTRUMENTACION ADMINISTRATIVA
M34	SECRETARIADO TECNICO DE COMITES
M35	VERIFICACION DE ADQUISICIONES DE OBRA PUBLICA
M41	SUBGERENCIA DE CONTROL DE EGRESOS ADMINISTRATIVOS
M43	CONTROL DE EGRESOS ADMINISTRATIVOS
M44	CONTROL DE NOMINA Y SEGUROS
M51	SUBG. DE CONTROL DE EFECTIVOS, METALES Y VALORES
M53	CONTROL DE EFECTIVOS, METALES Y VALORES
M54	CONTRDL DE FABRICACION DE BILLETES
M60	GERENCIA DE AUDITORIA
M63	AUDITORIA DE SISTEMAS
M71	SUBGERENCIA DE AUDITORIA DE OPER.BANCARIAS
M73	AUDTORIA DE OPER. INTERN. ORG. Y ACUERDOS
M74	AUDTORIA DE OPER. DE CRED. INTERNO Y EXTERNO
M81	SUBG. DE AUD. DE OPERC. ADMVAS. Y FIDEICOMISOS
M83	AUDITORIA DE OPER. DE ADMON. E INMUEBLES
M84	AUDITORIA DE FIDEICOMISOS Y SUCURSALES
M99	MOBILIARIO A CARGO DE SU DIRECCION-OFICINAS NO IDE
N04	ESPECIALES CON SALDO
P00	PENSIONES OFICINA CENTRAL
R01	DIRECCION DE SISTEMAS
R02	SUBGERENCIA DE ORGANIZACION
R03	REGULACION ORGANIZACIONAL
R10	GERENCIA DE INFORMATICA
R11	SUBGERENCIA DE SISTEMATIZACION
R13	REGULACION Y DESARROLLO DE INFORMATICA
R14	INGENIERIA DE SISTEMAS
R21	SUBGERENCIA DE COMPUTO Y TELECOMUNICACIONES
R23	CENTRO DE SERVICIOS DE COMPUTO
R24	TELECOMUNICACIONES
R25	SERVICIO DE COMUNICACION TELEFONICA
R33	GESTION ADMINISTRATIVA DE INFORMATICA
R34	PLANEACION INFORMATICA
R99	MOBILIARIO A CARGO DE SU DIRECCION-OFICINAS NO IDE
RA0	ALMACEN DISTRIBUCION
RA1	ALMACEN

BANCO CENTRAL  
 SISTEMA DE CONSULTA Y EXPLOTACION DEL REGISTRO  
 CATALOGO DE REFERENCIAS

P. 01 5

REF	DESCRIPCION
R7C	TELECOMUNICACIONES DELEG. CENTRO
RTD	TELECOMUNICACIONES DISTRIBUCION
RTL	TELECOMUNICACIONES DELEG. LEGARIA
S10	GERENCIA DE SEGURIDAD E INVESTIGACION
S12	GESTION ADMINISTRATIVA DE SEGURIDAD E INVESTIGAC.
S13	UNIDAD DE PROTECCION
S22	SUBGERENCIA DE INVESTIGACION
S32	SUBGERENCIA DE SEGURIDAD
S99	MOBILIARIO A CARGO DE SU DIRECCION-OFICINAS NO ICE
SB1	PREVENCION E INVESTIGACION
SB2	DOCUMENTOS COPIA Y FOTOGRAFIA
SB3	LABORATORIO QUIMICO
SC1	INSTRUCTORES DE SEGURIDAD
SC2	SISTEMAS DE SEGURIDAD
SC3	PROTECCION A INMUEBLES Y VALDRES
SC4	PRDTECCION A INMUEBLES Y VALORES
SD4	ALMACENES - SEGURIDAD
TD1	DIRECCION DE DISPOSICIONES DE BANCA CENTRAL
TD2	SUBGERENCIA DE ASUNTOS JURIDICOS ESPECIALES
TD3	SUBGERENCIA TECNICA DDEC
T10	GERENCIA DE DISPOSICIONES AL SISTEMA FINANCIERO
T11	SUBGERENCIA DE DISPOSICIONES AL SISTEMA FINANCIERO
T13	SUBGERENCIA DE INSTRUMENTACION LEGAL DE OP. DE B.C.
T16	COMPILACION DE DISPOSICIONES
T21	SUBGERENCIA DE DISPOS. A INTERMEDIARIOS NO BANCAR.
T3D	GERENCIA DE AUTORIZACIONES Y ASUNTOS JURIDICOS INT
T3I	SUBGERENCIA DE AUTORIZACIONES Y CONSULTAS DE B.C.
T33	SUBGERENCIA DE ASUNTOS JURIDICOS INTERNACIONALES
T40	GERENCIA DE INVESTIGACION JURIDICA DE BANCA CENT.
T50	COORDINACION DE DISPOSICIONES DE BANCA CENTRAL
T5I	SUBGERENCIA DE ASUNTOS JURIDICOS ESPECIALES
T99	MOBILIARIO A CARGO DE SU DIRECCION-OFICINAS NO IDE
W01	CAJA-TESORERIA
Z01	PERSONAL EN TRAMITE DE ADSCRIPCION
Z99	ACTIVO FIJO SIN UBICACION CONGCI0A

TOTAL DE REGISTROS : 475

FECH. DE ACEL. D.	DESCRIPCION DEL ARTICULO	REF	NUM. DE SERIE	IMPORTE	SUMA POR MES	SUMA POR AÑO
BAJAS DEL GRUPO DE TRANSPORTE TERRESTRE						
8/89	3 3 1 2 - CAMIONES - CHRYSLER - PICK-UP 89	L65	136205	26.047.72		
TOTAL MES 8/89						26.047.72
8/89	2 1 5 11 - AUTOMOVIL - VOLKSWAGEN - COMBI 89	L66	20F0085114	27.251.72		
TOTAL MES 8/89						27.251.72
12/89	2 1 3 5 - AUTOMOVIL - GENERAL MOTORS - CUTLASS 89	L86	305A-5472LS1019A7	42.521.74		
12/89	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-247943M	18.371.25		
12/89	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-250328M	18.371.25		
12/89	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-250824M	18.371.25		
TOTAL MES 12/89						97.635.49
TOTAL AÑO 89						144.914.73
6/90	2 1 4 1 - AUTOMOVIL - NISSAN - ICHI-YAN 90	L86	PR0-DE0285	26.247.63		
6/90	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-257322M	21.295.65		
6/90	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-257802M	21.295.65		
6/90	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-268074M	21.295.65		
6/90	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-268222M	21.295.65		
TOTAL MES 6/90						112.430.43
7/90	2 1 1 15 - AUTOMOVIL - CHRYSLER - SHADOW 90	L86	LT-048031	27.259.13		
TOTAL MES 7/90						27.259.13
8/90	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-300426M	22.145.22		
8/90	2 1 4 4 - AUTOMOVIL - NISSAN - TSURU 90	L86	E16-301922M	22.145.22		
TOTAL MES 8/90						44.330.44
TOTAL AÑO 90						185.020.00
4/91	2 1 3 4 - AUTOMOVIL - GENERAL MOTORS - CUTLASS 91	L86	MS-121356	48.235.29		
4/91	2 1 3 6 - AUTOMOVIL - GENERAL MOTORS - CUTLASS 91	L86	MS-123926	48.235.29		
TOTAL MES 4/91						96.471.78
TOTAL AÑO 91						96.471.78
TOTAL DEL GRUPO DE TRANSPORTE TERRESTRE						426.476.71

PARQUE DE RESERVA  
 DIRECCION DE CONTABILIDAD  
 CONTABILIDAD DE CAPITAL Y FIDEICOMISOS

REPORTE DE CAMBIOS DE MULTIPLOS Y EQUIPO  
 DEL MES DE JULIO DE 1995

ANALISIS	CONCEPTO	NUM. DE CUOTE	IMPORTE	IMP
2/94	2 1 5 15 TRANSPORTE TERRESTRE - AUTOMOVIL - VOLKSWAGEN - JETTA 95		50.687.00	184
2/94	2 1 5 15 TRANSPORTE TERRESTRE - AUTOMOVIL - VOLKSWAGEN - JETTA 95	32416711111111111111	50.687.00	184
<hr/>				
2/94	2 1 5 15 TRANSPORTE TERRESTRE - AUTOMOVIL - VOLKSWAGEN - JETTA 95		50.687.00	184
2/94	2 1 5 15 TRANSPORTE TERRESTRE - AUTOMOVIL - VOLKSWAGEN - JETTA 95	32416711111111111111	50.687.00	184
<hr/>				
TOTAL GRUPO			101.374.00	
12/94	13 2 12 36 EQUIPO DE COMPUTO ELECTRONICO - EQUIPO DE MICROCOMPUTACION - DIVERSO - UNIDAD LECTORA DE CD	4660408914	1.730.29	833
12/94	13 2 12 36 EQUIPO DE COMPUTO ELECTRONICO - EQUIPO DE MICROCOMPUTACION - DIVERSO - UNIDAD LECTORA DE CD	4660408914	2.016.00	833
<hr/>				
12/94	13 2 12 36 EQUIPO DE COMPUTO ELECTRONICO - EQUIPO DE MICROCOMPUTACION - DIVERSO - UNIDAD LECTORA DE CD	4360408911	2.016.00	833
12/94	13 2 12 36 EQUIPO DE COMPUTO ELECTRONICO - EQUIPO DE MICROCOMPUTACION - DIVERSO - UNIDAD LECTORA DE CD	4360408911	2.016.00	833
<hr/>				
TOTAL GRUPO			4.032.00	
12/94	13 6 2 15 EQUIPO DE COMPUTO ELECTRONICO - PROGRAMA O SOFTWARE - SOFTWARE DIVERSO - SFT COREL SC51		14.430.00	833
12/94	13 6 2 15 EQUIPO DE COMPUTO ELECTRONICO - PROGRAMA O SOFTWARE - SOFTWARE DIVERSO - SFT COREL SC51		1.443.00	833
<hr/>				
TOTAL GRUPO			1.443.00	
<hr/>				
TOTAL			106.849.00	

E A N C O I I X X X I X  
 DIRECCION DE CONTRALORIA  
 CONTABILIDAD DE CAPITAL Y FIDEICOMISOS  
 EN ADMINISTRACION

SISTEMA DE CONTROL FISICO DE BIENES MUEBLES  
 DISTRIBUCION DE BIENES REGISTRADO EN EL MES DE JULIO DE 1995 HORA 1  
 11/10/95

UNIDAD DISTRIBUIDORA	FECHA DE ASUJUE.	CARACTERISTICAS DEL BIEN MUEBLE			IMPORTE	DEPENDENCIA RECEPTORA
		DESCRIPCION DEL BIEN	NUM. DE SERIE			
	CUENTA	GRUPO DE TRANSPORTE	GRUPO TRANSPORTE TERRESTRE			
184 GESTION ADMINISTRATIVA SERVICIOS A LA INSTITU S/94	2 1 5	15 AUTOMOVIL - VOLKSWAGEN - JETTA 95			50.687.00	REC GESTION ADMINISTRATIVA DE INFORMACION
	TOTAL GRUPO	2 TRANSPORTE TERRESTRE			50.687.00	
	TOTAL CUENTA	GRUPO DE TRANSPORTE			50.687.00	
TOTAL GESTION ADMINISTRATIVA SERVICIOS A LA INSTITUCION TRASPASO DE	1	ARTICULOS CON IMPORTE DE			50.687.00	
	CUENTA	GRUPO EQUIPO DE COMPUTO ELECTRONICO	GRUPO EQUIPO DE COMPUTO ELECTRONICO			
REC OPERACION DE SISTEMAS.- D.C.	9/93 13 2 1 5	EQUIPO DE MICROCOMPUTACION - UNIDAD BASE DE PROCESAMIENTO - OLIVETTI M300 B3300	2902977		4.479.04	REC SUBGERENCIA DE INFORMATICA
	9/93 13 2 1 5	EQUIPO DE MICROCOMPUTACION - UNIDAD BASE DE PROCESAMIENTO - OLIVETTI M300 B3300	2904018		6.640.09	M54 DESARROLLO Y MANTENIMIENTO DE SISTEMAS.- D.C.
	9/93 13 2 1 5	EQUIPO DE MICROCOMPUTACION - UNIDAD BASE DE PROCESAMIENTO - OLIVETTI M300 B3300	2904087		6.640.09	M54 DESARROLLO Y MANTENIMIENTO DE SISTEMAS.- D.C.
	9/93 13 2 1 5	EQUIPO DE MICROCOMPUTACION - UNIDAD BASE DE PROCESAMIENTO - OLIVETTI M300 B3300	2904095		4.479.04	M54 DESARROLLO Y MANTENIMIENTO DE SISTEMAS.- D.C.
	9/93 13 2 2 8	EQUIPO DE MICROCOMPUTACION - TECLADO Q KEYCARD - OLIVETTI ESTANDARD 27-102	6783627		200.00	REC SUBGERENCIA DE INFORMATICA
	9/93 13 2 2 8	EQUIPO DE MICROCOMPUTACION - TECLADO Q KEYCARD - OLIVETTI ESTANDARD 27-102	6782656		200.00	M54 DESARROLLO Y MANTENIMIENTO DE SISTEMAS.- D.C.
	9/93 13 2 2 8	EQUIPO DE MICROCOMPUTACION - TECLADO Q KEYCARD - OLIVETTI ESTANDARD 27-102	6782673		200.00	REC GESTION ADMINISTRATIVA DE INFORMATICA
	9/93 13 2 2 8	EQUIPO DE MICROCOMPUTACION - TECLADO Q KEYCARD - OLIVETTI ESTANDARD 27-102	6782682		200.00	M54 DESARROLLO Y MANTENIMIENTO DE SISTEMAS.- D.C.
	9/93 13 2 3 3	EQUIPO DE MICROCOMPUTACION - MONITOR, VIDEO, DISPL AY, ETC. - MONITOR POLICROMATICO 14"	2678274		1.200.00	REC SUBGERENCIA DE INFORMATICA
	9/93 13 2 3 3	EQUIPO DE MICROCOMPUTACION - MONITOR, VIDEO, DISPL AY, ETC. - MONITOR POLICROMATICO 14"	2678330		1.200.00	REC GESTION ADMINISTRATIVA DE INFORMATICA
	9/93 13 2 3 3	EQUIPO DE MICROCOMPUTACION - MONITOR, VIDEO, DISPL AY, ETC. - MONITOR POLICROMATICO 14"	2678331		1.200.00	M54 DESARROLLO Y MANTENIMIENTO DE SISTEMAS.- D.C.
	9/93 13 2 3 3	EQUIPO DE MICROCOMPUTACION - MONITOR, VIDEO, DISPL AY, ETC. - MONITOR POLICROMATICO 14"	2678347		1.200.00	M54 DESARROLLO Y MANTENIMIENTO DE SISTEMAS.- D.C.
	TOTAL GRUPO	13 EQUIPO DE COMPUTO ELECTRONICO			27.838.26	
	TOTAL CUENTA	EQUIPO DE COMPUTO ELECTRONICO			27.838.26	
TOTAL OPERACION DE SISTEMAS.- D.C. TRASPASO DE	12	ARTICULOS CON IMPORTE DE			27.838.26	
TOTAL GENERAL TRASPASO DE	13	ARTICULOS CON IMPORTE DE			78.525.26	



LISTA DE EQUIPOS  
 REPORTE DE MOVILIDAD Y EQUIPO POR DEPENDENCIA

FECHA REC. DESCRIPCION	NUM. PLACA	NUM. DE SERIE	DEPENDENCIA	MONTAJO	TIPO DE REPOSICION
12 3 2 12 91/4		80103	TOTAL CPU 3	10.722,85	DEFINITIVO
			TOTAL CPU 12	729,52	
12 1 2 6 90/10	01919	032375		1.213,04	DEFINITIVO
13 1 2 6 90/10	1921	144265		1.213,04	DEFINITIVO
13 1 2 6 90/10	05048	344380		1.213,04	DEFINITIVO
13 1 2 6 90/10	05043	113782		1.213,04	DEFINITIVO
13 1 2 10 90/2	01950	0128379		7.051,48	DEFINITIVO
13 2 1 5 93/9		2503977		4.479,04	DEFINITIVO
13 2 1 5 93/9		2954018		4.479,04	DEFINITIVO
13 2 1 5 93/9		3904087		4.479,04	DEFINITIVO
13 2 1 5 93/9		2904090		4.479,04	DEFINITIVO
13 2 1 5 93/9		3904123		4.479,04	DEFINITIVO
13 2 1 5 93/9		2504170		4.479,04	DEFINITIVO
13 2 2 8 93/9		6782627		200,00	DEFINITIVO
13 2 2 8 93/9		6782656		200,00	DEFINITIVO
13 2 2 8 93/9		6782673		200,00	DEFINITIVO
13 2 2 8 93/9		6782682		200,00	DEFINITIVO
13 2 2 8 93/9		6782699		200,00	DEFINITIVO
13 2 2 8 93/9		6782942		200,00	DEFINITIVO
13 2 2 10 91/7		C-18010542		209,74	DEFINITIVO
13 2 3 3 93/9		2675274		1.200,00	DEFINITIVO
13 2 3 3 93/9		2675310		1.200,00	DEFINITIVO
13 2 3 3 93/9		2675331		1.200,00	DEFINITIVO
13 2 3 3 93/9		2675347		1.200,00	DEFINITIVO
13 2 3 3 93/9		2675382		1.200,00	DEFINITIVO
13 2 3 3 93/9		2675387		1.200,00	DEFINITIVO
13 2 4 1 90/10		01932		1.200,00	DEFINITIVO
13 2 4 1 92/5		0611114538		2.184,87	DEFINITIVO
13 2 4 1 92/5		0611372452		1.822,00	DEFINITIVO
13 2 4 12 90/2		0588098290		2.397,54	DEFINITIVO
13 2 4 12 92/5		2943167166		5.443,75	DEFINITIVO
13 2 4 13 90/6	02004			17.849,16	DEFINITIVO
13 2 4 13 91/7				17.024,25	DEFINITIVO
13 2 4 18 91/6	01949	3025A40562		5.629,20	DEFINITIVO
13 2 4 20 93/9	10461	004812148		1.340,00	DEFINITIVO
13 2 7 3 90/8		37305667001		14.521,24	DEFINITIVO
13 2 9 5 79/12		29125062	3047201	1,00	DEFINITIVO
13 2 12 1 79/12		0046134		1,00	DEFINITIVO
13 2 12 1 79/12		0546344		1,00	DEFINITIVO
13 2 12 24 79/12		619		1,00	DEFINITIVO
13 3 1 17 79/12				1,00	DEFINITIVO
13 3 1 17 90/6				643,48	DEFINITIVO
13 3 1 17 90/6				643,48	DEFINITIVO
13 3 1 17 90/6				643,48	DEFINITIVO
13 3 1 17 90/6				643,48	DEFINITIVO
13 3 1 17 90/6				643,48	DEFINITIVO
13 3 1 17 90/6				643,48	DEFINITIVO
13 3 2 1 91/12	00167	1163681		1.574,89	DEFINITIVO
13 3 2 1 92/7		4805		2.663,91	DEFINITIVO
13 3 4 1 92/9				2.008,28	DEFINITIVO
13 3 4 1 92/9				2.008,28	DEFINITIVO
13 3 5 7 91/1				71,91	DEFINITIVO
13 3 5 7 91/1	07888	4805		71,91	DEFINITIVO



OFICINA DE CONTABILIDAD DE LA FISCALIA  
 HISTORIA DE MOVIMIENTOS DEL FONDO  
 SISTEMA DE CUENTAS DE MANTENIMIENTO DE MOBILIDAD

Movimiento	Año	Mes	Grupo	Nivel_1	Nivel_2	Nivel_3	Numero de serie	Pertenencia	Ref. Traces	Importe	Saldo
* CAMBIO-ANTES *	93	09	013	02	01	05	29039777		M03 1	5 479.00	1
* CAMBIO-DESPUES *	93	09	013	02	01	05	29039777		M03 1	5 479.00	1
* CAMBIO-ANTES *	92	05	013	02	01	21	30473899		M03 1	5 144.00	1
* CAMBIO-DESPUES *	92	05	013	02	01	21	30473899		M03 1	5 144.00	1
*** BAJA ***	95	07	013	02	04	18	45630486		M03 1	5 150.00	1
*** BAJA ***	95	07	013	02	04	18	12350485		M03 1	5 150.00	1

## GLOSARIO DE TÉRMINOS

**ALIAS.** - Renombra temporalmente una tabla.

**ANSI.** Abreviación de "American National Standards Institute". Una organización voluntaria que ayuda a definir estándares, y que también representa a E.U. en la Organización Internacional de Estándares.

**ARCHIVO.** Conjunto de registros semejantes.

**ASCII.** Siglas "American Standard Code for Information Interchange". Forma estándar de codificar los caracteres en un patrón de 7 bits. El ASCII extendido utiliza 8 bits, y logra codificar en vez de 128 patrones , 256.

**ATRIBUTOS.** Valor descriptivo o propiedad asociada a una entidad.

**BASE DE DATOS.** Colección integrada de datos, interrelacionados entre sí. Almacenados sin redundancia perjudiciales para servir a una o más aplicaciones.

**BUGS:** Errores del sistema.

**CAMPO.** Unidad de datos más pequeña que se haya almacenada en la base de datos.

**CANAL DE COMUNICACIÓN.** Cable dedicado a comunicaciones este puede ser: tipo telefónico, par trenzado, coaxial y fibra óptica.

**CCITT.** Siglas del Comité Consultivo Internacional de Telegrafía y Telefonía. Fijas estándares internacionales en comunicaciones está en Génova, Suiza

**CIPS.** Centro de Información Presupuestal.

**CONCURRENCIA.** Son múltiples accesos por diferentes usuarios a la misma información.

**CSMA/CD.** Protocolo que nos sirve para prevenir las colisiones de información.

**DATA BASE MANAGEMENT SYSTEM (DBMS).** Sistema administrador de base de datos: conjunto de herramientas para el manejo, explotación y administración de la información.

**DATO.** Describe hechos o fenómenos sucedidos en el tiempo. Permite la toma de decisiones. Es un recurso básico de una empresa.

**DEPENDENCIA FUNCIONAL.** Es cuando la existencia de un atributo depende directamente de otro que no es llave.

**DETERMINANTE.** Es un atributo o atributos del cual depende funcionalmente en forma completa de otro atributo.

**DICCIONARIO DE DATOS (DD).** Contiene información de la información. Es decir contiene una descripción de la base de datos. Ayuda a identificar y clasificar la información, son tablas con descripción de los archivos, registros, programas y campos. Permite reconocer: sinónimos, referencias, alias, etc. para los usuarios.

**DMA.** Siglas de Direct Memory Acces. Método por el cual el procesador se libera de atender a cada byte que se transmite en un dispositivo o programa y la memoria, por lo cual la transmisión se hace sin atención, tan sólo para inicializarla y terminarla.

**EFICIENCIA (PERFORMANCE).** Es el tiempo de respuesta de una aplicación en el momento de acceder a una base de datos

**ENTIDAD.** Es un objeto que contiene ciertos atributos que lo caracterizan. Elemento básico que se desea administrar, guardándose información del mismo.

**ESQUEMA.** Es la descripción lógica global de la base de datos. Se encuentra en un diccionario de datos.

**ESTACIÓN DE TRABAJO.** Están representadas por cada una de las computadoras que están conectadas a la red, pueden ser PCs.

**HOMÓNIMOS.-** Cuando dos o más atributos de relaciones diferentes tienen el mismo nombre.

**IADE ó IMPAC.** Impuesto al activo de las empresas.

**IEEE.** Institute of Electrical and Electronics Engineers, es el instituto de profesionistas de más relevancia en el mundo. El comité 802 de IEEE ha definido diversos estándares para redes locales.

**INCONSISTENCIA.** Obtener diferentes salidas para peticiones similares en un mismo momento.

**INTEGRACIÓN.** Es la unión de muchos archivos separados.

**INTEGRIDAD DE ENTIDADES.** Ningún componente de la llave primaria PK puede tener valores nulos.

**INTEGRIDAD REFERENCIAL.** Si A es un atributo primario entonces cada valor del atributo A deberá ser nulo y corresponder a un valor de la llave primaria relacionada.

**INTEGRIDAD.** Reglas que los datos deben cumplir dictadas por el mundo real.

**ISR.** Impuesto sobre la renta.

**LLAVE (KEY o K).** Es un atributo o conjunto de atributos que hacen distinguible a una entidad dentro de un conjunto de entidades o a una relación dentro de un conjunto de relaciones.

**LLAVE CANDIDATO.** Es aquella que cumple con dos propiedades: a) unicidad (única) y b) minimización

**LLAVE FORÁNEA (FK ).** Conjunto de uno o más atributos que es llave primaria de otra relación o tabla, que sirve de acceso o conexión con otra relación o tabla

**LLAVE PRIMARIA (PK).** Es una llave candidato elegida de manera arbitraria como identificador de una tabla.

**LOCKING.** Tarea de control la concurrencia mediante el bloqueo de ciertos bytes de información, usualmente de un archivo o registro (file locking o record locking).

**LOGIN.** Acción de entrar a utilizar host o un server de una red. Establece la sesión y ser reconocido como usuario.

**MCE (Módulo de Consulta y Explotación).** Archivo que contendrá la relación de todos los reportes generados.

**MICROSOFT.** La empresa más importante de software para microcomputadoras. Creadora entre otros productos de: MS-DOS, OS-2, LAN MANAGER, EXCEL, SQL SERVER etc.

**MODELO DE DATOS.** Herramientas conceptuales para describir los datos, sus relaciones, sus semánticas y sus limitantes. Permiten interpretar en una manera abstracta fenómenos del mundo real.

**MODELO ENTIDAD-RELACIÓN (MER).** Permite diseñar un fenómeno real para un sistema de base de datos a través de diagramas con entidades y sus relaciones.

**MODELO LÓGICO BASADO EN REGISTROS.** Nos proporciona un panorama de la estructura general de la base de datos. Es el modelado que se acerca más a la implementación conceptual de la base de datos.

**MODELO RELACIONAL.** Base de datos constituida por un conjunto de archivos (tablas) y relaciones entre ellos.

**MODELO RETICULAR O DE RED.** La base de datos es un conjunto de registros ligados formando datos interceptados.

**NETWARE.** Sistema operativo de red desarrollado por Novell Inc. Tiene diversas versiones.

**NETBIOS.** Interfaz estándar (hasta hoy) para comunicar dos estaciones de trabajo en una red local.

**NORMALIZACIÓN.** Reestructuración de las tablas para disminuir la redundancia de la información. Las formas normales son: 1FN: Primera forma normal, 2FN: Segunda forma normal, 3FN: Tercera forma normal, 4FN: Cuarta forma normal, FNBC: Forma normal de Boyce/Codd, 5FN o FN/PR: Forma normal de proyección/reunión.

**NOVELL.** Uno de los principales fabricantes de productos para redes locales. Desde 1988 se ha enfocado predominantemente al mercado de sistemas operativos, desligándose casi totalmente del hardware para redes.

**OSI.** Open System Interconection. Estructura lógica y estándar de 7 protocolos.

**PROGRAMA DE APLICACIÓN.** Programa que manipula el DBMS para extraer datos de la base de datos.

**PROTOCOLO.** Conjunto de reglas convencionales, utilizadas para comunicar dos dispositivos de la misma naturaleza.

**PROTOTIPO.** Estrategia para desarrollar y afinar la definición de requerimientos durante el ciclo de vida de desarrollo del sistema.

**RECUPERACIÓN.** Capacidad de restaurar la integridad y la consistencia de una base de datos después de una falla en el sistema.

**REDES LOCALES.** Conjunto de computadoras enlazadas entre sí y con otros equipos.

**REDUNDANCIA.** Repetición de datos. Datos derivados, genera inconsistencias.

**REGISTRO.** Conjunto de campos almacenados, agrupados y asociados entre sí

**RELACIÓN.** Asociación entre entidades número de ocurrencias que pueden enlazarse de una entidad a otra.

**SEGURIDAD.** Protección de los datos contra accesos, modificaciones ó pérdidas.

**SERVER DEDICADO.** Se usa para administrar los recursos de la red.

**SERVER NO DEDICADO.** Además de administrar los recursos de la red funciona como estación de trabajo.

**SQL.** Del inglés Structure Query Lenguaje. El lenguaje de consulta y acceso a base de datos más común en la actualidad. Definido como estándar por IBM, ANSI e ISO.

**TABLAS.** Arreglo bidimensional o colección de registros. Formado por renglones (registros) y columnas (atributos).

**TARJETA DE INTERFACE.** Estas se encuentran instaladas dentro de cada computadora, según su especificación cada tarjeta determina la forma de conexión, hay tres tipos de tarjetas estas son: ARCNET, ETHERNET, TOKEN RING.

**TERMINADOR.** Componente que se coloca al extremo de un cable coaxial, y que consiste de una resistencia de la misma cantidad de ohms que la impedancia del cable

**TOPOLOGÍA.** Es un sistema de cableado que va a contener la forma de conexión de los equipos.

**VIEWS O VISTAS.** Es una ventana a través de la cual podemos ver o cambiar información de una tabla muy grande.