114
2ej.

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

SISTEMA DE INFORMACION PARA UNA CASA DE CAMBIO DE DIVISAS

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO MECANICO ELECTRICISTA
ELECTRICA-ELECTRONICA
P R E S E N T A
HECTOR MARTINEZ VALENZUELA

1997

TESIS CON
FALLA DE ORIGEN

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

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.

Quiero agradecer y dedicar este trabajo a todas aquellas personas que de alguna manera han contribuido a mi formación humana y profesional.

De manera muy especial:

A mis papás y mis hermanos.

A Irma y a Jimena.

A la memoria de mis abuelos Amalia y Agustín.

Vladimir.

Dedico el presente trabajo de tesis a mis padres Virginia y Emilio.

A mis hermanos Irma, Emilio, Enrique y Mario.

De manera especial al amor de mi vida mi esposa Norma, por todo su cariño, comprensión y apoyo.

A mis hijos Hectorín y Mauricio lo más grande que Dios me ha dado.

A mis familiares y amigos.

Héctor.

INTRODUCCIÓN

A lo largo de la historia, el hombre siempre se ha caracterizado por sostener una lucha constante en aras de eficientar sus métodos de producción a través de la investigación y el desarrollo tecnológico. En el siglo XVIII la humanidad vivió una profunda transformación con la llamada Revolución Industrial que en principio no fue vista con agrado por parte de las clases trabajadoras quienes veían amenazadas sus fuentes de empleo. Sin embargo, ahora sabemos que dicha transformación mejoró el nivel de vida de la gente acortando la jornada de trabajo y dando mayores oportunidades de empleo.

Hoy en día estamos viviendo otra transición, ahora de una sociedad industrial a una sociedad de información y, según los expertos, esta nueva etapa reducirá nuevamente el número de trabajadores para llevar a cabo tareas especializadas, pero los beneficios de la todavía naciente sociedad de información son muchos y muy claros para la mayoría de las personas.

La fuerza motriz que está detrás de esta transición es sin duda la computadora.

Actualmente casi cualquier empresa está consciente de la importancia de contar con sistemas de información basados en computadoras e incluso, la mayoría de ellas afirman que su éxito se debe a dichos sistemas, no sólo por el hecho de que les permiten controlar la información generada por la misma operación sino, por el apoyo que les brindan en la toma de decisiones.

Ubicándonos en el contexto económico, social y político por el que atraviesa el país y ante la apertura económica y el libre comercio que se comienza a vivir, es necesario que las empresas mexicanas estén preparadas para una fuerte competencia a nivel internacional y una de las medidas primordiales que deberán tomar es precisamente el contar con los medios necesarios que les provean de información oportuna y correcta.

En el caso específico del ramo bancario, se espera de un momento a otro que instituciones financieras extranjeras empiecen a operar en México, lo que provocará que las empresas bancarias nacionales se puedan ver amenazadas en cuanto a la captación del mercado mexicano.

El presente trabajo pretende desarrollar un producto de software que apoye a las empresas financieras específicamente en su departamento de cambio de divisas a través de aplicaciones automatizadas que les permitan mejorar su productividad y eficiencia con el objeto de incrementar su competitividad en el nuevo mercado internacional.

La estructura de esta trabajo sigue el ciclo de desarrollo del sistema.

El capítulo 1 explica la teoría de la Ingeniería de Software; se trata la evolución del software desde sus inicios hasta nuestros días, así como sus componentes y aplicaciones. Se habla también de los sistemas basados en computadoras y de sus elementos más importantes: la base de datos y los programas.

En el capítulo 2 daremos una explicación detallada de lo que es el departamento de cambio de divisas de una empresa bancaria, de su función, operación, así como de los conceptos más comunes que se manejan en los mercados de dinero y de valores. Haremos una reseña histórica del cambio de divisas que nos permita comprender la operación actual y a su vez nos ayude a prever líneas futuras.

Con el tercer capítulo iniciamos formalmente con el ciclo de vida del software, en este capítulo cubriremos la etapa de análisis y definición de requerimientos. Esta etapa se basa en un estudio profundo de la operación del departamento de cambios, identificando todos los procesos que la componen. Las actividades que deben cumplirse en esta fase se resumen en definición de

requerimientos, análisis estructurado y modelización de datos; estas actividades se verán finalmente plasmadas en un documento donde se relacionan los requisitos generales del software, los diagramas de flujo de datos con su diccionario de requisitos y el diagrama de entidad - relación.

El capitulo cuatro describe la etapa de diseño del software. En función del documento generado en la fase de análisis se realizan una serie de actividades propias del diseño, tales como, el diseño de datos, el diseño arquitectónico y el diseño procedimental, así mismo se definen productos intermedios dentro del desarrollo del producto: el diccionario de datos, la relación de estructuras de la base de datos, la definición estructural del producto, las especificaciones de la interfase con el usuario y el prototipo del sistema.

En el capitulo cinco se verá la implementación del sistema, entendiendo por implementación del sistema la creación por un lado de la base de datos y por otro lado el desarrollo de todas las aplicaciones que conformarán el sistema.

La creación de la base de datos se basa en las especificaciones obtenidas durante la fase de diseño y se sustenta en la teoria de bases de datos: sus elementos, el sistema de manejo de bases de datos (DBMS), el enfoque relacional y la arquitectura de un sistema relacional.

En la parte del desarrollo de las aplicaciones se mencionan las caracteristicas principales de los lenguajes de cuarta generación y especificamente de la herramienta de desarrollo.

En el capítulo seis se tratará la puesta en marcha del sistema, cubriendo así con todas las fases del ciclo de vida del software. La puesta en marcha consiste básicamente en la capacitación a los diferentes usuarios del sistema, es importante mencionar en este punto que la capacitación es diferente para cada uno de los usuarios dependiendo del perfil de sus actividades y de la relación que tendrán con el sistema.

Finalmente en el último capitulo se presentan las conclusiones.

ÍNDICE

CAPÍTULO I INGENIERÍA DE SOFTWARE

1.1.-	Software	1 - 1
1.1.1.-	Evolución del Software	1 - 2
1.1.2.-	Aplicaciones del Software	1 - 4
1.2 .-	Ingeniería de Sistemas	1 - 7
1.3.-	Modelos de la Ingeniería de Software	1 - 8
1.3.1.-	El Ciclo de Vida Clásico	1 - 8
1.3.2.-	Construcción de Prototipos	1 - 10
1.3.3.-	El Modelo en Espiral	1 - 12
1.3.4.-	Técnicas de Cuarta Generación	1 - 14
1.3.5.-	Conclusiones de los Modelos de Ingeniería de Software	1 - 15
1.4.-	Bases de Datos	1 - 16
1.4.1.-	Sistemas de Manejo de Bases de Datos	1 - 18
1.5.-	Sistemas basados en computadoras	1 - 19

CAPÍTULO II ANTECEDENTES: CASAS DE CAMBIO

	El mercado monetario	2 - 1
2.1.-	Divisas y su convertibilidad	2 - 1
2.2.-	Los bancos y el mercado de divisas	2 - 1
2.3.-	Reseña histórica	2 - 2
2.4.-	Operación del mercado de divisas	2 - 4
2.4.1.-	Cotización de divisas	2 - 4
2.4.1.1.-	Posiciones larga (long) y corta (short)	2 - 7
2.4.1.2.-	Estrategia cambiaria	2 - 8
2.4.2.-	Operaciones al contado	2 - 8
2.4.2.1.-	El cálculo de los tipos cruzados	2 - 10
2.4.2.2.-	Órdenes limitadas	2 - 12
2.4.2.3.-	Fijación de la fecha valor	2 - 14
2.4.2.4.-	Vigilancia de las posiciones	2 - 14
2.4.3.-	Operaciones a plazo	2 - 15
2.4.3.1.-	Tipo de cambio a plazo, prima y descuento	2 - 16
2.4.3.2.-	Operaciones <swap>	2 - 20
2.4.3.3.-	Operaciones <outright> de carácter especulativo	2 - 21
2.4.4.-	Operaciones del mercado monetario	2 - 23
2.4.4.1.-	Arbitraje del tipo de interés	2 - 23
2.4.4.2.-	Operación a plazo por operación a plazo	2 - 24

CAPÍTULO III ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS.

3.1.-	Definición de requerimientos	3 - 2
3.2.-	Descripción de la operación	3 - 3
3.2.1.-	Catálogos	3 - 3
3.2.2.-	Formas de Compra-Venta (DEAL'S)	3 - 3
3.2.3.-	Operación de compra - venta	3 - 4
3.2.4.-	Proceso de autorizaciones	3 - 5
3.2.5.-	Tesorería	3 - 5
3.2.6.-	Contabilidad	3 - 5
3.3.-	Requerimientos generales	3 - 6
3.4.-	Análisis estructurado	3 - 8
3.4.1.-	Diagrama de flujo de datos	3 - 8
3.4.2.-	Diccionario de requisitos	3 - 10
3.4.2.1.-	Diagrama de flujo de datos del software para la casa de cambio de divisas y su diccionario de requisitos	3 - 11
3.4.3.-	Diagrama de entidad - relación	3 - 30
3.4.3.1.-	Diagrama de entidad - relación para la casa de cambio	3 - 32

CAPÍTULO IV DISEÑO DEL SISTEMA

4.1.-	Diseño de datos	4 - 2
4.1.1.-	Diccionario de datos	4 - 2
4.1.2.-	Estructuras de datos	4 - 7
4.2.-	Diseño arquitectónico	4 - 17
4.3.-	Diseño procedimental	4 - 27

CAPÍTULO V IMPLEMENTACIÓN DEL SISTEMA

5.1.-	Conceptos Generales	5 - 1
5.1.1.-	Base de datos	5 - 1
5.1.2.-	Sistema de manejo de base de datos	5 - 2
5.1.3.-	El enfoque relacional	5 - 4
5.1.4.-	Arquitectura de un sistema relacional	5 - 5
5.2.-	Creación de la Base de Datos para la Casa de Cambio	5 - 10
5.2.1.-	Organización de la información	5 - 11
5.2.2.-	Métodos de acceso a la información	5 - 12
5.2.3.-	Secciones de programas precompilados	5 - 12
5.2.4.-	Tablas del sistema	5 - 13
5.3.-	Programación del sistema	5 - 15
5.3.1.-	Lenguajes de programación	5 - 16

5.3.2.- Lenguajes de cuarta generación 5 - 18
5.4.- Herramientas de cuarta generación para la casa de cambio 5 - 20
5.4.1.- Herramienta de definición de datos (Data definition tool) 5 - 20
5.4.2.- Herramienta para generación de formas (Form Painter tool) 5 - 22
5.4.3.- Herramienta para generación de reportes (Report Painter tool) 5 - 23
5.4.4.- Herramienta para mantenimiento de menúes (Menu Maintenance tool) 5 - 24
5.4.5.- Herramienta para generación de programas (Program Generator tool) 5 - 24
5.5.- Pruebas 5 - 25
5.5.1.- Objetivos de la prueba 5 - 26
5.5.2.- Métodos de prueba 5 - 26
5.5.3.- Desarrollo del proceso de pruebas 5 - 28

CAPÍTULO VI PUESTA EN MARCHA DEL SISTEMA

6.1.- Capacitación en el Sistema 6 - 2
6.1.1.- Capacitación en el uso del sistema para la compañía 6 - 3
6.2.- Migración de información al nuevo sistema 6 - 6
6.2.1.- Proceso de migración en la casa de cambio 6 - 7

CAPÍTULO VII CONCLUSIONES 7 - 1

- **BIBLIOGRAFÍA**
- **ANEXO 1**

CAPÍTULO 1

INGENIERÍA DE SOFTWARE

1.1.- Software.

Una descripción general del software podría ser: "instrucciones (programas de computadora) que cuando se ejecutan proporcionan la función y el comportamiento deseado, estructuras de datos que facilitan a los programas manipular adecuadamente la información, y documentos que describen la operación y el uso de programas".

Durante los primeros treinta años de la informática, el principal desafío fue el desarrollo del hardware de las computadoras, de forma que se redujera el costo de procesamiento y almacenamiento de datos. A lo largo de las últimas décadas los avances en microelectrónica han dado como resultado una mayor potencia de cálculo a la vez que una reducción del costo. Hoy, el problema es diferente. El principal desafío es mejorar la calidad y reducir el costo de las soluciones basadas en computadoras, soluciones que se implementan con el software.

La potencia de las grandes computadoras de la década anterior está hoy disponible en una computadora personal. Las enormes capacidades de almacenamiento y procesamiento del hardware moderno representan un gran potencial de cálculo. El software es el mecanismo que nos facilita utilizar y explotar este potencial.

1.1.1.- Evolución del Software.

El contexto en el que se ha desarrollado el software está fuertemente ligado a las casi cinco décadas de evolución de los sistemas informáticos. Un mejor rendimiento del hardware, una reducción del tamaño y un costo más bajo han dado lugar a sistemas informáticos más sofisticados. Hemos pasado de los procesadores con válvulas de vacío a los dispositivos microelectrónicos que son capaces de procesar 200 millones de instrucciones por segundo. Algunos autores denominan a este cambio como la "nueva revolución industrial" o la "tercera ola de cambio" en la historia de la humanidad.

La figura 1.1 describe la evolución del software dentro del contexto de las áreas de aplicación de los sistemas basados en computadoras.

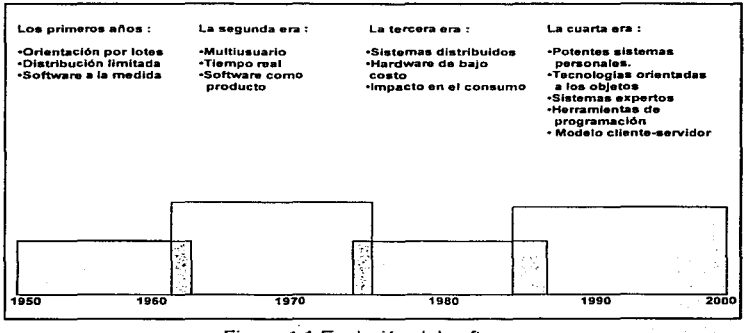


Figura 1.1 Evolución del software

Durante los primeros años de desarrollo de las computadoras, el hardware sufrió continuos cambios, mientras que el software se le veía simplemente como un añadido. El desarrollo del software se realizaba virtualmente sin ninguna planificación. Durante este período se utilizaba en la mayoría de los sistemas una orientación por lotes, mientras que el hardware se dedicaba a la ejecución de un único programa que, a su vez, se dedicaba a una aplicación específica.

En los primeros años, lo normal era que el hardware fuera de propósito general, mientras que el software se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña: la misma persona lo escribía, lo ejecutaba y si fallaba, lo depuraba.

Debido a este entorno personalizado del software, el diseño era un proceso implícito, realizado en la mente de alguien y la documentación normalmente no existía.

La segunda Era en la evolución de los sistemas de computadoras, se extiende desde la mitad de la década de los 60 hasta finales de los 70. La multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre-máquina. Los sistemas de tiempo real podían recoger, analizar y transformar datos de múltiples fuentes, controlando así los procesos y produciendo salidas en milisegundos en lugar de en minutos. Los avances en los dispositivos de almacenamiento en línea condujeron a la primera generación de sistemas de control de bases de datos.

La tercera Era en la evolución de los sistemas de computadora comenzó a finales de los 70 y llega hasta el momento actual. El procesamiento distribuido -- múltiples computadoras, cada una ejecutando funciones concurrentemente y comunicándose con alguna otra -- incrementó notablemente la complejidad de los sistemas informáticos. Las redes de área local y de área global, las comunicaciones digitales de alto ancho de banda y la creciente demanda de acceso "instantáneo" a los datos, supusieron una fuerte presión sobre los desarrolladores de software.

La tercera Era también se caracteriza por la llegada y el amplio uso de los microprocesadores y las computadoras personales.

La cuarta Era del software comenzó hace 2 años aproximadamente. Las tecnologías orientadas a objetos están desplazando rápidamente a enfoques de desarrollo de software más convencionales en muchas áreas de aplicación.

Las técnicas de cuarta generación para el desarrollo de software ya están cambiando la forma en que algunos segmentos de la comunidad informática construyen los programas de computadora. Por fin, los sistemas expertos y el software de inteligencia artificial se han trasladado del laboratorio a las aplicaciones prácticas, para un amplio rango de problemas del mundo real. El software de redes neuronales artificiales ha abierto excitantes posibilidades para el reconocimiento de formas y habilidades de procesamiento de información al estilo de como lo hacen los humanos.

1.1.2.- Aplicaciones del software.

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimentales, es decir, un algoritmo (exceptuando el software de los sistemas expertos y de redes neuronales). Para determinar la naturaleza de una aplicación de software, hay dos factores importantes que se deben considerar: el contenido y el determinismo de la información. El contenido se refiere al significado y a la forma de la información de entrada y salida.

El determinismo de la información se refiere a la predecibilidad del orden y del tiempo de llegada de los datos.

Software de sistemas: El software de sistemas es un conjunto de programas que han sido escritos para servir a otros programas. Algunos programas de sistemas (compiladores, editores y utilerías de control de archivos)

procesan estructuras de información complejas pero determinadas. Otras aplicaciones de sistemas procesan datos en gran medida indeterminados (ciertos componentes del sistema operativo, utilerías de manejo de periféricos, procesadores de telecomunicaciones). En cualquier caso, el área del software de sistemas se caracteriza por una fuerte interacción con el hardware de la computadora, una gran utilización por múltiples usuarios, una operación concurrente que requiere una planificación, una compartición de recursos y una sofisticada gestión de procesos, unas estructuras de datos complejas y múltiples interfases externas.

Software de Tiempo Real: El software que mide, analiza y controla sucesos del mundo real conforme ocurren, se denomina de tiempo real. Entre los elementos del software de tiempo real se incluyen: un componente de adquisición de datos que recolecta y da formato a la información recibida del entorno externo, un componente de análisis que transforma la información según lo requiera la aplicación, un componente de control/salida que responda al entorno externo y un componente de monitorización que coordina todos los demás componentes.

Software de Gestión: El procesamiento de información comercial constituye la mayor de las áreas de aplicación del software. Los "sistemas discretos" (por ejemplo: nóminas, sistemas de contabilidad financiera, inventarios, etc.) han evolucionado hacia el software de sistemas de información de gestión, que accesa a una o más bases de datos grandes que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes con el objeto de facilitar las operaciones comerciales o apoyar las tomas de decisiones. Además de las tareas convencionales de procesamiento de datos, las aplicaciones de software de gestión también realizan cálculo interactivo (por ejemplo: el procesamiento de transacciones en puntos de ventas).

Software de Ingenieria y Científico: El software de ingenieria y científico está caracterizado por los algoritmos "manejo de números". Las aplicaciones más caracteristicas son en la astronomía, vulcanología, biología molecular, dinámica orbital de lanzamientos espaciales, etc.

Software Empotrado: Los productos inteligentes se han convertido en algo común en casi todos los mercados de consumo e industriales. El software empotrado reside en memoria de sólo lectura y se utiliza para controlar productos y sistemas de los mercados industriales. El software empotrado puede ejecutar funciones muy limitadas (por ejemplo: el control de los teclados de un horno de microondas) o suministrar una función significativa y con capacidad de control (funciones digitales en un automóvil, tales como control de la gasolina, aceite, sistemas de frenado).

Software de Computadoras Personales: El mercado del software de las computadoras personales ha germinado en la pasada década. El procesamiento de textos, las hojas de cálculo, los gráficos por computadora, entretenimientos, control de bases de datos, aplicaciones financieras, de negocios y personales, y redes o acceso a bases de datos externas son sólo algunos de los cientos de aplicaciones. De hecho, el software de las computadoras personales continúa representando uno de los diseños del software más innovadores en el campo del software.

Software de Inteligencia Artificial: El software de Inteligencia Artificial (IA) hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo. Actualmente el área más activa de la I.A. es la de los sistemas expertos o sistemas basados en el conocimiento. Otras áreas de aplicación para el software de I.A. es el reconocimiento de patrones (imágenes y voz), la prueba de teoremas y juegos.

En los últimos años ha surgido una nueva rama del software de I.A. llamada redes neuronales artificiales. Una red neuronal simula la estructura de proceso del cerebro (las funciones de la neurona biológica) y a la larga puede llevar a una clase de software que pueda reconocer patrones complejos y aprender de "experiencia" pasada.

1.2.- Ingeniería de Sistemas.

... Algunos autores han definido a la Ingeniería de Software como: "El establecimiento y uso de principios de ingeniería, orientados a obtener software económico que sea fiable y funcione de manera eficiente sobre máquinas reales".

La Ingeniería de Software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de tres elementos clave -- métodos, herramientas y procedimientos -- que facilitan al responsable controlar el proceso del desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva.

Los métodos de la Ingeniería de Software indican "cómo" construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requisitos del sistema y del software, diseño de estructuras de datos, arquitectura de programas y procedimientos algoritmicos, codificación, prueba y mantenimiento.

Las herramientas de Ingeniería de Software suministran un soporte para los métodos. Actualmente existen herramientas para soportar cada uno de los métodos mencionados anteriormente. Cuando se integran las herramientas de manera que la información creada por alguna de ellas pueda ser usada por otra, se establece un sistema para el soporte del desarrollo del software, llamado ingeniería de software asistida por computadora (CASE). CASE combina software, hardware y bases de datos para crear un entorno de ingeniería de software para el hardware.

Los procedimientos de la Ingeniería de Software relacionan los métodos y las herramientas facilitando un desarrollo racional y oportuno del software de computadora. Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas (documentos, informes, formas, etc.) que se requieren, los controles que ayudan a asegurar la calidad y coordinar los cambios, y las directrices que permitirán a los desarrolladores del software a evaluar el progreso.

La Ingeniería de Software está pues compuesta por una serie de pasos que abarcan los métodos, las herramientas y los procedimientos antes mencionados. Estos pasos se denominan paradigmas o modelos de la Ingeniería de Software. La elección de un modelo se lleva a cabo en función de la naturaleza del proyecto y de la aplicación, los métodos y herramientas a usar y los controles y entregas requeridos.

1.3.- Modelos de la Ingeniería de Software.

1.3.1.- El ciclo de vida clásico.

El ciclo de vida clásico de la ingeniería de software o también conocido como "modelo en cascada" exige un enfoque sistemático y secuencial del desarrollo del software que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento. La figura 1.2 muestra las actividades a desarrollar en base a este modelo.

Ingeniería y Análisis del Sistema: Partiendo de que el software es siempre parte de un sistema mayor, el trabajo debe comenzar estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software. Este planteamiento del sistema es esencial cuando el software debe interrelacionarse con otros elementos, tales como hardware, personas y bases de datos. La Ingeniería y Análisis del sistema abarca los requisitos globales a nivel del sistema.

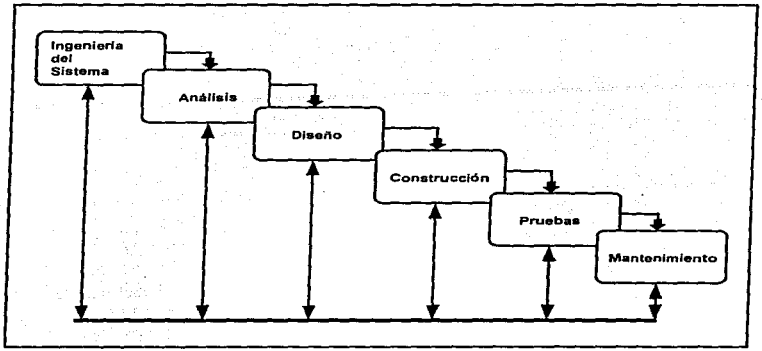


Figura 1.2 Ciclo de vida clásico

Análisis de los requisitos del software: El proceso de recopilación de los requisitos se centra e intensifica especialmente para el software. Con el objeto de comprender la naturaleza de los programas que hay que construir, el ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfases requeridas. Es importante mencionar que el resultado de esta fase debe quedar plasmado en un documento de análisis que normalmente deberá ser revisado con el cliente.

Diseño: El diseño del software se enfoca en cuatro atributos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la

calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

Codificación: El diseño debe traducirse en una forma legible para la computadora. El paso de codificación realiza esta tarea.

Prueba: Una vez que se ha generado el código, se deben hacer pruebas al programa. La prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento: El software sufrirá cambios después de ser liberado (una posible excepción es el software empotrado). Los cambios ocurrirán debido a que se hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo, o debido a que el cliente requiera ampliaciones funcionales o de rendimiento. El mantenimiento del software aplica cada uno de los pasos precedentes del ciclo de vida a un programa existente en vez de a uno nuevo.

1.3.2.- Construcción de Prototipos.

En ocasiones un cliente no logra identificar los requisitos de entrada, proceso o salida, o bien, el programador no esta seguro de la forma en que debe de realizarse la interacción hombre-máquina. En estas situaciones, puede ser mejor método de ingeniería de software la construcción de un prototipo.

La construcción de prototipos es un proceso que facilita al programador la creación de un modelo del software a construir. El modelo tomará una de las tres formas siguientes:

- Un prototipo en papel o un modelo basado en computadora que describa la interacción hombre-máquina, de forma que facilite al usuario la comprensión de como se producirá tal interacción.

- Un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado.

- Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deban ser mejoradas en el nuevo trabajo de desarrollo.

La figura 1.3 muestra la secuencia de sucesos del modelo de construcción de prototipos.

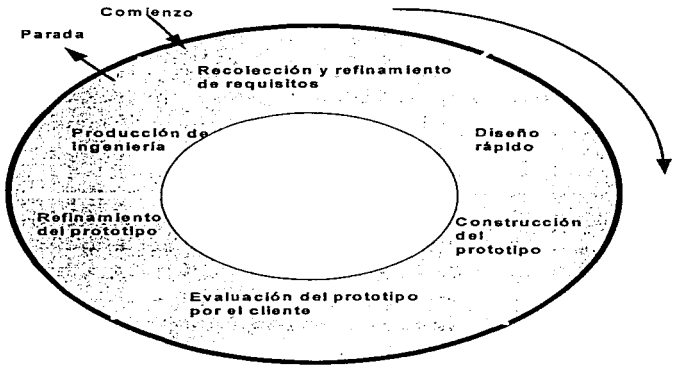


Figura 1.3 Creación de prototipos

Como en todos los métodos de desarrollo de software, la construcción de prototipos comienza con la recolección de los requisitos. El ingeniero y el cliente se reúnen y definen los objetivos globales para el software, identificando todos los requisitos conocidos y perfilan las áreas en donde será necesario una mayor definición. Luego se produce un "Diseño rápido". El diseño rápido se enfoca sobre la representación de los aspectos del software visibles al usuario (métodos de entrada y formatos de salida). El diseño rápido conduce a la construcción de un prototipo. El prototipo es evaluado por el cliente y el usuario y se refinan los requisitos del software a desarrollar.

1.3.3.- El Modelo en Espiral.

El modelo en espiral para la ingeniería de software ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgo, que no se contempla en esos modelos. El modelo en espiral define cuatro actividades principales, representadas por los cuatro cuadrantes de la figura 1.4.

a) Planificación: determinación de objetivos, alternativas y restricciones.
b) Análisis de riesgo: análisis de alternativas e identificación/resolución de riesgos.
c) Ingeniería: desarrollo del producto de "siguiente nivel".
d) Evaluación del cliente: valoración de los resultados de la ingeniería.

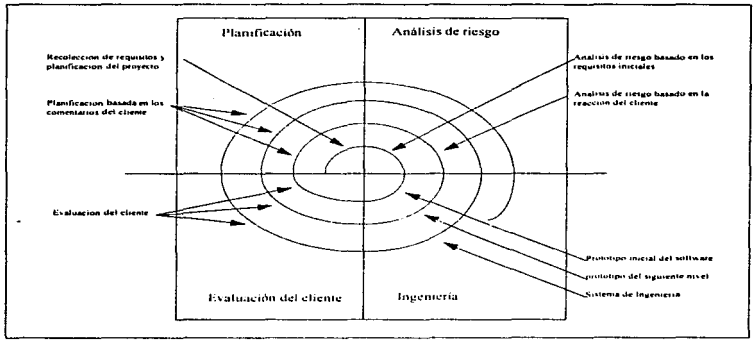


Figura 1.4 Modelo en espiral

Con cada iteración al rededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completas. Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones y se analizan e identifican los riesgos. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede recurrir a la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado del desarrollo como al cliente.

El cliente evalúa el trabajo de ingeniería y sugiere modificaciones. En base a los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo.

Cada vuelta alrededor de la espiral requiere ingeniería, que se puede llevar a cabo mediante el enfoque del ciclo de vida clásico o de la creación de prototipos.

El modelo en espiral para la ingeniería de software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque "evolutivo" que permite al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel. Se apoya en la creación de prototipos como un mecanismo de reducción del riesgo.

1.3.4.- Técnicas de Cuarta Generación.

Las técnicas de cuarta generación (T4G) están conformadas por un amplio espectro de herramientas de software orientadas a facilitar la especificación de algunas características del software a alto nivel. Posteriormente, la herramienta genera automáticamente el código fuente basándose en la especificación técnica.

Actualmente, un entorno para el desarrollo de software que soporte el paradigma T4G puede incluir todas o algunas de las siguientes herramientas: lenguajes no procedimentales para consulta a bases de datos, generación de informes, manipulación de datos, interacción y definición de pantallas, generación de código, facilidades gráficas de alto nivel y facilidades de hoja de cálculo.

En la figura 1.5 se describe el paradigma T4G para la ingeniería de software. Al igual que otros modelos, T4G comienza con la fase de recolección de requisitos. Idealmente, el cliente describe los requisitos, que son a continuación, traducidos directamente a un prototipo operativo.

Para aplicaciones pequeñas, se puede ir directamente desde el paso de recolección de requisitos al paso de implementación, usando un lenguaje de cuarta generación no procedimental (L4G). Sin embargo, es necesario un mayor esfuerzo para desarrollar una estrategia de diseño para el sistema, incluso si se utiliza un L4G. El uso de T4G sin diseño en grandes proyectos podrá causar dificultades como: poca calidad, mantenimiento pobre, mala aceptación por el cliente.

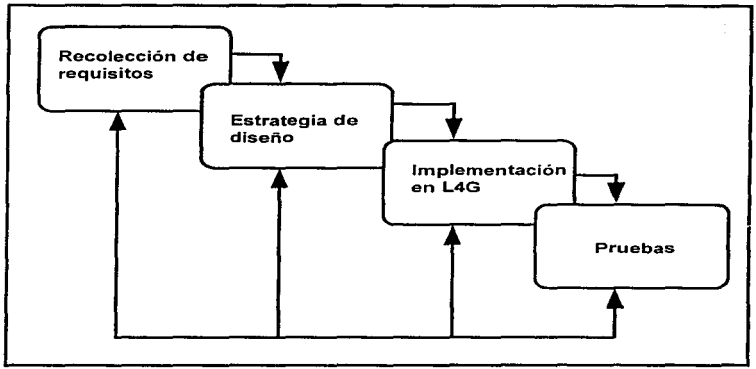


Figura 1.5 Técnicas de cuarta generación

La implementación mediante un L4G permite, al que desarrolla el software, centrarse en la representación de los resultados deseados, que es lo que se traduce automáticamente en un código fuente que produce dichos resultados.

Para transformar una implementación T4G en un producto, se debe de dirigir una prueba completa, desarrollar una documentación con sentido y ejecutar el resto de actividades de "transición" requeridas en los otros modelos de Ingeniería de Software.

1.3.5.- Conclusiones de los Modelos de Ingeniería de Software.

En la vida profesional, los ingenieros en computación se enfrentan a proyectos que por su magnitud o complejidad se hace difícil la selección de

alguno de los modelos de ingenieria de software descritos hasta ahora, debido a que ninguno de ellos cubre en su totalidad las exigencias que el mismo proyecto impone. En muchos casos, los modelos pueden y deben combinarse de forma que se utilicen las ventajas de cada uno en un único proyecto.

De alguna manera, el modelo en espiral lo hace directamente, combinando la creación de prototipos y algunos elementos del ciclo de vida clásico, en un enfoque evolutivo para la ingenieria de software. Pero ninguno de los modelos puede servir como base en la cual se integren los demás.

La figura 1.6 muestra cómo pueden combinarse los tres modelos mencionados durante un trabajo de desarrollo de software. En todos las casos, el trabajo comienza con la recolección preliminar de requisitos. A partir de ese punto, se puede tomar cualquiera de los caminos indicados en la figura. Por ejemplo, si se puede especificar completamente el sistema en un principio, podemos seguir los pasos del ciclo de vida clásico; si los requisitos son inciertos, se puede usar un prototipo. Usando el prototipo como guía, el ingeniero puede después volver a los pasos del ciclo de vida clásico (diseño, codificación y prueba). Se pueden usar las técnicas de cuarta generación para implementar el prototipo o para implementar el sistema durante la fase de codificación del ciclo de vida.

En conclusión podemos decir que la naturaleza de la aplicación nos dictará el modelo básico a seguir, pero siempre complementándolo con actividades de los otros modelos que nos puedan ser de utilidad en un momento dado.

1.4.- Bases de Datos.

Un sistema de bases de datos es un sistema computarizado de información para el manejo de datos por medio de paquetes de software llamados sistemas de manejo de bases de datos (DBMS). Los tres componentes principales de un sistema de bases de datos son el hardware, el software (DBMS) y los datos por manejar.

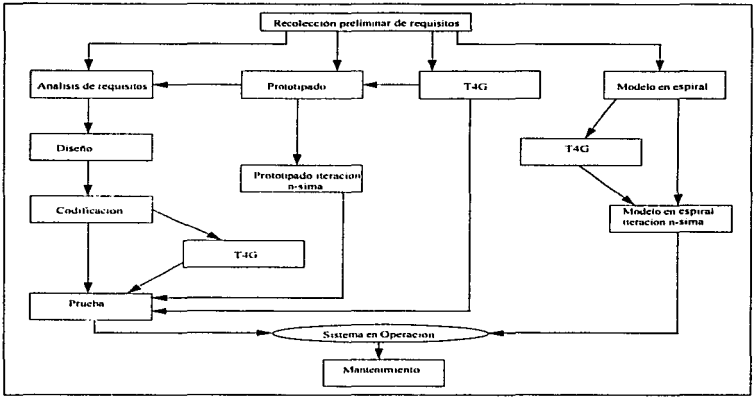


Figura 1.6 Combinación de modelos

Una base de datos es una colección de archivos interrelacionados creados con un DBMS. El contenido de una base se obtiene combinando datos de todas las diferentes fuentes en una organización, de tal manera que los datos estén disponibles para todos los usuarios, y los datos redundantes puedan eliminarse, o al menos minimizarse. La figura 1.7 muestra la base de datos como un recipiente de datos a ser compartidos por varios programas.

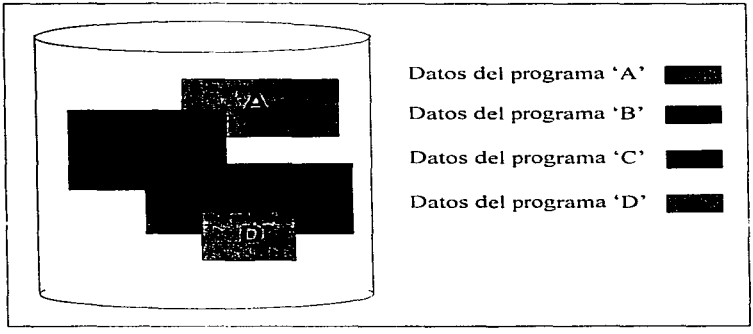


Figura 1.7 Base de datos

1.4.1.- Sistemas de Manejo de Bases de Datos.

El sistema de manejo de bases de datos (DBMS) es la porción más importante del software de un sistema de base de datos.

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica. Las funciones principales de un DBMS son:

a) Crear y organizar la base de datos.

b) Establecer y mantener las trayectorias de acceso a la base de datos, de tal manera que los datos en cualquier parte de la base se puedan accesar rápidamente.

c) Manejar los datos de acuerdo con las peticiones de los usuarios.

d) Mantener la integridad y seguridad de los datos.

e) Registrar el uso de las bases de datos.

El DBMS interpreta y procesa las peticiones del usuario para recobrar la información de la base. La figura 1.8 muestra como un DBMS sirve de interfase entre las peticiones del usuario y la base de datos.

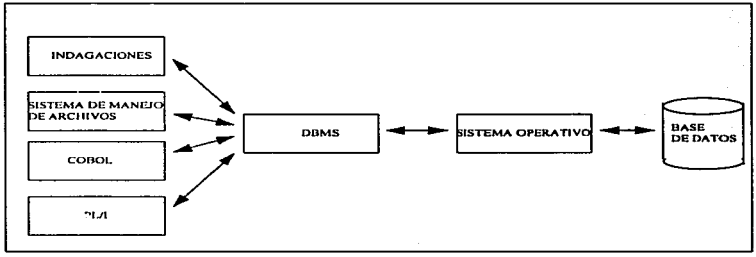


Figura 1.8 Relación usuario-base de datos

Los requerimientos a la base de datos pueden tener distintas formas, pueden teclearse directamente desde la terminal, o codificarse como programas en lenguajes de alto nivel y presentarse para procesamiento interactivo o por lotes. En la mayoría de los casos, una petición de consulta tendrá que atravesar varias capas de software en el DBMS y en el sistema operativo antes de que se pueda accesar la base de datos física.

1.5.- Sistemas Basados en Computadoras.

Entendiendo por sistema un conjunto u ordenación de elementos relacionados de tal manera que forman una unidad o un todo orgánico, o bien, un conjunto de hechos, principios, reglas, etc. clasificados y ordenados de tal manera

que muestran un plan lógico uniendo las diferentes partes. Podemos definir un "sistema basado en computadoras" como: Un conjunto u ordenación de elementos organizados para llevar a cabo algún método, procedimiento o control mediante el proceso de información.

Los elementos que componen un sistema basado en computadora se muestran en la figura 1.9 :

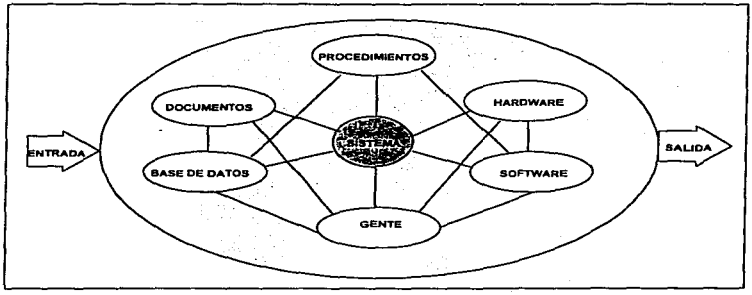


Figura 1.9 Sistemas basados en computadoras

Software: Los programas de computadora, las estructuras de datos y la documentación asociada, que sirven para realizar el método lógico, procedimiento o control requerido.

Hardware: Los dispositivos electrónicos (CPU, Memoria, etc.) que proporcionan la capacidad de computación y los dispositivos electromecánicos (sensores, motores, bombas) que proporcionan las funciones del mundo exterior.

Gente: Los individuos que son usuarios y operadores del software y del hardware.

Bases de Datos: Una colección grande y organizada de información a la que se accede mediante el software y que es una parte integral del funcionamiento del sistema.

Documentación: Los manuales, los impresos y otra información descriptiva que explica el uso y/o la operación del sistema.

Procedimientos: Los pasos que definen el uso específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

CAPÍTULO II
ANTECEDENTES: CASAS DE CAMBIO

EL MERCADO MONETARIO

2.1.- Divisas y su convertibilidad.

El comercio internacional y en mayor medida los movimientos registrados en los mercados internacionales financiero y monetario, constituyen la base de las operaciones de cambio de divisas.

Se entiende por divisa cualquier crédito expresado en moneda extranjera y pagaderos en el extranjero, que puede consistir en haberes (en moneda extranjera) en bancos extranjeros, o en letras de cambio o cheques (también en moneda extranjera) y pagaderos en el extranjero.

Una moneda o divisa se considera convertible cuando los medios de pago expresados en dicha moneda pueden ser libremente convertidos por el poseedor en cualquier otra moneda.

2.2.- Los bancos y el mercado de divisas.

Los bancos son los intermediarios normales entre la oferta y la demanda de divisas. El objetivo principal del departamento de cambios de un banco consiste en facilitar a su clientela comercial y privada la conversión de haberes en otras monedas. Esta conversión puede efectuarse mediante operaciones al contado o a plazo.

La propia actividad de los bancos en el mercado de divisas tiende a establecer un cambio uniforme para una divisa determinada en las diversas plazas financieras del mundo. Si el cambio de una divisa se desvía demasiado, el equilibrio se restablece mediante el arbitraje (técnica que consiste en beneficiarse de las diferencias de cambio en las distintas plazas, y que se fundamenta principalmente en la oferta y la demanda de la divisa en cuestión).

Paralelamente al desarrollo del comercio internacional, el mercado de divisas experimentó una enorme expansión a principios de los años ochentas. Esta expansión ha adquirido una amplitud todavía mayor bajo el régimen del flotamiento de las monedas, que se ha caracterizado por las pronunciadas fluctuaciones de los tipos de cambio.

2.3.- Reseña histórica.

Las operaciones cambiarias son probablemente tan antiguas como el dinero mismo, pero la técnica del comercio de divisas sólo data de fines del siglo pasado. En aquel entonces se había formado un sistema monetario basado en el oro como valor de referencia. Este sistema creó la confianza necesaria para que en los pagos internacionales se efectuasen adeudos y abonos en cuentas abiertas en el extranjero, sin efectuar transferencia efectiva alguna del correspondiente contravalor. La característica principal del sistema basado en el oro consistía en que era un sistema de tipos de cambio fijos, cuyas paridades se fijaban con respecto al oro.

En este sistema con posibilidades de intervención estatal, las fluctuaciones de los tipos de cambio eran insignificantes y los riesgos de poca importancia.

Al término de la segunda guerra mundial, los Estados Unidos y la Gran Bretaña instauraron un sistema monetario libre, multilateral y estable. El Fondo Monetario Internacional (FMI) fue creado para velar por el buen funcionamiento del nuevo sistema monetario, cuyos objetivos pueden resumirse de la siguiente manera: crear un sistema monetario internacional con tipos de cambio estables; eliminar las restricciones de cambio todavía existentes y restablecer la convertibilidad de todas las monedas.

Cada país miembro del FMI determinaba la paridad de su moneda con relación al oro o al dólar y se comprometía a mantener, mediante intervenciones de su Banco Central, las variaciones del tipo de cambio dentro de un margen de fluctuación de ± 1% con respecto a la paridad establecida. Con objeto de evitar devaluaciones injustificadas, sólo se podía modificar la paridad previa consulta con el FMI.

El dólar americano representó un papel primordial en este sistema. Su función más importante fue la de moneda de reserva. Su libre convertibilidad en oro hacía del dólar un patrón de paridad. El dólar llegó a ser, como era lógico, la moneda por excelencia de las transacciones internacionales.

A partir del año de 1973, el orden monetario internacional se ha caracterizado por la coexistencia de una variada gama de sistemas de tipos de cambio. Casi todos los países industrializados practican un sistema de tipo de cambio flotante controlado. Algunos países europeos se han asociado para constituir un bloque monetario, manteniendo entre ellos paridades fijas, pero dejan flotar sus monedas con respecto a las demás.

Los países en desarrollo, han conservado tipos de cambio fijos con respecto al dólar u otra moneda de reserva.

2.4.- Operación del mercado de divisas.

2.4.1.- Cotización de divisas.

La mayoría de los países aplican la cotización directa. Esto significa que los tipos de cambio dan el equivalente de una determinada cantidad de moneda extranjera cotizada. Debido a ello las monedas extranjeras se expresan en México en nuevos pesos.

Sin embargo, cuando se trata del cambio profesional, practicado entre bancos, se cotiza normalmente al tipo del dólar. Esto es, los valores de las diferentes monedas nacionales son expresados indicando el precio de un dólar en dichas monedas.

La información de los tipos de cambio con la que debe de contar el departamento de cualquier banco en el mundo es la siguiente:

	Cotización de compra (demanda)	Cotización de venta (oferta)
$ / fr.s.	1.9760	1.9780
$ / DM	2.1740	2.1760
$ / fl.h.	2.3660	2.3690
$ / fr.fr.	5.0325	5.0375

Este cuadro indica las cotizaciones de compra y de venta aplicadas en las operaciones entre bancos. En las transacciones con la clientela se pueden aplicar márgenes algo más amplios.

Constantemente los clientes y los pequeños bancos recurren al departamento de divisas para transacciones por otras monedas diferentes al dólar. En estos casos, se aplica la técnica de tipos cruzados que consiste en obtener el tipo de cambio medio de una moneda respecto de otra basándose en los tipos medios del dólar respecto a cada una de las monedas involucradas en la transacción.

Esto es, si quisiéramos obtener el tipo de cambio medio entre la moneda A y la moneda B, tendríamos las siguientes cotizaciones respecto al dólar.

	COTIZACIÓN DE COMPRA	COTIZACIÓN DE VENTA
$ / MONEDA "A"	C.C.A	C.V.A
$ / MONEDA "B"	C.C.B	C.V.B

En primer lugar habría que obtener el tipo de cambio medio respecto al dólar de cada moneda:

$$\text{tipo de cambio medio de "A"} = \frac{C.C.A + C.V.A}{2} \quad \ldots \text{T.C.M. "A"}$$

$$\text{tipo de cambio medio de "B"} = \frac{C.C.B + C.V.B}{2} \quad \ldots \text{T.C.M. "B"}$$

que tabularmente podríamos expresarlo de la siguiente manera

	TIPO DE CAMBIO MEDIO
$ / MONEDA "A"	T.C.M. "A"
$ / MONEDA "B"	T.C.M. "B"

que expresado matemáticamente sería

(T.C.M. "A") A = 1 $ . . . . (1)

(T.C.M. "B") B = 1 $ . . . . (2)

igualando las ecuaciones (1) y (2) tendríamos :

(T.C.M. "A") A = (T.C.M. "B") B

si quisiéramos adquirir moneda "B" pagando con moneda "A" , tendríamos que el valor de "B" sería:

$$B = \frac{\text{(T.C.M. "A")}}{\text{(T.C.M. "B")}} \quad \text{(A)}$$

y de igual forma para adquirir moneda "A" pagando con moneda "B" :

$$A = \frac{\text{(T.C.M. "B")}}{\text{(T.C.M. "A")}} \quad \text{(B)}$$

2.4.1.1.- Posiciones larga (long) y corta (short).

Un banco que actúa en negocios internacionales tiene que mantener suficientes saldos activos en las principales monedas. Las cuentas corrientes mantenidas con corresponsales extranjeros no deben ser sobregiradas; pero en el caso que, por corto tiempo y por casualidad, aparezca un saldo deudor, se tendrá que pagar el interés correspondiente. Dichos saldos activos (working balances) son fondos en divisas y, por esto mismo, posiciones long (se cuenta con más divisas de las que se han vendido). Como estas divisas deben ser comparadas con otra moneda -por ejemplo con la propia moneda nacional- se tiene automáticamente una posición short por los respectivos importes en la moneda nacional, es decir, para el caso de un banco mexicano se tendrán las correspondientes deudas en pesos mexicanos (se han vendido más divisas de las que se han comprado).

El departamento de divisas del banco tiene que vigilar constantemente sus posiciones en las diferentes monedas, lo cual se hace actualmente por medio de computadoras. La llamada posición del cambista (dealer position) tiene que reflejar la posición de riesgo en las diferentes monedas, prescindiendo de los vencimientos; tiene que registrar no sólo las transacciones al contado (spot), sino también las operaciones a plazo y las posiciones de riesgo en divisas resultantes de las operaciones en el mercado monetario. (El término "posición de riesgo" (risk position), es usado con frecuencia para aclarar que se refiere a la posición global, es decir, incluyendo las posiciones a plazo (forward position) y no sólo la posición al contado (spot)).

2.4.1.2.- Estrategia cambiaria.

El jefe del departamento de cambios proporciona a sus colaboradores los lineamientos para sus actividades cambiarias del día y define sus tácticas basándose en los siguientes factores :

a) Los tipos de cierre en Nueva York, Londres, Suiza y Tokio.

b) Los nuevos acontecimientos económicos y políticos.

c) La propia posición de divisas.

En base a las cotizaciones de apertura del dólar, se calculan las cotizaciones de apertura de las principales monedas con respecto al peso mexicano que servirán de punto de partida para los negocios iniciales.

El departamento de divisas del banco no se limita a ejecutar órdenes de sus clientes, sino que actúa también por su propia cuenta. A lo largo del día se intenta comprar divisas a los mejores tipos posibles, así como venderlas con beneficios.

2.4.2.- Operaciones al contado.

Supongamos que nuestra cotización del cierre del $ / fr.s. fue ayer de 1.9760 / 1.9780, mientras en Nueva York cerró a 1.9750 / 1.9770. Si un banco nos pregunta cuál es nuestra cotización, nosotros cotizamos algo más bajo, por ejemplo 1.9740 / 1.9760. Si ellos no concluyen ninguna transacción, podemos suponer que ese banco considera normal nuestra cotización (los cambistas dicen <par> o <paridad>) por el momento. Podemos, pues, mantener esta cotización.

En la práctica, los cambistas no mencionan el tipo completo, sino simplemente los dos últimos decimales : 40 a 60, pues los profesionales saben perfectamente qué cifras vienen delante.

Si el banco B de Londres nos llama y, habiéndole dado nosotros la cotización anterior, nos dice: <A 40 vendo a ustedes $ 2 millones.> el jefe de los cambistas ajustará probablemente sus cotizaciones quizás a 35 / 55, pues si nosotros nos quedásemos con nuestras cotizaciones anteriores, podrían llegar más dólares de los que necesitamos.

El trámite administrativo consistiría en llenar la ficha de compra conteniendo los siguientes datos: cliente, fecha, fecha valor, compra o venta, tipo de cambio, tipo cruzado, moneda nacional, recibimos, pagamos. También se efectuarían las siguientes operaciones:

- Registro de la transacción en las posiciones (la computadora indicará las nuevas posiciones minutos después de concluida la operación).
- La sección de correspondencia del departamento de divisas expide para la otra parte una confirmación (impresa por la computadora) y se efectúan los pagos.
- El departamento de contabilidad hace los asientos correspondientes.

En días de intensa actividad, se llegan a efectuar hasta más de mil transacciones. Los negocios profesionales entre bancos se efectuarán siempre en dólares.

En ocasiones, clientes o pequeños bancos desean negociar con nosotros <moneda a moneda>, lo cual, significa que el dólar no está implicado en la operación.

2.4.2.1.- El cálculo de los tipos cruzados.

Por ejemplo, el cliente C quiere comprar 5 millones de marcos alemanes, a cambio de francos suizos.

Tenemos que calcular el cambio pedido partiendo de los correspondientes cambio del dólar, que son los siguientes:

$ / DM	2.1740 - 2.1760
$ / fr.s.	1.9760 - 1.9780

Procedemos de la manera siguiente:

- El cliente quiere comprar marcos alemanes. Si el departamento de divisas no los tiene, tendrá que comprarlos (preferentemente en Alemania) y pagarlos en dólares, al cambio de compra del dólar fijado por el banco alemán, es decir, 2.1740 marcos alemanes.
- Con el fin de obtener los dólares que se tienen que pagar en Alemania, vende por dólares a un banco de Zurich los francos suizos que se recibió del cliente, al precio de venta del dólar, es decir, frs.s. 1.9780.

Obtenemos el tipo de venta de los marcos alemanes por francos suizos con la siguiente ecuación:

X frs.s.	=	100 DM
2.1740 DM	=	1 $
1 $	=	1.9780 frs.s.

$$X\ (100\ DM) = \frac{100 \times 1.9780}{2.1740} = 90.98\ frs.s.$$ (cotización de venta)

Vendiendo marcos alemanes a frs.s. 90.98, no se obtendrá ningún beneficio, pues si compra marcos alemanes en el mercado y vende francos suizos, frs.s. 90.98 es realmente su propio costo. Esta es la razón por la cual se añade a este importe un pequeño margen de beneficio.

Cuando lo que el cliente quiere no es comprar, sino vender marcos alemanes a cambio de francos suizos, el cálculo será el siguiente

X frs.s.	=	100 DM
2.1760 DM	=	1 $
1 $	=	1.9760 frs.s.

$$X\ (\ 100\ DM\ ) = \frac{100 \times 1.9760}{2.1760} = 90.81\ frs.s.$$ (cotización de compra)

Pero si este cliente no fuera un suizo, sino un alemán, no se usarían los tipos que se calcularon. No se comprarían marcos alemanes por francos suizos, sino que se venderían francos suizos por marcos alemanes. Dicho de otro modo, él no desea tener un tipo de cambio del marco alemán con respecto al franco suizo, sino un tipo de cambio del franco suizo con respecto al marco alemán.

Basándonos en las mismas cotizaciones

$ / DM	2.1740 - 2.1760
$ / fr.s.	1.9760 - 1.9780

el correspondiente cálculo sería el siguiente:

X DM	=	100 frs.s.
1.9780 frs.s.	=	1 $
1 $	=	2.1740 DM

$$X\ (\ 100\ frs.s.\ ) = \frac{100 \times 2.1740}{1.9780} = 109.91\ DM$$ (cotización de compra)

Es obvio que este precio del franco suizo expresado en marcos alemanes puede transformarse en el precio del marco alemán expresado en francos suizos. El cálculo se efectúa de la siguiente manera:

X frs.s. = 100 DM

109.91 DM = 100 frs.s.

$$X\ (\ 100\ DM\ ) = \frac{100 \times 100}{109.91} = 90.98\ frs.s.$$

2.4.2.2.- Órdenes limitadas.

Un banco del Brasil nos dió ayer en la tarde una orden limitada, por la cual se le pidió al departamento de divisas que comprara 10 millones de francos franceses a cambio de marcos alemanes a 43.15.

Suponiendo que las cotizaciones de apertura eran esta mañana:

$ / DM 2.1740 - 2.1760

$ / fr.fr. 5.0325 - 5.0375

¿Se podía ejecutar esta orden a la apertura?

No, pues el precio de venta habría sido el siguiente:

$$X\ (\ 100\ fr.fr.\ ) = \frac{100 \times 2.1760}{5.0325} = 43.24\ DM$$

Si la cotización dólar / franco francés se mantiene inalterada, ¿a qué cotización debería bajar la paridad dólar / marco alemán para que se pudiera ejecutar la orden?

X DM = 1 $
1 $ = 5.0325 fr.fr.
100 fr.fr. = 43.15 DM

$$X\ (1\ \$) = \frac{5.0325 \times 43.15}{100} = 2.1715\ DM$$

Si la cotización del dólar con respecto al marco alemán permanece inalterada, ¿a qué nivel debería subir la paridad $ / fr.fr.?

X fr.fr. = 1 $
1 $ = 2.1760 DM
43.15 DM = 100 fr.fr.

$$X\ (1\ \$) = \frac{2.1760 \times 100}{43.15} = 5.0429\ fr.fr.$$

¿A qué cotización corresponden 43.15 DM en moneda francesa?

X fr.fr. = 100 DM
43.15 DM = 100 fr.fr.

$$X\ (100\ DM) = \frac{100 \times 100}{43.15} = 231.75\ fr.fr.$$

2.4.2.3.- Fijación de la fecha valor.

La fecha valor es la fecha en la cual las divisas deberán ser abonadas por ambas partes. En todas las operaciones normales de cambio, la fecha valor es el segundo día laborable después de la fecha en que se concluye la transacción (para dar tiempo a que se efectúen los correspondientes trámites administrativos). En el mundo occidental, los negocios al contado efectuados el jueves llevarán por fecha la del lunes siguiente.

2.4.2.4.- Vigilancia de las posiciones.

Como se tienen negocios en muchas monedas, el jefe cambista debe vigilar constantemente su posición, para no acumular monedas débiles ni quedarse corto en monedas fuertes. Las cotizaciones que se den serán, por lo tanto, influenciadas por su posición actual y en definitiva, por su opinión o estimación sobre las monedas principales.

Ilustraremos esto con un ejemplo práctico. Se ha empezado el día con una moderada posición <short> en libras esterlinas. Teniendo la intención de dejar la posición más o menos inalterada, pues esta convencido de que la libra esterlina es una moneda débil, aunque ello no significa que vaya a debilitarse hoy. Al primer banco que ha preguntado el cambio £ / $ esta mañana, le han dado la cotización de 2.2645 / 55, y dicho banco ha vendido entonces 3 millones de libras esterlinas a la cotización de compra de 2.2645. Por lo tanto, bajará su cotización siguiente, p.ej. a 2.2642 / 52, <enmarcado> así el tipo anterior, según el lenguaje de los cambistas. Al obrar de este modo, se piensa que, probablemente, a 42, ya no ofrecerán libras y si encontramos compradores a 52, se obtendrá una ganancia de siete puntos.

Naturalmente, si cotiza a 44 / 54 y aún se encuentran tomadores, su ganancia será mayor; pero al mismo tiempo, a la cotización de compra, relativamente alta, de 44, se corre el riesgo de recibir más libras, que es lo que se desea evitar.

2.4.3 Operaciones a plazo.

Se puede comprar o vender divisas no sólo al contado, sino también a plazo (es decir, para su entrega en una fecha posterior convenida). En la práctica, la cotización a plazo es casi siempre mayor (<report>, suplemento, prima), o menor (<deport>, descuento) que la cotización al contado.

Se recurre a las operaciones a plazo con fines diversos. Estas operaciones permiten principalmente cubrir o asegurar los riesgos inherentes a las operaciones de cambio que resultan de transacciones comerciales o financieras. A la combinación de una compra al contado con una venta a plazo (o viceversa) en las operaciones del mercado monetario se le denomina <swap>. Para evitar confusiones al hablar de operaciones a plazo, los cambistas utilizan la expresión compra / venta <outright> si se trata de una operación a plazo simple. Las operaciones <outright> pueden constituir una garantía de cambio; sin embargo, sino tienen carácter comercial o financiero, vienen a ser operaciones de índole especulativa.

2.4.3.1.- Tipo de cambio a plazo, prima y descuento.

En los tipos de cambio o cotizaciones a plazo los bancos trabajan únicamente con las diferencias entre los precios al contado y los precios a plazo, expresadas en decimales de la moneda en que se cotiza. Dicho de otro modo, trabajan con los <reports> y los <deports>; también se utilizan para expresar esta diferencia la expresión <tipo de swap>.

El hecho de que utilicemos la diferencia entre la cotización a plazo y la cotización al contado (<report> y <deport>), tiene su justificación. Los <report> y los <deport> permanecen con frecuencia inalterados. La cotización a plazo puede, en muchos casos, no tener ninguna trascendencia; para destacar que se habla de un tipo de cambio a plazo se utiliza la expresión <tipo outright>.

El siguiente cuadro muestra cómo los tipos al contado y los tipos de <swap> aparecen en el tablón de cotizaciones. Los tipos de <swap> se expresan siempre en decimales de la correspondiente divisa con respecto al dólar (al igual que en el sistema utilizado para indicar el cambio al contado de una moneda con respecto al dólar). Suponiendo que los tipos de cambio el 18 de febrero eran los siguientes:

	$ / fl. h.		$ / l. it.	
Al contado	2.3660-	2.3690	1032.00-	1035.00
1 mes	155-	150	0.00-	50
2 meses	300-	295	50-	2.00
3 meses	445-	440	2.00-	4.00
6 meses	870-	855	4.00-	8.00
12 meses	1440-	1410	15.00-	23.00

Se obtiene el cambio a plazo sumando el <report> con el cambio al contado, o restando el <deport> de este último. Aunque los tipo de <swap> no van precedidos del signo + o del signo -, el cambista ve en seguida que el dólar es cotizado con un <deport> con respecto al florín holandés, pero lo es con un <report> con respecto a la lira.

Ya se trate de una operación al contado o a plazo, el precio de compra debe ser siempre inferior al precio de venta. Además, necesitamos para las cotizaciones a plazo un margen algo más amplio que para las cotizaciones al contado. Se obtiene cuando, en caso del <deport>, la cifra del lado de la compra es mayor que la del lado de la venta y viceversa en caso de <report>.

Por ejemplo:

$ / fl.h. cambio al contado	2.3660-2.3690
- <deport> a 3 meses	445- 440
= cambio en plazo	2.3215-2.3250

El margen es de 30 puntos decimales en la cotización al contado, pero de 35 puntos decimales en el tipo a plazo.

Por otra parte, la diferencia de 2.00-4.00 par el dólar a 3 meses con respecto a la lira indica un <report>.

$ / l.it. cambio al contado	1032.00-1035.00
+ <report> a 3 meses	2.00- 4.00
= cambio a plazo	1034.00-1039.00

El margen es de 3.00 para la cotización al contado, pero asciende a 5.00 para la cotización a plazo.

Así como las condiciones del interés de las monedas de que se trate harán que una moneda a plazo sea tratada con prima o con descuento, también la relación del interés determinará el grado de desviación entre la cotización al contado y la cotización a plazo. Lo decisivo no es el nivel del interés de una moneda, sino la diferencia del interés de las dos monedas tratadas en los mercados financieros (internacionales). Ello está en relación con el hecho de que las operaciones a plazo originan en el banco posiciones abiertas; si éste quiere cubrirlas para eliminar los riesgos provenientes del cambio y de la liquidez, se requerirán las consiguientes transacciones que implican gastos o ganancias.

Las monedas extranjeras remuneradas con tipos de interés más altos son tratadas a una cotización a plazo inferior a la cotización al contado = <deport>.

La cotización a plazo de la moneda remunerada con un tipo de interés más bajo se sitúa siempre por encima de la cotización al contado = <report>.(Ej.: el banco se endeuda a tipos de interés bajo e invierte en la contramoneda remunerada a tipo de interés más alto, de lo cual resulta un beneficio neto obtenido a través de la cotización más alta.)

Matemáticamente, se calculan los <deport> y <report> mediante las siguientes fórmulas:

$$\text{Swap} = \frac{\left[\left(\text{tipo de depósito moneda b} \times \text{spot} \right) - \left(\text{tipo de depósito moneda a} \times \text{cotización a plazo aproximada} \right) \right] \times \text{plazo}}{100 \times 360}$$

donde: moneda a = moneda cotizada

moneda b = contramoneda

$$\text{Cotización a plazo aproximada} = \text{cotización a plazo} + \frac{(\text{moneda de depósito b} - \text{moneda de depósito a}) \times \text{spot} \times \text{plazo}}{100 \times 360}$$

Si el tipo de interés de la contramoneda es menor que el de la moneda cotizada, el swap será de signo negativo y la cotización a plazo será inferior a la cotización al contado. En tales casos resultan <deports>. En el caso de que el tipo de interés de la moneda "b" sea mayor que el de la moneda "a", se registra un <report>.

El comercio internacional crea constantemente una necesidad de operaciones a plazo para asegurarse contra los riesgos del cambio.

Pueden proceder de operaciones no comerciales haberes en divisas extranjeras cuyo cambio haya de asegurarse:

- Las compras de títulos, las colocaciones en el mercado monetario, los préstamos a filiales extranjeras, las inversiones directas, etc., constituyen haberes en divisas si estas operaciones se han efectuado en monedas extranjeras y, para eliminar el riesgo proveniente del cambio, se pueden vender a plazos los importes correspondientes.
- Los capitales tomados a préstamo en los mercados financieros extranjeros constituyen compromisos en moneda extranjera, y también en este caso se puede eliminar el riesgo inherente al cambio, comprando a plazo las sumas de moneda correspondientes.

También se puede eliminar el riesgo inherente al cambio mediante operaciones a corto plazo, incluso cuando son transacciones a plazos medio o largo las que han creado esta necesidad. Resulta difícil concluir operaciones a plazo de más de doce meses con la mayoría de las monedas. No obstante, se puede cubrir el riesgo monetario que una operación a largo plazo implica,

renovando regularmente, por ejemplo cada doce meses, un contrato a plazo. Claro está que, en tal caso, sólo se conocen los costos del seguro para el primer período, pues los correspondientes a los períodos siguientes no se pueden conocer de antemano; pero esto no debe ser un motivo para renunciar a cubrirse contra los riesgos.

Los tipos de cambio cruzados (<cross rates>) se calculan en las operaciones a plazo de la misma manera que en las operaciones al contado.

2.4.3.2.- Operaciones <Swap>.

Las operaciones del mercado monetario, sobre todo cuando se trata de exportaciones de capital a corto plazo, van con frecuencia vinculadas a operaciones <swap>. Si la exportación de capital implica la colocación en otra moneda, y si esta colocación ha de cubrirse contra los riesgos del cambio, debemos concluir una operación <swap>. Como hemos dicho, se entiende por <swap> la combinación de una compra al contado con una venta simultánea a plazo (o viceversa) de determinada moneda.

Existen dos razones principales para efectuar colocaciones a corto plazo en el extranjero:

- O bien el mercado monetario nacional no ofrece posibilidades de colocación apropiadas para la liquidez a corto plazo.
- O bien la colocación en otros países y en otras monedas ofrece un rédito mayor, aunque haya de cubrirse la operación contra el riesgo inherente al cambio.

Contrariamente a una opinión muy extendida, la cobertura del riesgo proveniente del cambio no origina necesariamente gastos. Normalmente se aseguran las monedas <débiles> frente a las <fuertes>, comprando las débiles a plazo; y para mucha gente, son débiles aquellas monedas que experimentan un descuento. En tales casos, la cobertura implica gastos. Sin embargo, ha sucedido ya con frecuencia que la moneda aparentemente <débil> se ha robustecido, mientras que la moneda considerada <fuerte> (moneda con prima) se ha debilitado.

2.4.3.3.- Operaciones <outright> de carácter especulativo.

Las operaciones <outright> son sencillas operaciones a plazo, es decir, operaciones a plazo que no forman parte de una operación <swap>. De las explicaciones precedentes se infiere que las operaciones <outright> pueden constituir una cobertura para una posición en moneda extranjera. Ahora bien, conviene señalar que las operaciones <outright> pueden no tener una base comercial o financiera, y tener en tal caso una finalidad exclusivamente especulativa. En efecto, las operaciones <outright> se prestan muy bien a la actividad especulativa, pues -- por lo menos en principio -- no se requiere para tales transacciones disponer de medios líquidos. No obstante, los bancos exigen, en la mayoría de los casos un determinado margen de cobertura. Si el especulador compra a plazo moneda X por moneda Y, no tiene la intención de tomar moneda X al vencimiento pagándola realmente con moneda Y, sino que se propone compensar la operación a plazo, al llegar su vencimiento, con una operación al contado, ello con la esperanza de que la moneda X se haya encarecido entre tanto.

No obstante, las experiencias de estos últimos años han demostrado que, bajo el régimen del flotamiento de las divisas, la evolución de los tipos de cambio contradice con frecuencia las previsiones. Las operaciones <outright> pueden originar tanto pérdidas considerables como apreciables ganancias. Este es, en primer lugar, asunto que interesa a los especuladores (ya sean estos particulares, firmas o bancos), pero también reviste una importancia considerable para el banco que actúa de contraparte.

Si el cliente no está en condiciones de cumplir su obligación, es el banco el que ha de cargar con la pérdida. Así pues, el banco sufrirá pérdidas en el comercio de divisas, ello no ya porque él mismo estuviera especulando, sino a causa de la insolvencia del cliente. Esta es la razón por la cual un banco prudente examinará con detenimiento la solvencia de las personas con las cuales puede ser llamado a efectuar operaciones de cambio a plazo, y exigirá una cobertura suficiente en todos aquellos casos en que lo estime necesario. Cuando se exige un margen de seguridad al operar con monedas flotantes, ello se hace tanto si dicha moneda se compra como si se vende, puesto que el riesgo existe en ambos sentidos para todas las monedas.

Lo que les interesa a las casas de cambio de divisas no son los tipos de cambio a plazo, sino la diferencia entre los tipos de cambio a plazo y los tipos de cambio al contado, es decir, el descuento o la prima. No hay por qué extrañarse de que el tipo de cambio al contado y el tipo de cambio a plazo de una moneda evolucionen paralelamente. Lo que interesa es saber por qué la diferencia entre los dos tipos de cambio es más o menos grande de un día a otro, es decir, por qué y cuándo el descuento o la prima son más o menos altos.

2.4.4.- Operaciones del mercado monetario.

Hemos demostrado la existencia de numerosas relaciones recíprocas entre el mercado monetario y el de divisas, sobre todo entre los tipos del interés y el mercado de divisas a plazo.

Numerosas transacciones ofrecen, pues, al mismo tiempo, aspectos que las relacionan con el mercado monetario y con el de divisas.

Todos los grandes bancos intervienen activamente en las operaciones del mercado monetario que consiste en aceptar depósitos de fondos (a plazo fijo) y en volver a distribuirlos en forma de préstamos. Por diversas razones, sucede con frecuencia que, por ambas partes, se opera con monedas diferentes, lo cual conduce a efectuar operaciones <swap>.

Aunque un banco no se ocupe de operaciones del mercado monetario tendrá que operar en el mercado monetario desde el momento en que opere con posiciones en divisas al contado.

Una operación del mercado de divisas estrechamente ligada al mercado monetario se da cuando un cliente (o banco) desea concluir negocios con fecha valor a 1 día, y no con fecha valor normal (el segundo día laborable), lo cual es perfectamente posible, pero a un cambio algo diferente del cambio al contado. (Los cambios no son idénticos más que cuando las dos monedas en cuestión tienen exactamente los mismos tipos de interés).

2.4.4.1.- Arbitraje del tipo de interés.

En el origen, se entendía por <arbitraje del tipo de interés> la aceptación de depósitos y el préstamo, por la misma duración y a un tipo de interés más alto, de las cantidades así aceptadas. Ahora bien, como quiera que el margen de interés es muy pequeño en esta clase de operaciones, este sector de actividad viene limitado por consideraciones relacionadas con la estructura del balance.

En efecto, los bancos no quieren engrosar sus balances con operaciones de poco rendimiento y que exigirían costosos aumentos de capital.

Siempre se ha tendido a obtener beneficios de las diferencias de los tipos de interés activos y pasivos, combinando transacciones a largo plazo con otras a corto plazo, como por ejemplo préstamos a largo plazo con aceptación de préstamos a corto plazo. Esto viene a ser una forma de arbitraje del tipo de interés. Como es sabido, las operaciones de esta clase son tan atractivas como peligrosas.

2.4.4.2.- Operación a plazo por operación a plazo.

Una última operación llamada <operación a plazo por operación a plazo> (forward against forward) viene a ser la alta escuela en el campo del arbitraje de los tipos de interés. Se basa en la estimación de la evolución futura, no de una moneda, sino de la diferencia del tipo de interés de dos monedas. Por ejemplo, vendiendo la moneda A a plazo fijo (pongamos por caso a seis meses), a cambio de la moneda B, y volviendo a comprar simultáneamente la moneda A a cambio de la moneda B, pero a plazo más corto (por ejemplo a un mes), se especula sobre la modificación futura de la prima y del descuento (diferencia de los tipos de interés) durante el resto del plazo de la primera operación (cinco meses) después del vencimiento de la operación a corto plazo, además, no existe ningún riesgo proveniente del cambio. La elección del plazo (p. ej. 12 meses por 6 meses, ó 6 meses por 1 mes) depende, como es natural, de las fechas para las cuales se espera que se produzca una evolución divergente de los tipos de interés. En la práctica, resulta más fácil operar, en vez de con dos operaciones <outright>, con dos operaciones <swap>, cuyas transacciones al contado se compensan.

CAPÍTULO III

ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS

El desarrollo de un sistema siempre inicia con la etapa de análisis y definición de requerimientos sin importar el modelo de ingeniería de software que se elija para apoyar dicho desarrollo.

El resultado fundamental del análisis y definición de requerimientos es el de conocer la situación del negocio, más no el de proponer soluciones a los problemas. Esta etapa facilita al ingeniero de sistemas la especificación de la función del software, la descripción de la interfase con otras entidades del sistema y el establecimiento de las restricciones de diseño que se deben considerar.

El análisis debe orientarse esencialmente a hacer una evaluación del problema y a sintetizar una solución general; para tal efecto, el analista necesitará conocer perfectamente el flujo y estructura de la información asociada al sistema, deberá definir y elaborar todas las funciones del software, así como especificar las características de la interfase del sistema.

Cada una de las actividades que se realizan durante el análisis debe quedar plasmada en un documento que puede ser revisado y autorizado por el cliente y que además sirva como apoyo a las etapas subsecuentes del desarrollo del sistema.

En términos generales, el análisis se compone por tres actividades cuyos productos se mencionan a continuación:

ACTIVIDAD	PRODUCTO
1.- Definición de requerimientos	Especificación de requisitos generales.
2.- Análisis estructurado	Diagramas de flujo de datos
	Diccionario de requisitos.
3.- Modelización de datos	Diagrama de entidad relación.

3.1.- Definición de requerimientos.

Se entiende por requerimientos todas aquellas necesidades que no son cubiertas por el sistema actual, esté automatizado o no dicho sistema.

Los requerimientos que pudieran llegar a tenerse en un sistema de información en operación serían por ejemplo la forma en que se están recibiendo o procesando los datos, información para ciertos usuarios que están careciendo de ella, control en alguna actividad especifica, apoyo para toma de decisiones, etc.

Con el fin de detectar todos y cada uno de los requerimientos es necesario estudiar a fondo tanto la operación del negocio, como el funcionamiento del sistema actual. Para llevar a cabo dicho estudio existen varios métodos tales como entrevistas con los usuarios, cuestionarios, observación física de la operación diaria, etc.

3.2.- Descripción de la operación.

Del estudio de la operación de la casa de cambio pudimos detectar seis procesos principales que son descritos a continuación:

3.2.1.- Catálogos.

Existe en la casa de cambio una gran cantidad de información que por sus características de poca variabilidad en el tiempo es manejada como catálogos. La empresa requiere de un control sobre la información acerca de sus clientes recurrentes, de los bancos con los que trabaja a nivel mundial, de las divisas y sus características, de los usuarios y actividades, etc.

3.2.2.- Formas de Compra-Venta (DEAL'S)

Toda operación de compra-venta que realiza la casa de cambio debe quedar soportada por un "Deal". Un deal es una forma preimpresa en donde se especifica el nombre del cliente, la fecha, tipo de operación (compra c venta), la divisa, tipo de cambio entre otros datos. Dicha forma debe cumplir con las disposiciones de la Secretaría de Hacienda en cuanto a contener el registro federal de contribuyentes de la compañía, así como la dirección de la matriz o de la sucursal donde se haya realizado la operación.

El control de los deal's es de vital importancia para la casa de cambio ya que la pérdida de alguno de ellos podría derivar en una operación fraudulenta a la que la empresa debería hacer frente.

La compañía requiere del control de deal's desde que los recibe el impresor hasta que son utilizados en cualquiera de las sucursales. Es muy importante conocer la ubicación de un deal en un momento dado, esto es, si está en almacén, en matriz o en cualquiera de las sucursales, así como el estado en el que se encuentran, es decir, si ya fué usado en alguna operación, si fué inutilizado por algún error o si esta disponible para usarse.

3.2.3.- Operación de compra - venta.

Las operaciones de compra - venta que realiza una casa de cambio de divisas se divide en tres áreas de aplicación:

a) Operaciones interbancarias: Este tipo de operaciones, como su nombre lo indica se llevan a cabo con otras instituciones bancarias ya sean nacionales o extranjeras. Las operaciones interbancarias se realizan mediante transferencias de fondos entre cuentas que tiene la compañía en diferentes bancos alrededor de todo el mundo incluyendo por supuesto bancos mexicanos.

Para la casa de cambio que estamos estudiando, las transacciones interbancarias se hacen mediante conversaciones telefónicas, fax o telex.

b) Operaciones de promoción: Son aquellas operaciones que se realizan con compañías que recurrentemente solicitan un servicio de la casa de cambio y que generalmente es por montos relativamente grandes.

c) Operaciones de mercado de menudeo: Son aquellas operaciones de montos pequeños que solicitan clientes que esporádicamente acuden a la casa de cambio.

En función de la divisa que en un momento dado le es solicitada a la casa de cambio, las operaciones de venta se dividen en tres tipos:

1) Venta de monedas: Son aquellas operaciones donde la casa de cambio vende cualquier moneda exceptuando el dólar.

2) Venta del dólar americano: Por la importancia que reviste el dólar americano en el mercado de divisas a nivel mundial, es común que se le trate separado a las otras monedas.

3) Metales: Son las operaciones de venta donde a la casa de cambio se le solicita cualquier tipo de metal con valor financiero como el oro, la plata, etc.

Es importante aclarar en este punto que los cambistas generalmente están especializados en alguno de los tipos de venta mencionados anteriormente y que su actividad está restringida a sólo ese tipo de venta.

Para la casa de cambio es esencial el registro y control de cada una de las operaciones de compra - venta que realiza, ya que éstas servirán de soporte en las interfases con contabilidad y tesorería.

El registro de cada operación que se realiza deberá actualizar la posición de cada divisa como fué explicado en el capítulo 2 del presente trabajo.

3.2.4.- Proceso de autorizaciones.

Cada uno de los cambistas que laboran en la compañía tiene un límite en cuanto al monto de las operaciones que realizan. Si una operación en particular excede el límite otorgado al cambista, este debe solicitar una autorización para llevar a cabo la operación.

Existen diferentes niveles de autorizaciones, desde los encargados o gerentes de sucursales hasta los directivos de la empresa.

3.2.5.- Tesorería.

Dentro de la casa de cambio existe una área de tesorería que se encarga del manejo y control del flujo de efectivo.

Entre sus funciones principales está la de invertir los saldos al final del día de las cuentas que tiene la empresa en los bancos en todo el mundo.

3.2.6.- Contabilidad.

En esta área se lleva a cabo la contabilización de todas las operaciones que se realizan en la casa de cambio.

El origen de esta contabilización se encuentra precisamente en el registro de las transacciones de compra - venta.

El área de contabilidad interna de la casa de cambio debe reportar todos sus resultados al área de contabilidad del grupo financiero; actualmente este reporte se hace a través de listados impresos que sirven como base para captura en el grupo.

3.3.- Requerimientos generales.

En base a una serie de entrevistas con diferentes usuarios y después de varios días de permanecer en la casa de cambio llegamos a adquirir un conocimiento de la operación de la empresa que nos permitió definir los requerimientos generales para cada uno de los procesos establecidos en la sección anterior:

1.- Registro y control de Catálogos.

- Registro y control de clientes recurrentes.
- Registro y control de bancos asociados.
- Registro y control de divisas y sus características.
- Registro y control de las divisiones y sucursales de la empresa.
- Registro y control de las cuentas bancarias de la empresa.
- Registro y control de autorizadores de operaciones especiales.
- Registro y control de los usuarios del sistema.
- Registro y control de las cuentas contables.

2.- Registro y control de Deal's.

- Registro y control del stock de deal's en el almacén.
- Registro y control de las distribuciones de deal's a las sucursales.
- Registro y control de uso de los deal's.

3.- Registro y control de operaciones.

- Registro y control de operaciones de venta.
- Registro y control de operaciones de compra.
- Actualización y conocimiento de la posición por divisa.
- Registro y control de operaciones de futuros.
- Registro y control de garantías de operaciones de futuros.
- Actualización de saldos en cuentas bancarias.

4.- Registro y control de autorizaciones.

- Control de operaciones de montos considerables.
- Registro y control de autorizaciones otorgadas por funcionarios.

5.- Control de flujo de efectivo.

- Conocimiento de saldos en cuentas bancarias.
- Registro de inversiones por divisa.
- Actualización de la posición por divisa por inversiones.
- Interfase con tesorería corporativa.

6.- Registro y control de contabilidad.

- Control de la contabilización de las operaciones realizadas.
- Generación automática de los registros contables en base a las operaciones realizadas.

Interfase con el área de contabilidad corporativa.

3.4.- Análisis estructurado.

Al igual que los métodos clásicos de análisis de requisitos, el análisis estructurado es una actividad que tiene como fin la construcción de un modelo que simule el comportamiento del sistema.

Para llevar a cabo su objetivo el análisis estructurado se vale de una notación única que permite comprender perfectamente al modelo. Dicho modelo debe reflejar el flujo y el contenido de la información, fragmentando el sistema funcionalmente y según los distintos comportamientos, se establece la esencia de lo que se debe construir.

3.4.1.- Diagrama de flujo de datos.

La herramienta de modelado que utilizamos para describir la transformación de entradas a salidas de información es un diagrama de flujo de datos.

Los diagramas de flujo de datos consisten en procesos, agregados de datos, flujos y terminadores:

- Los procesos se representan por medio de bloques o rectángulos en el diagrama. Representan las diversas funciones que el sistema lleva a cabo. Las funciones transforman entradas en salidas.
- Los flujos se muestran por medio de flechas. Son las conexiones entre los procesos y representan la información que generan como salida.
- Los agregados de datos se representan por medio de rectángulos no cerrados. Muestran colecciones de datos. Estas colecciones de datos derivarán en archivos o bases de datos en la etapa de diseño.

FIGURA 3.1 NOTACION BASICA

- Los terminadores muestran las entidades externas con los que el sistema se comunica. Típicamente se trata de individuos o grupos de personas, sistemas de cómputo externos y organizaciones externas.

La figura 3.1 ilustra la notación básica que se usa para crear un DFD.

Aunque el diagrama de flujo de datos proporciona una visión global suficientemente clara de los componentes funcionales del sistema, no da detalles de éstos. Para mostrar detalles acerca de qué información se transforma y de cómo sucede dicha transformación, se ocupa una herramienta textual de modelado: el diccionario de requisitos.

3.4.2.- Diccionario de requisitos.

El diccionario de requisitos es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de almacenamiento y cálculos intermedios. El diccionario de requisitos define su objetivo haciendo lo siguiente:

- Describe el significado de los flujos y almacenamientos que se muestran en los DFD.
- Describe la composición de los paquetes de datos en los almacenamientos.
- Especifica los valores y unidades relevantes de piezas elementales de información en los flujos de datos y en los almacenamientos de datos.

El diccionario de requisitos es pues utilizado como gramática casi formal para describir el contenido de los objetos definidos durante el análisis estructurado.

3.4.2.1 Diagrama de flujo de datos del software para la casa de cambio de divisas y su diccionario de requisitos.

Al realizar el análisis de las entradas y salidas del sistema se definieron las entidades externas al sistema que guardan cierta relación con él.

En el diagrama de flujo de datos inicial se presenta al sistema como un solo proceso con las entidades externas. Esta representación se puede observar en la figura 3.2.

- Banco de México: esta institución impone reglamentaciones a las casas de cambio nacionales que de alguna manera afectan a su operación. La empresa cambiaria debe informar diariamente al Banco de México la posición con la que cerró en las principales monedas, así mismo el Banco de México sirve en ocasiones como intermediario en transacciones interbancarias.
- Clientes interbancarios: son bancos nacionales o internacionales que solicitan el servicio de la casa de cambio.
- Clientes de promoción: son empresas relativamente importantes que recurrentemente acuden a la casa de cambio a solicitar divisas extranjeras que generalmente utiliza para importación de productos.
- Clientes de menudeo: son clientes que esporádicamente hacen compras / ventas de divisas o metales y generalmente es por montos pequeños.
- Contabilidad: esta área interna de la empresa requiere de la información de las operaciones realizadas para su contabilización.
- Tesorería: esta es también una área interna de la empresa que requiere conocer los saldos de las cuentas bancarias para su inversión en diferentes instrumentos. Los saldos de dichas cuentas bancarias se están ajustando constantemente en cada operación de compra - venta.

FIGURA 3.2 REPRESENTACION INICIAL ENTIDADES EXT.

- Grupo Financiero: el grupo financiero al que pertenece la casa de cambio requiere un flujo de información constante por parte de la empresa en relación a las ventas, compras, posiciones, utilidades, inversiones, contabilización, etc.
- Operación sucursales: esta entidad representa a las sucursales tanto del área metropolitana como del interior de la república que harán uso del sistema a través del registro de sus operaciones.
- Auditoria: esta área de la empresa tiene como misión el verificar que no existan desviaciones entre lo registrado y lo realmente operado.

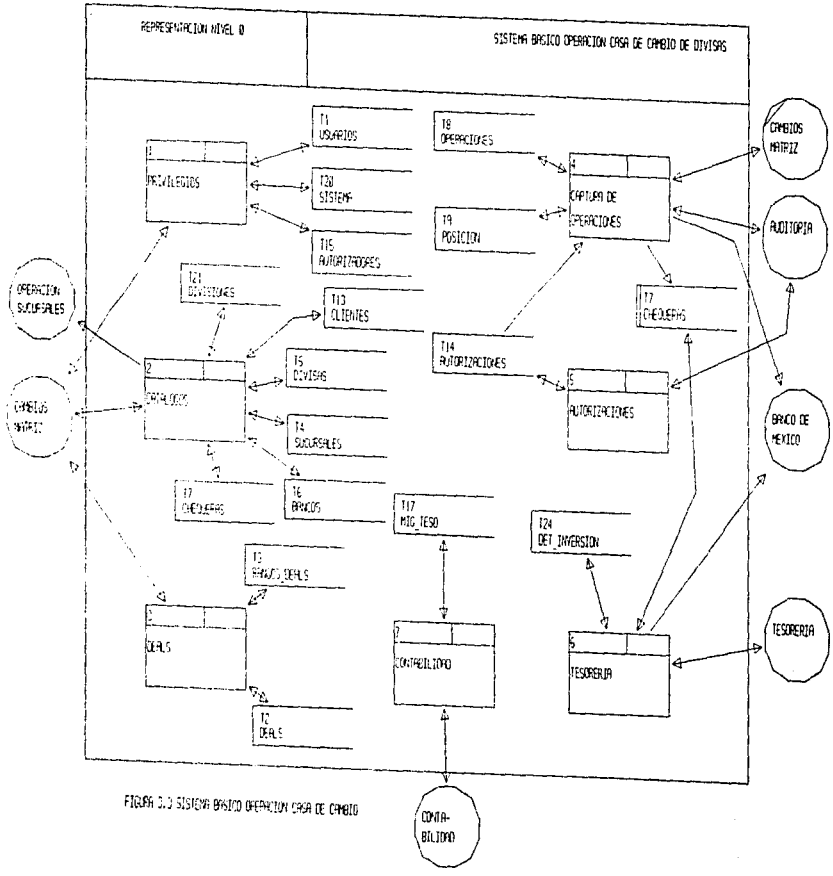


FIGURA 3.3 SISTEMA BASICO OPERACION CASA DE CAMBIO

Proceso principal: 1.- PRIVILEGIOS.

Descripción: proceso principal de registro y control de los principales catálogos de control del sistema.

Integrado por los **subprocesos siguientes**:

- **Proceso 1.1- Altas, bajas y cambios:** proceso de mantenimiento a los catálogos de control de privilegios asignados a los usuarios del sistemas.
- **Proceso 1.2- Reportes y consultas:** proceso de explotación de los datos de asignación de privilegios para control del sistema.

Estructuras de datos asociadas:

T1.- USUARIOS: Contiene los datos necesarios para el acceso al sistema, ubicación y especialidad del cambista.
T20.- SISTEMA: Mediante un parámetro se controla el nombre de la casa de cambio.
T15.- AUTORIZADORES: Catálogo de los funcionarios con capacidad de emitir autorizaciones.

FIG 3.4 PROC MANTENIMIENTO CATALOGOS PRIVILEGIOS

Proceso principal: 2.- CATÁLOGOS.

Descripción: proceso principal de registro y control de los catálogos del sistema.

Integrado por los **subprocesos siguientes:**

- **Proceso 2.1- Altas, bajas y cambios:** proceso de mantenimiento a todos los catálogos que componen el sistemas.
- **Proceso 2.2- Reportes y consultas:** proceso de explotación de los datos contenidos en el módulo de Catálogos.

Estructuras de datos asociadas:

T4.- SUCURSALES: Catálogo de las sucursales con que cuenta la casa de cambio.
T5.- DIVISAS: Descripción de las características de las distintas divisas que maneja la empresa.
T6.- BANCOS: Contiene una lista de los principales bancos nacionales y extranjeras con los que tienen cuentas de cheques.
T11.- CUENTAS: Catálogo de cuentas proporcionado por la Comisión Nacional Bancaria.
T12.- ESPECIES: Catálogo de las maneras en que pueden ser efectuados los pagos.
T13.- CLIENTES: Contiene a los principales clientes que reciben servicios de la casa de cambio en forma recurrente.
T19.- PUESTOS: Contiene los puestos de los colaboradores de la casa de cambio.
T21.- DIVISIONES: Contiene la agrupación de las sucursales existentes.

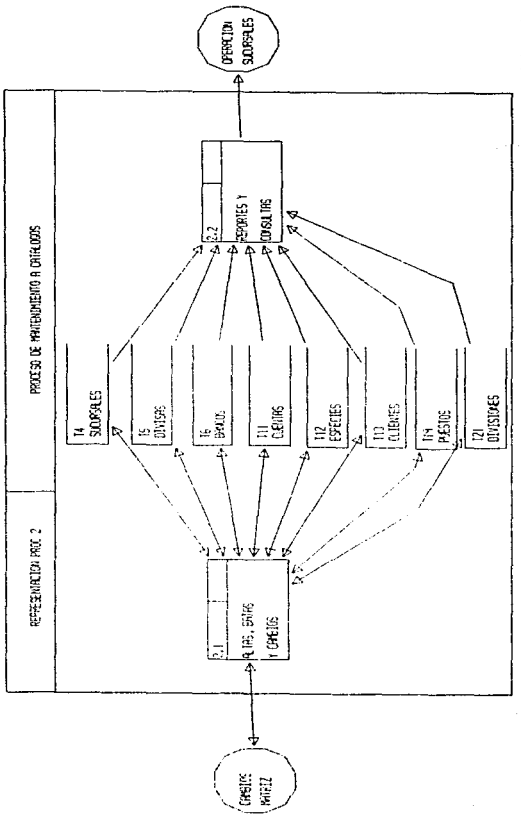


FIGURA 3.5 PROCESO DE MANTENIMIENTO A CATALOGOS

Proceso principal: 3.- DEALS

Descripción: proceso principal de registro y control de los deal's utilizados en la captura de las operaciones de compras y ventas de divisas.

Integrado por los **subprocesos siguientes:**

- **Proceso 3.1- Generación de deal's:** proceso de generación de los deal's entregados por el proveedor y que deben existir físicamente en el almacén.
- **Proceso 3.2- Distribución de deal's:** proceso de distribución de dotaciones de deal's asignados a cada una de las sucursales.
- **Proceso 3.3- Inutilización de deal's:** proceso en el cual se marcan los deals que ha sufrido algún daño por lo que no se llegarán a utilizar.
- **Proceso 3.4- Cambios:** proceso en el que se puede modificar la información ya generada, distribuida o inutilizada.
- **Proceso 3.5- Reportes y consultas:** proceso de explotación de la información contenida en el módulo de deal's.

Estructuras de datos asociadas:

T2.- DEALS: Contiene el detalle de las formas deal's asignadas a la sucursal.
T3.- RANGOS_DEALS: Contiene los rangos distribuidos y por distribuir de las formas deal's.

FIGURA 3.6 PROCESO DE REGISTRO Y CONTROL DE DEALS

Proceso principal: 4.- OPERACIONES

Descripción: proceso principal de registro y control de las operaciones realizadas en compra - venta de divisas.

Integrado por los **subprocesos siguientes:**

- **Proceso 4.1- Alta de operaciones compra - venta:** proceso de registro detallado de las transacciones efectuadas en la casa de cambios con fecha valor hasta 48 horas, saldos en chequeras, inversiones y stocks por divisa.
- **Proceso 4.2- Actualizaciones:** proceso de modificación de la información registrada en las principales tablas del sistema.
- **Proceso 4.3- Control de futuros:** proceso de registro detallado de las operaciones realizadas en la casa de cambios a largo plazo.
- **Proceso 4.4- Reportes y consultas:** proceso de explotación de la información contenida en el módulo de operaciones.

Estructuras de datos asociadas:

T8.-OPERACIONES: Contiene el detalle de las operaciones registradas en compras y ventas, así como, inversiones realizadas por Tesorería.
T9.- POSICIÓN: Muestra la posición global de la compañía por divisa.
T7.-CHEQUERAS: Contiene las cuentas de la empresa en los bancos nacionales y extranjeros con sus respectivos saldos.
T16.- FUTUROS: Aquí se guardan los registros de transacciones a largo plazo mientras no se confirme que sí se efectuará la operación.
T22.- INVERSIONES: Contiene el control de tasas de interés obtenidas para cada chequera.

T23.- STOCK_DIVISA: Proporciona el control de la cantidad de divisas en cada sucursal.

T24.- DET_INVERSION: Mantiene el detalle de las inversiones.

FIGURA 3.7 PROC. REGISTRO Y CONTROL DE OPERACIONES

Proceso principal: 5.- AUTORIZACIONES

Descripción: proceso principal de registro y control de las autorizaciones requeridas en la operación de la casa de cambios al sobrepasar los límites definidos para los cambistas.

Integrado por los **subprocesos siguientes:**

- **Proceso 5.1- Alta de autorizaciones:** proceso de generación de claves de autorización para el registro de operaciones de las sucursales.
- **Proceso 5.2- Reportes y consultas:** proceso de extracción de la información disponible en el módulo de autorizaciones.

Estructuras de datos asociadas:

T14.- AUTORIZACIONES: Se registran las autorizaciones asignadas para el control de las operaciones de la casa de cambio.

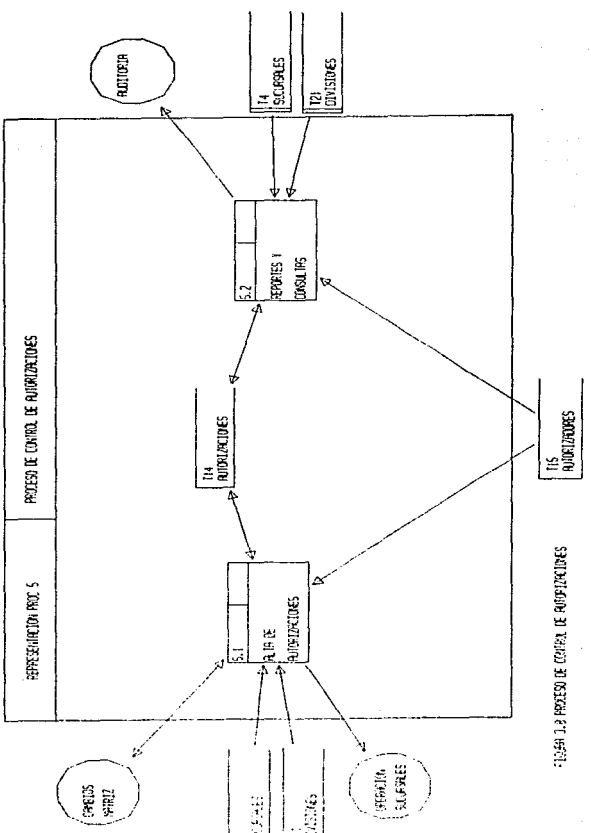


FIGURA 3.3 PROCESO DE CONTROL DE AUDITORIAS

Proceso principal: 6.- TESORERÍA

Descripción: proceso principal de registro y control de saldos en chequeras originados por las operaciones de compra - venta e inversiones realizadas.

Integrado por los **subprocesos siguientes:**

- **Proceso 6.1- Impresión de cheques:** proceso de control e impresión de cheques.
- **Proceso 6.2- Procesos de información corporativa:** proceso de obtención de la información registrada en la operación diaria de la casa de cambio que será migrada al grupo financiero.
- **Proceso 6.3-** Reportes y consultas: proceso de explotación de la información contenida en el módulo de inversiones.

Estructuras de datos asociadas:

T7.-CHEQUERAS: Contiene las cuentas de la empresa en los bancos nacionales y extranjeros con sus respectivos saldos.

FIGURA 3.9 PROCESOS DE CONTROL PARA TESORERIA

Proceso principal: 7.- CONTABILIDAD

Descripción: proceso principal de registro y control de los movimientos contables originados por la captura diaria de las operaciones.

Integrado por los **subprocesos siguientes:**

- **Proceso 7.1-** Procesos de información corporativa: proceso de obtención de la información registrada en la operación diaria de la casa de cambio que será migrada al grupo financiero.
- **Proceso 7.2-** Mantenimiento a fórmulas: proceso de agrupación de los movimientos contables necesarios en fórmulas para el registro en contabilidad de las operaciones de la casa de cambio.
- **Proceso 7.3-** Reportes y consultas: proceso de explotación de la información contenida en el módulo de contabilidad.

Estructuras de datos asociadas:

T10.- MIG_CONTA: Información contable debida a la operación diaria que será migrada al grupo financiero.
T18.- FORMULAS: contiene las cuentas agrupadas y asientos necesarios para la presentación de la información en formatos establecidos.

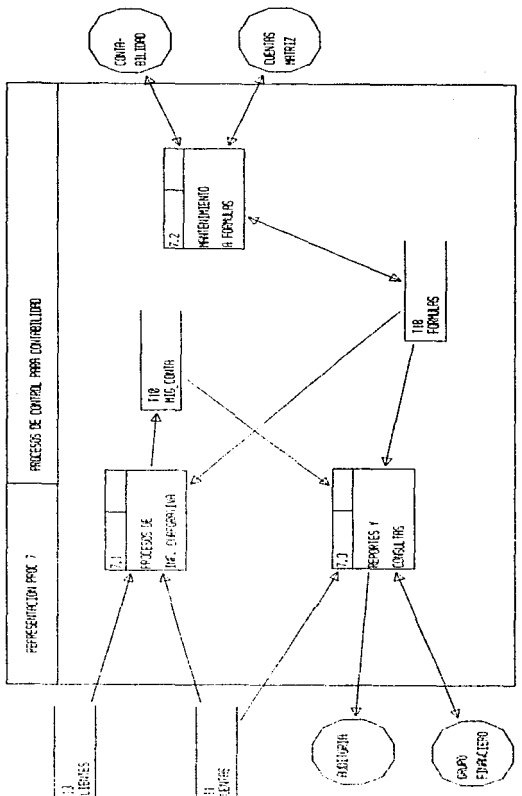


FIGURA 3.10 PROCESO DE CONTROL PARA CONTABILIDAD

3.4.3.- Diagrama de entidad - relación.

El diagrama de entidad - relación es un modelo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema.

La importancia de un diagrama de entidad - relación radica en que las estructuras de datos y sus relaciones pueden ser tan complejas que se deseara enfatizarlas y examinarlas independientemente del proceso que se llevará a cabo. De hecho, esto se da sobre todo si mostramos el modelo del sistema a los usuarios ejecutivos de mayor nivel de una organización.

Para la etapa de diseño el diagrama de entidad - relación representa un gran beneficio ya que realza las relaciones entre almacenes de datos en el DFD que de otra forma se hubieran visto solo en la especificación de proceso.

Hay cuatro componentes principales en un diagrama de entidad - relación:

1. Tipo de objetos: se representan por medio de una caja rectangular y ser refieren a una colección o conjunto de objetos cuyos miembros cumplen con las siguientes características:

- cada objeto puede identificarse de manera única por algún medio
- el sistema no puede operar sin tener acceso a dichos objetos
- cada objeto puede describirse por uno o más datos

Los tipos de objetos serán la representación en el sistema de algo material del mundo real.

2. Relaciones: los objetos se conectan entre sí mediante relaciones. Una relación representa un conjunto de conexiones entre objetos y se representa por medio de un rombo.
3. Indicadores asociativos de tipo de objeto: son una notación especial del diagrama de entidad - relación, representan algo que funciona como objeto y como relación, es decir, representan una relación acerca de la cual se desea mantener una información.

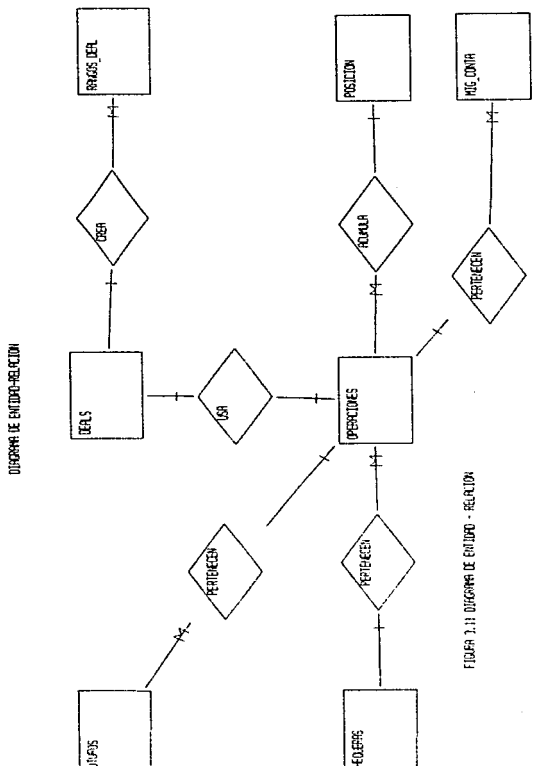


FIGURA 3.11 DIAGRAMA DE ENTIDAD - RELACION

4. Indicadores de subtipo/supertipo: los tipos de objeto subtipo/supertipo consisten en tipos de objeto de una o más subcategorías conectadas por una relación.

3.4.3.1.- Diagrama de entidad - relación para la casa de cambio.

Una vez elaborados el diagrama de flujo de datos y el diccionario de requisitos donde se definieron las entidades que integrarán el sistema de información, procederemos a la elaboración del diagrama de entidad - relación.

La creación de dicho diagrama se inicia considerando todas las tablas que arrojo el diccionario de requisitos. Se analizan detalladamente los datos que representan cada una de las tablas y se definen las relaciones existentes entre ellas. La clasificación de estas relaciones (uno a uno, uno a muchos) se define a partir de la relación de los datos contenidos en las entidades.

En la figura 3.11 se muestra el diagrama de entidad - relación obtenido.

CAPÍTULO IV

DISEÑO DEL SISTEMA

Al iniciar el desarrollo de sistemas en ingeniería del software, nuestros esfuerzos se deben dirigir a la aplicación de diversas técnicas y principios que nos lleven a definir, con suficiente detalle, los procesos que nos permitirán tener un modelo o representación general del sistema que construiremos más adelante. Este primer paso se conoce como la etapa de diseño del sistema.

En este capítulo empezaremos la primera de las etapas dentro del desarrollo de sistemas apoyándonos en los productos que se generaron en la etapa de análisis, tales como los diagramas de flujo de datos, los diagramas de requisitos y el diagrama de entidad-relación.

Ya que se han establecido los requisitos del software, el diseño es la primera de tres actividades técnicas: diseño, implementación y pruebas del sistema. Cada una de estas actividades transforma la información de manera que al final obtenemos un software completamente validado.

Para el diseño del sistema nos daremos a la tarea de crear los productos de software necesarios para obtener una representación adecuada del sistema. Estos productos son el diseño de datos, el diseño arquitectónico y el diseño procedimental.

En las fases de diseño es donde se toman las decisiones que afectarán en primer lugar a la correcta implementación de las aplicaciones y en segundo lugar a la facilidad para realizar el mantenimiento a los programas. De aquí la importancia que tiene la actividad de diseño en el desarrollo del sistema.

El diseño es la única forma que se tiene para poder traducir adecuadamente los requisitos del cliente en un producto o sistema acabado. El diseño de software es la base para las siguientes etapas del desarrollo de sistemas y de las fases de mantenimiento.

4.1.- Diseño de datos.

La actividad inicial en el diseño de un sistema de software es la referente al diseño de datos y consiste en la transformación del modelo de campos de información en las estructuras de datos que se van a requerir para la implementación del software.

Este punto es de suma importancia porque la calidad de nuestro sistema en buena medida dependerá de la correcta identificación y uso de los datos que conformaran las tablas del sistema.

En la medida en que los datos estén bien diseñados nos llevarán a mejores estructuras de programas, reduciendo la complejidad procedimental.

Precisamente los productos de salida de esta actividad son el diccionario de datos y las estructuras del sistema.

4.1.1.- Diccionario de datos.

En los desarrollos actuales, el diccionario de datos juega un papel muy importante porque constituye la entrada o primer actividad por realizar cuando se trabaja con lenguajes de cuarta generación.

Para la determinación del diccionario de datos hay que obtener a partir de los diagramas de flujo de datos los campos que consideramos que son relevantes para ser utilizados en el sistema.

En un diccionario de datos representamos las relaciones entre los datos y las restricciones sobre los elementos de una estructura de datos. En las aplicaciones se pueden definir fácilmente algoritmos que aprovechen las relaciones especificadas en el diccionario de datos.

Una vez que tenemos los datos esenciales, tenemos que listarlos especificando para cada campo sus características principales como su definición y descripción.

La definición del campo debe contener si se trata de un entero, numérico o carácter y de que longitud se compone. La descripción debe ser breve resaltando los detalles finos del campo en cuestión. Estos detalles se refieren a los diversos valores que puede llegar a tener.

A continuación listamos el diccionario de datos para el Sistema de la Casa de Cambio de divisas materia de este trabajo:

CAMPO	DEFINICIÓN	DESCRIPCIÓN
auxiliar	entero 6	Campo utilizado para clasificaciones de cuentas contables
concepto	caracter 40	Descripción del concepto del asiento contable
consecutivo	entero 2	Número consecutivo para los diversos registros de fórmulas
consec_aut	entero 6	Número consecutivo para control de autorizaciones
consec_cte	entero 6	Número consecutivo para control de clientes
control_inv	entero 6	Número consecutivo para el detalle de las inversiones
cotizacion_compra	numérico 7.6	Cotización de compra
cotizacion_venta	numérico 7.6	Cotización de venta
cve_baja	carácter 1	Clave para saber si se hizo la operación S-si,N-no,P-pend
cve_futuro	carácter 1	Clave para determinar si la operación fue un futuro S / N
debe_haber	carácter 1	Clave para distinguir si es un asiento de D-debe / H- haber
descripcion	carácter 40	Descripción del día festivo
desde	entero 7	Número de inicio para un rango de deals (formas de operación)
dia	entero 2	Número de día del mes
dias_invertido	entero 2	Número de días en que se invirtió un importe excedente
direccion1	carácter 40	Descripción de la dirección del cliente
direccion2	carácter 40	Descripción de la dirección del cliente
divisa_otorga	entero 3	Número de divisa que se otorga
divisa_recibe	entero 3	Número de divisa que se recibe
division	entero 1	Número de la división operativa de la empresa
equiv_compra	numérico 11.4	Importe equivalente en moneda nacional de las compras
equiv_venta	numérico 11.4	Importe equivalente en moneda nacional de la ventas
equivalente_mn	numérico 11.4	Importe equivalente en moneda nacional de la posición
especialidad	entero 1	Número para la especialidad del usuario: 1-dólar, 2-monedas, 3-metales, 4-dólar-moneda, 5-dólar-metales, 6-monedas-metales, 7- todo
estado	caracter 1	Clave de estado para el deal: U-usado,I-inutilizado,D-disponible
estatus	carácter 1	Clave para el estado del asiento: A-alta, B-baja
ext	entero 5	Número de extensión telefónica
fax	entero 9	Número telefónico del fax
fecha_cheque	fecha	Fecha en la que se imprimió el cheque
fecha_conta	fecha	Fecha en la que se migran los registros a contabilidad
fecha_dist	fecha	Fecha de distribución de los deals a las sucursales
fecha_edo	fecha	Fecha en la que se actualizo el estado del deal
fecha_gen	fecha	Fecha en la que se generaron los deals
fecha_operacion	fecha	Fecha en la que se realizo la operación de compra/venta
fecha_registro	fecha	Fecha de registro
fecha_ult	fecha	Fecha de ultima actualizacion
fecha_valor	fecha	Fecha en la cual las divisas serán abonadas por ambas partes

CAMPO	DEFINICIÓN	DESCRIPCIÓN
hasta	entero 7	Número de fin para un rango de deals
imp_pos_ini	numérico 11.4	Importe de la posición de apertura o inicio del día
imp_posicion	numérico 11.4	Importe de la posición (diferencia entre compras y ventas)
importe	numérico 11.4	Importe del asiento contable
límite	numérico 11.0	Importe límite de autorización de operaciones para autorizador
marca_activo	carácter 1	Clave para conocer si esta activa la tasa de interés S-si, N-no
mes	entero 2	Número del mes festivo
miscelanea	carácter 40	Campo para comentarios importantes sobre los clientes
monto_conta	numérico 11.4	Monto en moneda nacional para contabilización de operaciones
monto_invertido	numérico 11.4	Monto invertido
monto_otorga	numérico 11.4	Monto que se otorga por la divisa
monto_recibe	numérico 11.4	Monto que se recibe por la divisa
nombre_aut	carácter 40	Nombre del autorizador de operaciones importantes
nombre_banco	carácter 40	Nombre del banco
nombre_casa	carácter 40	Nombre de la casa de cambio de divisas
nombre_contacto	carácter 40	Nombre de la persona (contacto) en oficinas del cliente
nombre_cte	carácter 40	Nombre del cliente
nombre_cheq	carácter 40	Descripción de la cuenta de cheques
nombre_divisa	caracter 40	Descripción de la divisa
nombre_division	caracter 40	Nombre de la división operativa
nombre_especie	caracter 40	Descripción de la especie
nombre_formula	carácter 40	Descripción de los conceptos de los asientos contables
nombre_puesto	carácter 40	Descripción del puesto del usuario
nombre_suc	caracter 40	Nombre de la sucursal u oficina de servicios
nombre_usu	carácter 40	Nombre del usuario
num_autorizac	entero 6	Número de la autorización asignada a la operación realizada
num_autorizador	entero 3	Número para personas que pueden autorizar operaciones
num_banco	entero 4	Número para identificar a cada banco
num_cliente	entero 4	Número asignado a los clientes importantes y recurrentes
num_cuenta	entero 4	Numeros asignado por CNB para manejo de cuentas contables
num_chequera	entero 4	Número para identificar las chequeras utilizadas
num_deal	entero 7	Número de forma de operación y de control
num_divisa	entero 3	Número que identifica la divisa
num_especie	entero 2	Número para el tipo en que se realiza la operación 1-efectivo, 2-transferencia, 3-orden de pago, 4-cheque de viajero, 5-giro, 6-remesa
num_formula	entero 3	Número asignado a un grupo de asientos contables
num_puesto	entero 2	Número para los diversos puestos utilizados en la empresa
num_s1	entero 2	Identificador de la primera subcuenta de mayor
num_s2	entero 2	Identificador de la segunda subcuenta de mayor

CAMPO	DEFINICIÓN	DESCRIPCIÓN
num_s3	entero 2	Identificador de la tercera subcuenta de mayor
password	carácter 8	Clave secreta
pos_l_c	carácter 1	Clave para identificar la posición: C-corta, L-larga
pos_l_c_ini	carácter 1	Clave para identificar la posición de apertura: C-corta,L-larga
privilegios	carácter 1	Identificador especial para uso de aplicaciones restringidas
punto_reorden	numérico 9.0	Punto en cual se solicita una dotación de determinada divisa
referencia	entero 7	Identificador del deal utilizado en los registros contables
rendimiento	numérico 11.4	Importe obtenido en la inversión realizada
rfc	carácter 13	Registro federal de causantes del cliente
saldo	numérico 11.4	Saldo de la chequera
situacion_rango	carácter 1	Clave para el estado del rango: D-distribuido, R-generado
stock_ideal	numérico 9.0	Cantidad ideal de divisas para menudeo
stock_real	numérico 9.0	Cantidad real de divisas para menudeo
sucursal	entero 3	Número asignado al sucursal u oficina de servicio
t_prom_compra	numerico 7.6	Tipo promedio de compra actual
t_prom_ini	numerico 7.6	Tipo promedio de inicial
t_prom_venta	numerico 7.6	Tipo promedio de venta actual
tasa_interes	numérico 3.2	Tasa de interes anualizada de forma simple
telefono	entero 9	Teléfono del cliente
tipo_cambio	numérico 7.6	Tipo de cambio
tipo_divisa	caracter 1	Clasificación de las divisas: D-dólar, M-metales, O-monedas
tipo_operac	carácter 1	Clasificación de las operaciones: C-compra, V-venta
tipo_prom	numérico 7.6	Tipo de cambio promedio actual
total_compra	numérico 13.4	Totales de compra en la divisa en cuestión
total_venta	numérico 13.4	Totales de venta en la divisa en cuestión
usuario	carácter 8	Clave del usuario

4.1.2.- Estructuras de datos

En una estructura de datos vamos a representar las relaciones lógicas existentes entre los elementos individuales de datos, los campos. Como se ha mencionado, es vital un buen diseño de las estructuras porque invariablemente afectarán al diseño procedimental.

Las estructuras de datos nos determinan la organización, los métodos de acceso, el grado de asociatividad y las alternativas de procesamiento para la información.

Para un eficiente diseño de la estructura de datos se deben tomar en cuenta las operaciones que se efectuarán sobre dicha estructura. De aquí la importancia de apoyarnos en los productos del análisis (los diagramas de flujo de datos, de requisitos y de entidad-relación).

A continuación podemos observar las estructuras de datos diseñadas para la Casa de Cambios en esta etapa de diseño:

	T1 USUARIOS
k	usuario
	division
	sucursal
	password
	nombre_usu
	especialidad
	num_puesto
	privilegios

	T2 DEALS
k	division
k	sucursal
k	num_deal
	estado
	fecha_edo
	usuario

	T3 RANGOS_DEALS
k	division
k	sucursal
k	situacion_rango
k	desde
	hasta
	usuario
	fecha_gen
	fecha_dist

T4 SUCURSALES

k	division
k	sucursal
	nombre_suc

T5 DIVISAS

k	num_divisa
	nombre_divisa
	tipo_divisa
	cotizacion_compra
	cotizacion_venta
	fecha_operación

T6 BANCOS

k	num_banco
	nombre_banco

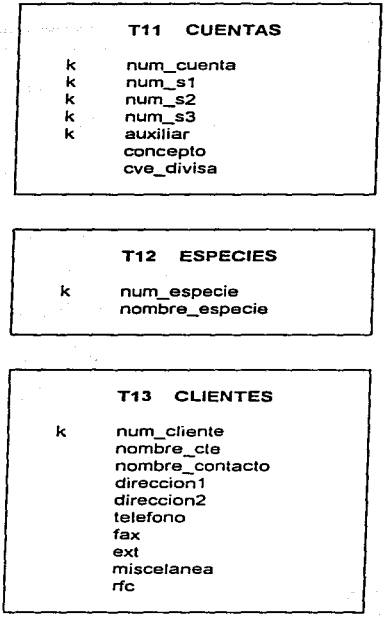
T7 CHEQUERAS

k	num_banco
k	num_chequera
	nombre_cheq
	num_divisa
	saldo
	control_inv
	fecha_ult

	T8 OPERACIONES
k	division
k	sucursal
k	num_deal
k	tipo_operac
	divisa_recibe
	divisa_otorga
	tipo_cambio
	monto_recibe
	monto_otorga
	monto_conta
	fecha_registro
	fecha_valor
	fecha_cheque
	fecha_conta
	num_especie
	num_cliente
	usuario
	num_banco
	num_cheque
	num_chequera
	num_autorizac
	num_formula
	cve_futuro
	nombre_cte
	num_puesto

	T9 POSICION
k	num_divisa
	tipo_prom
	equivalente_mn
	imp_posicion
	pos_l_c
	t_prom_compra
	t_prom_venta
	total_compra
	total_venta
	equiv_compra
	equiv_venta
	imp_pos_ini
	t_prom_ini
	pos_l_c_ini
	fecha_ult

	T10 MIG_CONTA
	num_cuenta
	num_s1
	num_s2
	num_s3
	auxiliar
	concepto
	importe
	fecha_conta
	referencia
	division
	sucursal
	estatus

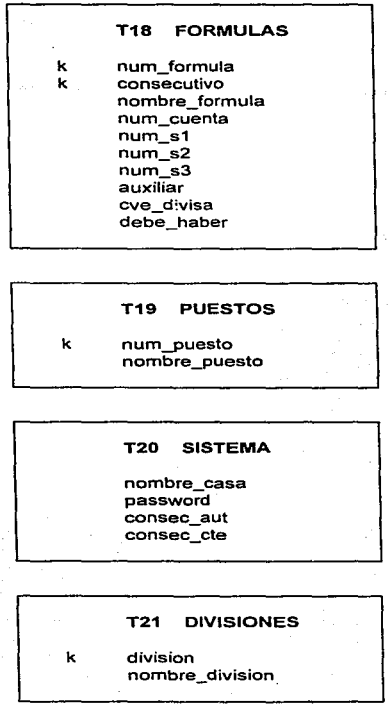

T11 CUENTAS
k num_cuenta
k num_s1
k num_s2
k num_s3
k auxiliar
concepto
cve_divisa
T12 ESPECIES
k num_especie
nombre_especie
T13 CLIENTES
k num_cliente
nombre_cte
nombre_contacto
direccion1
direccion2
telefono
fax
ext
miscelanea
rfc

	T14 AUTORIZACIONES
k	num_autorizac
	division
	sucursal
	num_deal
	tipo_operac
	fecha_operacion
	fecha_registro
	cve_autorizador

	T15 AUTORIZADORES
k	num_autorizador
	nombre_aut
	limite

T16 FUTUROS	
k	division
k	sucursal
k	num_deal
k	tipo_operac
	divisa_recibe
	divisa_otorga
	tipo_cambio
	monto_recibe
	monto_otorga
	monto_conta
	fecha_reg
	fecha_valor
	fecha_conta
	num_especie
	num_cliente
	usuario
	num_banco
	num_chequera
	num_autorizac
	num_formula
	cve_futuro
	nombre_cte
	num_puesto
	nombre_cte
	cve_baja

T17 DIAS_FESTIVOS	
k	mes
k	dia
	descripcion


T18 FORMULAS
k num_formula
k consecutivo
nombre_formula
num_cuenta
num_s1
num_s2
num_s3
auxiliar
cve_divisa
debe_haber
T19 PUESTOS
k num_puesto
nombre_puesto
T20 SISTEMA
nombre_casa
password
consec_aut
consec_cte
T21 DIVISIONES
k division
nombre_division

T22 INVERSIONES

k	num_banco
k	num_chequera
	tasa_interes
	marca_activo

T23 STOCK_DIVISA

k	division
k	sucursal
k	num_divisa
	stock_ideal
	stock_real
	punto_reorden
	fecha_ult

T24 DET_INVERSION

k	num_banco
k	num_chequera
k	control_inv
	rendimiento
	monto_invertido
	tasa_interes
	fecha_registro
	dias_invertido

4.2.- Diseño arquitectónico.

El objetivo principal del diseño arquitectónico es desarrollar una estructura de sistema modular y representar las relaciones de control entre los módulos. Además, el diseño arquitectónico mezcla la estructura de programas y la estructura de datos y define las interfases que facilitan el flujo de los datos a lo largo del programa.

La explotación de los diagramas de flujo de datos elaborados en la etapa de diseño nos llevarán a desarrollar la estructura de todo el sistema de una forma lógica. De tal manera que llegando al máximo detalle de los diagramas podemos ir obteniendo las opciones de menúes que conformarán la interfase con los usuarios finales.

Revisando los diagramas de flujo de datos desarrollados para la casa de cambio de divisas obtuvimos el diseño arquitectónico que se muestra a continuación.

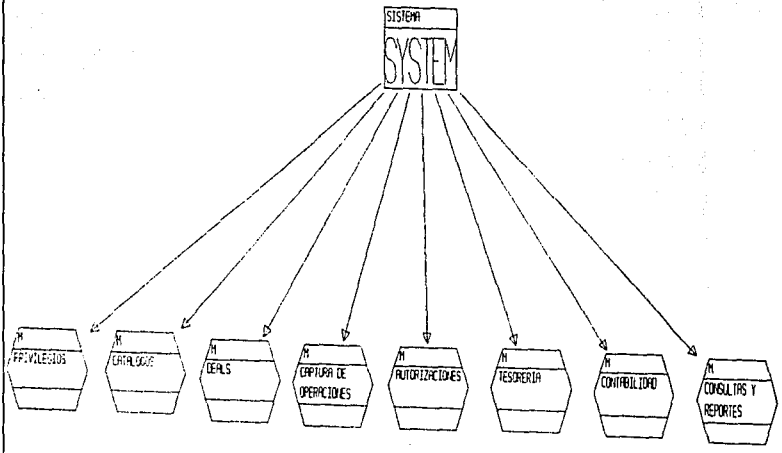


FIGURA 4.1 Estructura del sistema (General)

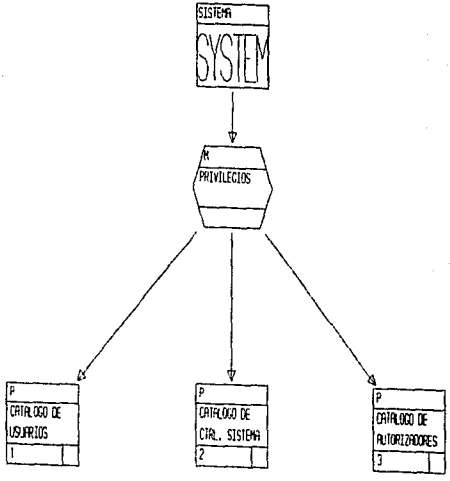


FIGURA 4.2 Estructura del sistema (Privilegios)

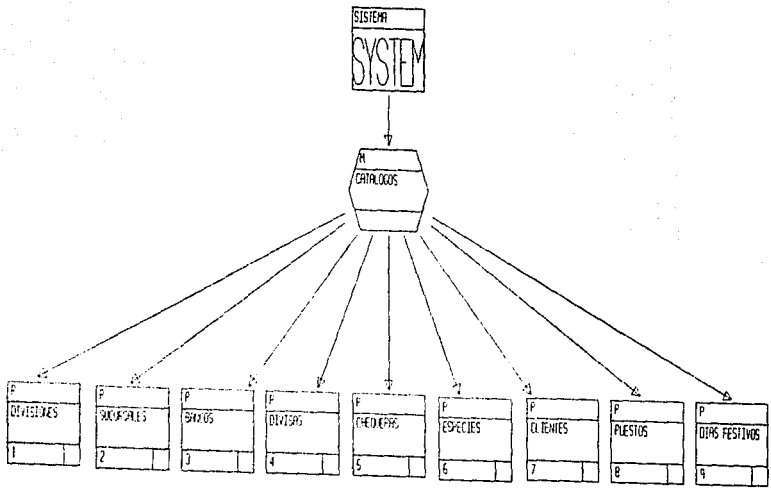


FIGURA 4.7 Estructura del sistema (Catalogos)

FIGURA 4.4 Estructura del sistema (Deals)

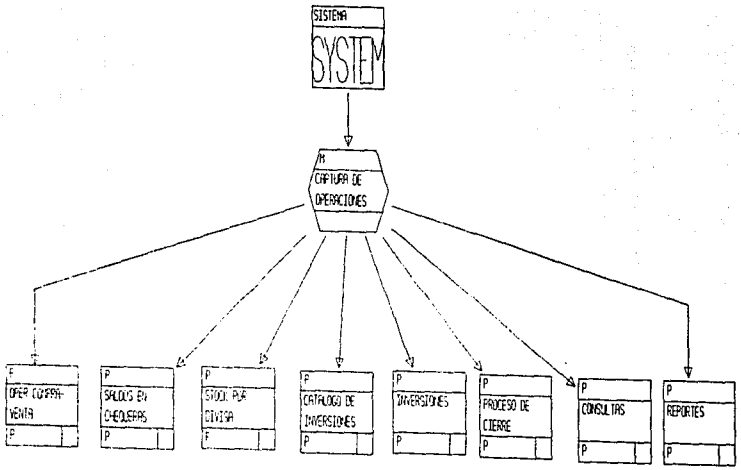


FIGURA 4.5 Estructura del sistema (Operaciones)

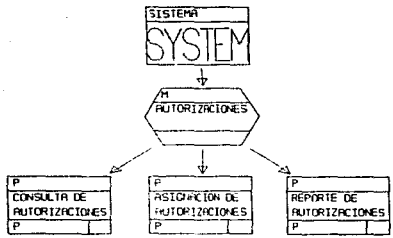


FIGURA 4.6 Estructura del sistema (Autorizaciones)

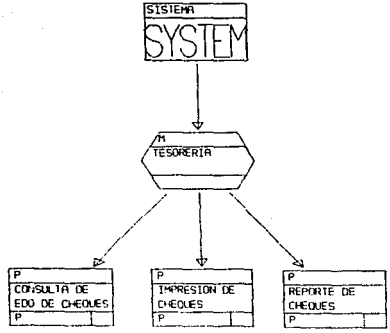


FIGURA 4.7 Estructura del sistema (Tesorería)

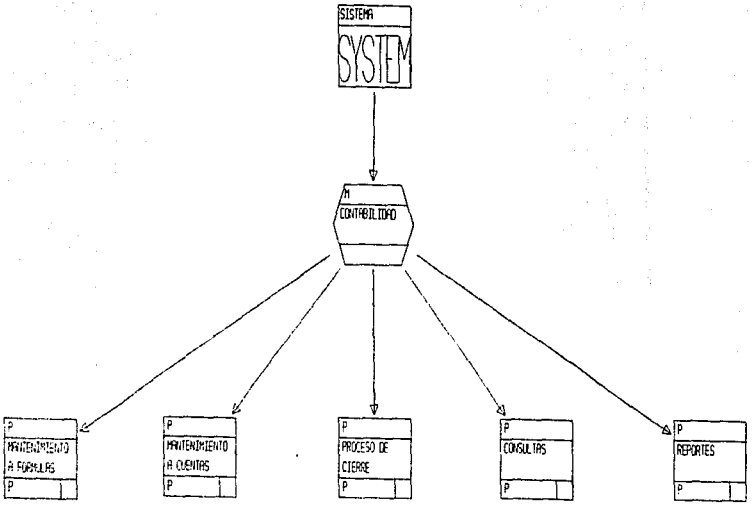


FIGURA 4.8 Estructura del sistema (Contabilidad)

FIGURA 4.9 Estructura del sistema (Consultas y Reportes)

4.3.- Diseño procedimental.

El diseño procedimental se realiza después de que se ha establecido la estructura del sistema y de los datos. La especificación procedimental que define los detalles algoritmicos debería explicarse en un lenguaje natural.

El diseño procedimental debe especificar los detalles de los procedimientos sin ambigüedades.

Usando un lenguaje natural podríamos describir un conjunto de pasos procedimentales de demasiadas formas diferentes. Frecuentemente, el significado concreto depende del contexto. A menudo redactamos como si fuera posible establecer un diálogo con el lector. Por estas y muchas otras razones, debemos usar una forma más restringida para la representación de los detalles procedimentales.

Las técnicas más comunes para definir el diseño procedimental son el diagrama de flujo y el pseudocódigo.

El diagrama de flujo ha perdido popularidad debido al surgimiento de herramientas de programación de cuarta generación con características no procedurales en donde no existe una secuencia lógica como lo sugiere el diagrama de flujo. Por esta razón el pseudocódigo ha cobrado mayor importancia y se ha convertido en la técnica más poderosa en la definición del diseño procedimental.

El pseudocódigo en los lenguajes de cuarta generación sólo contiene las macroactividades que se realizan de tal manera que el desarrollador pueda entender que es lo que se requiere que realice el programa.

Es importante mencionar que el pseudocódigo no especifica los campos, estructuras, ni métodos de acceso a utilizar, ya que la herramienta de cuarta generación obtiene esta información del diccionario y las estructuras de la base de datos.

CAPÍTULO V

IMPLEMENTACIÓN DEL SISTEMA

5.1.- Conceptos Generales.

Con el objeto de lograr un entendimiento completo de la etapa de instrumentación dentro del desarrollo de un sistema de información, es indispensable conocer una serie de conceptos tales como: Base de datos, Sistema de manejo de bases de datos (DBMS), tipos de DBMS, el enfoque relacional, mismos que serán tratados en el presente capítulo.

5.1.1.- Base de datos.

Se entiende por base de datos un conjunto ordenado de archivos de información consolidada, estructurada y normalizada (no redundante), que permite el acceso y actualización de la información en forma simultánea, sincronizada y consistente a uno o varios procesos de usuario.

Una base de datos debe cumplir con una serie de requisitos tales como:

- Contener los datos sin considerar dónde o cómo se almacenan.
- Contener los datos en su forma más simple (no elaborado).
- Ser fácilmente expandible.
- Ser accesible.
- No contener redundancias.
- Ser independiente de las aplicaciones.
- Integridad en los datos, seguridad y contar con mecanismos de recuperación.

5.1.2.- Sistema de manejo de base de datos.

Se define como DBMS al conjunto de rutinas, funciones, métodos de acceso, áreas de almacenamiento, de trabajo y de control requeridas para el tratamiento de la información bajo el concepto de Base de datos.

Dentro de las características más importantes de un DBMS, podemos mencionar las siguientes:

- Controla centralmente al recurso corporativo: DATOS.
- Distribuye los datos en forma paralela.
- Separa "Formato de datos" de "Uso de datos".
- Fundamenta la nueva generación de lenguajes.
- Maneja los datos más técnicamente.

Los sistemas de manejo de base de datos proporcionan una gran cantidad de ventajas, a continuación mencionaremos las más significativas:

- Permiten controlar, centralizar y consolidar la información.
- Facilitan el uso compartido de la información por múltiples usuarios en tiempo real.
- Proporcionan seguridad en la protección y recuperación de la información.
- Aumentan la facilidad de uso de los sistemas de procesamiento de información.
- Reducen los requerimientos de procesos de mantenimiento a la información.
- Reducen el tiempo requerido para el desarrollo de aplicaciones.
- En general, incrementan la eficiencia en el uso de un procesador desde el punto de vista de los usuarios.

FIGURA 5.1 ESQUEMA DE BASE DE DATOS

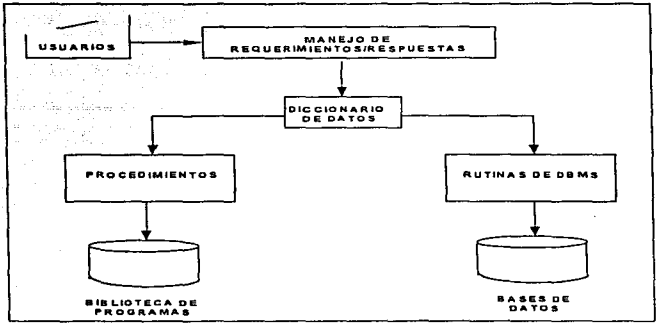
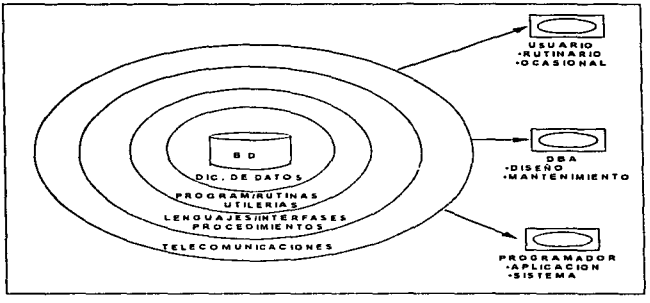


FIGURA 5.2 DBMS



5.1.3.- El enfoque relacional.

En el sistema DBMS relacional, una estructura lógica se representa por medio de tablas bidimensionales llamadas relaciones. Una entidad se representa por un renglón en la tabla.

La ventaja principal del enfoque relacional está en la simplicidad de su representación en la estructura lógica de la base de datos y en la flexibilidad para establecer relaciones de datos por medio de campos de conexión. Todas las entidades en una base de datos relacional están representadas como tablas separadas y no están colocadas en ninguna jerarquía fija como es el caso de los árboles. La mayoría de los usuarios finales probablemente entiendan las tablas mucho mejor de los que entenderían una estructura compleja de red, por lo tanto, el enfoque relacional puede facilitar la comunicación entre el DBA y los usuarios finales.

El enfoque relacional hace posible el alcanzar mayor independencia de los datos usando campos de conexión en vez de señaladores para enlazar registros relacionados en diferentes archivos (o relaciones). El sistema relacional establece tal asociación insertando el valor clave - principal del registro padre en un archivo en el campo de conexión correspondiente de todos sus registros hijos.

Una característica única del sistema relacional es su independencia de trayectorias de Entrada / Salida. Ya que una base de datos relacional consta de una colección de tablas separadas, cualquier tabla (o relación) se puede accesar directamente sin necesidad de accesar otras relaciones basadas en una estructura de datos fija, tale como un árbol o una red.

El enfoque relacional sirve también como un mecanismo flexible de visión o escenario, para que los usuarios individuales puedan seleccionar datos. Un escenario puede incluir datos de cualquier columna (campos) y renglones (registros).

5.1.4.- Arquitectura de un sistema relacional.

La arquitectura relacional se puede expresar en términos de tres niveles. A continuación se explican los componentes de la arquitectura que se observan en la figura 5.3:

1) Modelo relacional de datos:

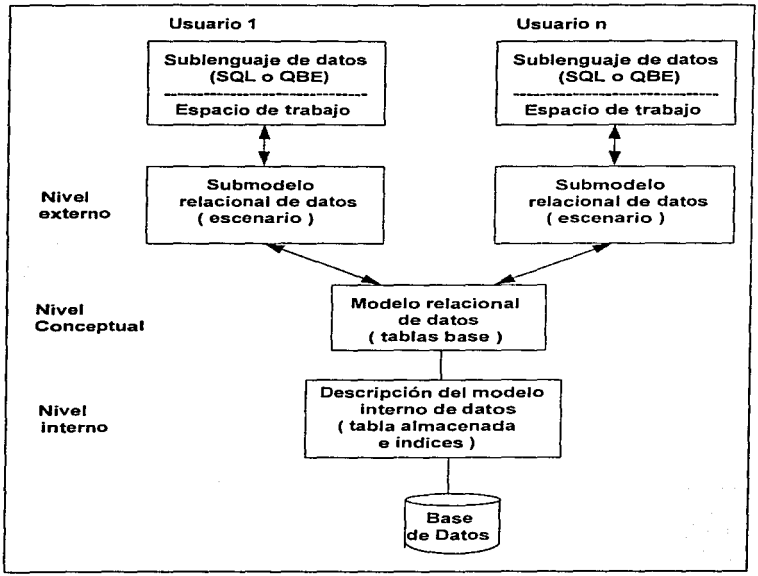
En el nivel conceptual, el modelo relacional de datos está representado por una colección de relaciones almacenadas (también llamadas tablas base). Cada registro de tipo conceptual en un modelo relacional de datos se implanta como un archivo almacenado distinto.

2) Submodelo de datos:

Los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos; cada uno consta de uno o más escenarios (views) para describir los datos requeridos por una aplicación dada. Un escenario puede incluir datos de una o más tablas de datos.

Cada programa de aplicación está provisto de un buffer (llamado "área de trabajo del usuario" o user work area, UWA), donde el DBMS puede depositar los datos recuperados de la base para su procesamiento, o puede guardar temporalmente sus salidas antes de que el DBMS las escriba en la base de datos.

Figura 5.3 Arquitectura de un sistema relacional.



3) Esquema de almacenamiento:

En el nivel interno, cada tabla base se implanta como un archivo almacenado. Para las recuperaciones sobre las claves principal o secundaria se pueden establecer uno o más índices para accesar un archivo almacenado.

4) Sublenguaje de datos:

Es un lenguaje de manejo de datos para el sistema relacional. Originalmente, Codd propuso dos sublenguajes de datos: i) álgebra relacional, ii) cálculo relacional. Codd ha demostrado que ambos lenguajes son "relacionalmente completos", esto es, cualquier relación que pueda derivarse de una o más tablas de datos también se puede derivar con un solo comando del sublenguaje. Por tanto, el modo de operación de Entrada / Salida en un sistema relacional se puede procesar en la forma una tabla a la vez en lugar del procesamiento tradicional de un registro a la vez. En otras palabras, se puede recuperar una tabla en vez de un solo registro con la ejecución de un comando del sublenguaje de datos.

El álgebra relacional es un lenguaje de procedimientos de alto nivel que permite el uso de SELECT, JOIN, PROJECT y otros operadores para derivar la tabla deseada en diversos pasos desde las tablas base originales en el modelo relacional, o también otras tablas derivadas intermedias. Por otro lado , el cálculo relacional es un potente lenguaje de consulta no algorítmico, que permite al usuario la recuperación de datos mediante el establecimiento de condiciones de consulta sin necesidad de codificar en forma detallada los pasos de la recuperación.

Aunque el álgebra relacional no es tan accesible para el usuario como lo es el cálculo relacional, proporciona los conceptos básicos para el desarrollo de lenguajes relacionales de manejo de datos. Los dos principales sublenguajes relacionales SQL (STRUCTURED QUERY LANGUAGE) y QBE (QUERY BY EXAMPLE) , están implementados con el mismo espíritu que el cálculo relacional. Sin embargo, la mayoría de los comandos SQL tienen equivalentes algebraicos. En otras palabras, un comando SQL se puede traducir a una operación relacional algebraica equivalente.

Una de las características importantes de SQL es que el lenguaje se puede usar no sólo como DML para manejar los datos de la base, sino también como lenguaje de definición de datos (data definition language, DDL) para definir tablas base, escenarios o incluso archivos almacenados en los niveles conceptual, externo e interno, respectivamente. Más aún, el SQL es un lenguaje flexible cuyos comandos se pueden emitir ya sea interactivamente desde una terminal o incluirse en el lenguaje del programa principal. Con estas atractivas características, el SQL se ha adaptado en varios sistemas relacionales.

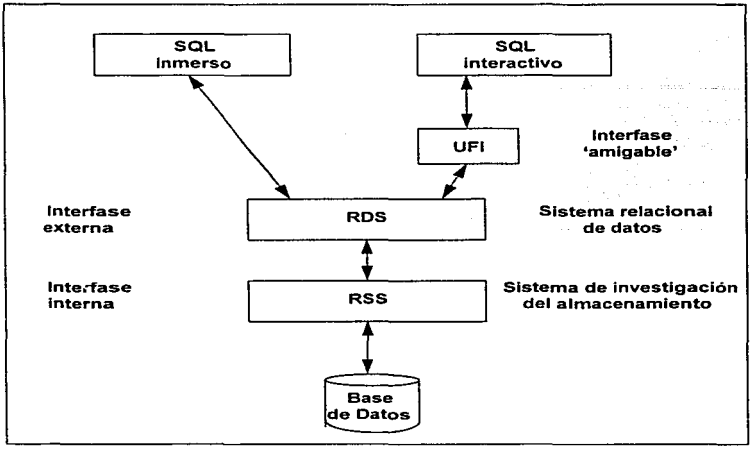
Los sistemas relacionales constan de dos subsistemas principalmente:

i) sistema relacional de datos (relational data system, RDS)

ii) sistema de investigación de almacenamiento (research storage system, RSS).

Como se observa en la figura 5.4 el RDS es una interfase externa similar al sistema de control de la base de datos. Sus funciones principales son el análisis sintáctico y la optimización. Cuando el sistema recibe una proposición SQL, se le analiza gramaticalmente con base en la gramática del lenguaje para determinar si pertenece al lenguaje y cuál es su estructura. El optimizador de RDS escogerá una trayectoria eficiente de acceso y el comando SQL se traducirá a los comandos CALL adecuados para llamar rutinas de control de ejecución del RDS.

Figura 5.4 Componentes



La interfase amigable con el usuario (user friendly interface, UFI) es una interfase con una terminal para recibir comandos interactivos SQL desde la terminal. Sin embargo, los comandos SQL dentro del lenguaje de programación del sistema central se analizan directamente mediante precompilador del RDS para su análisis gramatical. Aquí, el precompilador funciona como el compilador del lenguaje SQL. Antes de que el compilador COBOL traduzca el lenguaje del sistema central, digamos COBOL, el programa pasa a través del precompilador para que reemplace las proposiciones SQL incluidas con proposiciones CALL para controlar las rutinas del RDS.

El sistema de investigación de almacenamiento (RSS) es una interfase interna que maneja los datos almacenados, índices, buffers, distribución del almacenamiento, etc. Sin embargo, el RSS confía en los métodos de acceso subyacentes del sistema operativo para el manejo de archivos de nivel bajo.

5.2.- Creación de la Base de Datos para la Casa de Cambios.

Es muy importante mencionar en este punto que al momento de surgir el requerimiento de un sistema que permitiera el registro y control de la información generada por la operación de una casa de cambio de divisas, tema del presente trabajo, se tomó la decisión en conformidad con el personal responsable de la compañía cambiaria de realizar la implementación del sistema en un lenguaje de cuarta generación (POWER BUILDER) y en un ambiente de base de datos relacional (SYBASE).

Sin embargo, debido a problemas ajenos a nosotros de carácter económico y de política interna de la compañía, nos hemos visto en la necesidad de hacer algunos cambios en la implementación; si bien se respetarán los conceptos básicos propuestos en relación a la utilización de un lenguaje de cuarta generación y una base de datos relacional, el lenguaje en que desarrollaremos las aplicaciones será "CA-GENERATOR" que es una herramienta de cuarta generación para desarrollo de software y un manejador de base de datos relacional "CA-DB VAX", ambos productos comercializados por "COMPUTER ASSOCIATES INTERNATIONAL INC."

Cabe mencionar que el manejador de base de datos CA-DB VAX incluye como lenguaje relacional para el manejo y definición de datos a SQL. Algunas de las características más importantes de este DBMS se describen a continuación.

5.2.1.- Organización de la Información.

Toda la información de la base de datos se encuentra almacenada en archivos, dichos archivos o files son definidos en SQL con un nombre de file, un nombre físico que servirá para su localización dentro del sistema (nodo, disco, directorio, nombre) y el tipo de información que contendrá, ya sea datos, índices o ambos.

La base de datos se define como un gran archivo donde se almacenará toda la información para el DBMS (tablas, indices, views, relaciones y datos del sistema).

Para cada uno de los files donde se guardará toda la información del sistema se asigna un espacio de tamaño determinado denominado "tablespace". Se deben definir espacios para los files con el objeto de asociar los archivos físicos a los espacios para tablas (tablespaces) definidos. La información del DBMS se encuentra en un archivo llamado como la base de datos y asociado a un espacio de tabla denominado SYSTEM. Como se mencionó anteriormente, se pueden definir archivos de datos y de índices, para después asociarlos a un espacio de tabla y, finalmente, crear la tabla que a su vez será asociada al espacio correspondiente. De esta manera la información de la tabla se almacena en los archivos o files asociados a su espacio de tabla definido (figura # 5). Es importante mencionar que en caso de que se agote el espacio de los archivos que se encuentran asociados al espacio de la tabla, puede crearse uno nuevo y asociarse a dicho espacio resolviendo rápidamente el problema.

Para la creación de los archivos de la base es conveniente utilizar nombres lógicos dentro del sistema operativo con el objeto de tener una independencia entre los archivos de la base y su localización física; esta técnica permitirá balancear las cargas de accesos a los diferentes dispositivos de almacenamiento de la información.

5.2.2.- Métodos de acceso a la información.

El manejador de base de datos "CA-DB VAX" maneja dos tipos de índices para el mejor acceso a la información contenida en las tablas: índices B-tree e índices hash.

Una tabla puede tener un número ilimitado de índices B-tree, cada uno de los cuales puede crearse sobre un sólo campo o bien sobre la unión de varios, no necesariamente contiguos en la tabla. Cada índice puede definirse como único y por lo tanto no permitir llaves duplicadas o como no único. La información de los índices (árboles) de una tabla se almacena en los archivos (files) que están asociados al espacio de tabla (tablespace) en el que se encuentra ésta y que son del tipo de índices.

El índice de tipo hash es más eficiente en búsqueda directa ya que utiliza un algoritmo o fórmula que, aplicada a la llave del registro, define una localización física exacta por lo que el acceso sólo lleva una lectura a la información. Las desventajas del índice hash es que sólo se utiliza cuando la llave se proporciona completa, la administración del espacio de la tabla es más compleja que con índices B-tree y no se utiliza para ordenamientos de la información.

Para una misma tabla es común definir archivos de datos y archivos de índices, logrando con esto balancear los accesos a los dispositivos de almacenamiento de la información ya que los datos se pueden colocar en un dispositivo diferente al de los índices.

5.2.3.- Secciones de programas precompilados.

Al momento de precompilar un programa que contiene inserciones de instrucciones de SQL, el manejador de base de datos define la ruta más eficiente de acceso a la información para el programa en cuestión y la almacena en el espacio del sistema en secciones independientes para cada aplicación. De esta manera, al ejecutarse el programa, el DBMS conoce ya el método de acceso óptimo a los datos.

El DBMS intenta accesar la información siempre através de un índice. El método de acceso secuencial es utilizado cuando no hay otra opción ya que puede resultar muy lento y en tablas de tamaño considerable afecta fuertemente el performance del sistema.

5.2.4.- Tablas del sistema.

La información de todos los elementos de la base de datos es almacenada también en tablas; de esta forma, existe una tabla que contiene la información de cada tabla de la base de datos, existe otra que guarda la información de los archivos de la base, otra contiene los datos de los índices, etc. El acceso a estas tablas es como cualquier tabla de la base, únicamente que por razones de integridad su actualización está restringida.

La tabla 5.1 muestra los nombre asignados a cada una de las tablas definidas en el diseño de la base de datos; estos nombres deben de cumplir con dos requisitos: no exceder 18 caracteres y no contener espacios.

El proceso de creación de la base de datos puede resumirse en las siguientes actividades.

1.- Definición del nombre lógico de la base de datos.

2.- Creación de la base de datos.

3.- Para cada una de las tablas:

3.1.- Definición de los nombres lógicos de sus files.
3.2.- Creación del espacio para tabla (tablespace).
3.3.- Creación de los archivos de la tabla.
3.4.- Asociar los archivos al espacio para tabla.
3.5.- Creación de la tabla asociándola al espacio correspondiente.
3.6.- Creación de los índices de la tabla.

El anexo 1 muestra a detalle la creación de la base de datos para la Casa de Cambio.

tabla 5.1 Definición de nombres de tablas

NOMBRE	DESCRIPCIÓN
USUARIOS	Usuarios del sistema
DEALS	Formas de operación
RANGOS_DEALS	Rangos de formas de operación
SUCURSALES	Sucursales de la compañía
DIVISAS	Divisas de cambio
BANCOS	Sucursales bancarias
CHEQUERAS	Chequeras de la compañía
OPERACIONES	Registro de las operaciones de la compañía
POSICION	Posición de las divisas de la casa de cambio
MIG_CONTA	Contabilización de la operaciones
CUENTAS	Catálogo de cuentas
ESPECIES	Tipos de especies
CLIENTES	Clientes de la empresa
AUTORIZACIONES	Autorización de operaciones
AUTORIZADORES	Autorizadores de operaciones
FUTUROS	Operaciones futuras
DIAS_FESTIVOS	Días festivos
FORMULAS	Grupos de movimientos contables
PUESTOS	Puestos el personal de la casa de cambio
SISTEMA	Control del sistema
DIVISIONES	Divisiones de la compañía
INVERSIONES	Inversiones de capital
STOCK_DIVISA	Stocks por divisa
DET_INVERSION	Detalle de inversiones

5.3.- Programación del sistema.

Todas las etapas de la ingeniería de software que preceden a la instrumentación o codificación están orientadas hacia un objetivo final: traducir las representaciones del software a una forma que pueda ser comprendida por la computadora. La etapa de instrumentación puede ser definida como un proceso que transforma el diseño en un lenguaje de programación.

Resulta de vital importancia la correcta elección del lenguaje de programación, ya que las características propias del lenguaje elegido, así como el estilo de programación pueden afectar profundamente a la calidad y posteriormente al mantenimiento del software.

Entre los criterios que se aplican durante la evaluación de los lenguajes disponibles están:

1.- Área de aplicación general.
2.- Complejidad algorítmica y computacional.
3.- Entorno en donde se ejecutará el software.
4.- Consideraciones de rendimiento.
5.- Complejidad de las estructuras de datos.
6.- Conocimiento del personal de desarrollo de software.
7.- Disponibilidad de un compilador.

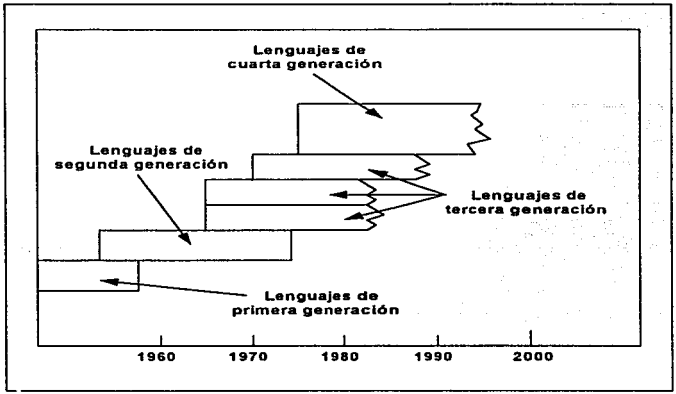
No obstante todos estos criterios son básicos al momento de elegir un lenguaje de programación, el que se refiere al área de aplicación es sin duda el que reviste mayor importancia durante el proceso de selección.

5.3.1.- Lenguajes de programación.

Como se mencionó anteriormente, la etapa de implementación o codificación traduce una representación del software, dada por un diseño detallado, a una realización en un lenguaje de programación. El proceso de traducción continúa cuando un compilador acepta el código fuente como entrada y produce como salida un código objeto dependiente de la máquina que a su vez traducido a un código máquina constituido por las instrucciones reales que dirigen la lógica microprogramada de la unidad central de proceso.

Con el objeto de entender el desarrollo de los lenguajes de programación a lo largo de su historia, los expertos hacen una división de dicho desarrollo en cuatro generaciones; a continuación explicaremos brevemente cada una de éstas:

Figura 5.5 Generaciones de los lenguajes.



Primera Generación: La primera generación se refiere a aquellos lenguajes cuya codificación era a nivel de máquina. No obstante esta generación de lenguajes terminó antes de la época de los sesentas actualmente siguen siendo de mucha utilidad. El código máquina y el lenguaje ensamblador representan la primera generación; estos lenguajes dependientes de la máquina muestran el menor nivel de abstracción con el que se puede representar un programa.

Existen tantos lenguajes ensambladores como arquitecturas de procesadores con sus correspondientes conjuntos de instrucciones. Es importante mencionar que estos lenguajes es conveniente usarlos solo cuando un lenguaje de alto nivel no satisfaga los requisitos o no esté disponible.

Segunda Generación: La segunda generación de lenguajes fue desarrollada a finales de los años 50 y principios de los 60 y ha servido como base para todos los lenguajes de programación modernos (tercera generación). La segunda generación de lenguajes está caracterizada por su amplio uso, la enorme cantidad de bibliotecas de software y la gran familiaridad y aceptación. Dentro de esta generación podemos mencionar a lenguajes como FORTRAN, COBOL, ALGOL y BASIC.

Tercera Generación: Los lenguajes de tercera generación (también llamados lenguajes de programación moderna o estructurada) se caracterizan por sus potentes posibilidades procedimentales y de estructuración de datos. Los lenguajes de esta clase se pueden dividir en tres categorías: lenguajes de alto nivel de propósito general (PASCAL, Modula 2 y C), lenguajes de alto nivel orientados a objetos (C++, Smalltalk y Eiffel) y lenguajes especializados (LISP, PROLOG y FORTH).

5.3.2.- Lenguajes de cuarta generación.

El objetivo en el desarrollo de software ha sido el de generar programas de computadora con cada vez mayores niveles de abstracción. Los lenguajes de la primera generación trabajan a nivel de instrucciones máquina, el menor nivel de abstracción posible. Los lenguajes de segunda y tercera generación han subido el nivel de representación de los programas de computadora, pero aún hay que especificar distintos procedimientos algoritmicos perfectamente detallados. Los lenguajes de cuarta generación (L4G) han venido a elevar el nivel de abstracción.

Los L4G contienen una sintaxis distinta para la representación del control y para la representación de las estructuras de datos. Por sus características se han dividido en tres grandes categorías:

Lenguajes de petición: hasta ahora, la gran mayoría de los L4G se han desarrollado para ser usados conjuntamente con aplicaciones de bases de datos. Tales lenguajes de petición permiten al usuario manipular de forma sofisticada la información contenida en la base de datos previamente creada.

Generadores de Programas: más que basarse en una base de datos predefinida, un generador de programas permite al usuario crear programas en un lenguaje de tercera generación usando notablemente menos sentencias. Estos lenguajes de programación de muy alto nivel hacen un gran uso de la abstracción de datos y de procedimientos. Desafortunadamente para los que trabajan en el campo de sistemas y productos de ingeniería, la mayoría de los generadores de programas se centran exclusivamente en aplicaciones de sistemas de información de negocios y generan programas en COBOL.

Otros lenguajes de cuarta generación: aunque los lenguajes de petición y los generadores de programas son los L4G más comunes, existen otras categorías. Los lenguajes de soporte a la toma de decisiones permiten a los usuarios llevar a cabo una gran variedad de análisis "que pasa si", que van desde los simples modelos de hojas de cálculo bidimensionales hasta los sofisticados sistemas de modelos estadísticos y de investigación de operaciones. Los lenguajes de prototipos se han desarrollado para apoyar el uso de prototipos facilitando la creación de interfases de usuario y de diálogos, además de proporcionar medios para la modelización de datos.

5. 4.- Herramientas de cuarta generación para la casa de cambio.

El ambiente de cuarta generación que se esta empleando, esta compuesto por las siguientes herramientas:

5.4.1.- Herramienta de definición de datos (Data definition tool):

En este módulo se definen las unidades individuales de datos de las aplicaciones, los campos. Estos son los bloques de construcción con los cuales se crearán otras entidades.

Se pueden definir cualquiera de los siguientes tipos de campo:

- Carácter
- Entero
- Numérico
- Fecha
- Tiempo

Una vez que se han creado los campos y almacenado en el diccionario de aplicaciones de la base de datos, se pueden utilizar en diferentes entidades tales como tablas o archivos, views, call parameters, runtime parameters, formas o pantallas, y reportes.

En esta parte se pueden especificar otras características para cada campo en particular. Así podemos decir si el campo acepta sólo mayúsculas o minúsculas, si tiene una etiqueta y en que posición aparecerá en una pantalla o en un reporte, si el campo es signado, de que longitud, si tiene decimales, si lleva el símbolo $ por ser tipo importe (money), el formato de la fecha, si tiene mascarilla, si se requiere que el campo sea llenado parcial o totalmente, también se pueden asociar tres tipos de listas de valores: válidos, sugeridos e inválidos, así como textos de ayuda en donde se redacta el uso que se le da al campo en cuestión.

En otro de los módulos de esta herramienta se definen las tablas de la base de datos. Además de dar de alta los campos que componen la tabla, se especifica también el nombre de la tabla, el dueño, el nombre del tablespace (área de almacenamiento que puede contener más de una tabla), y el modo en que se compartirá la tabla cuando sea accesada por múltiples transacciones, por página (page), por renglón (row), por tabla (table) o por ninguna (donde la primera transacción que la accese la bloquea en forma exclusiva). La aplicación automáticamente determina el tamaño de la tabla a partir del tamaño de sus columnas. Otras características que se deben de seleccionar son la llave primaria y si acepta nulos o no. También se seleccionan los campos que forman los índices de las tablas y si aceptan llaves duplicadas o no, tipo único o de tipo hash en donde este tipo de índice realiza cálculos en los valores de la columna para determinar en que parte de la base de datos almacenar los datos de la tabla. Es válido copiar las especificaciones de tablas existentes.

De forma similar a como se dan de alta las estructuras de la base de datos, también podemos crear los views. Estos son como ventanas que ven a una o a varias tablas de la base de datos. Los view no contienen datos, pero accesa columnas selectas de datos de otras tablas y views, y despliega las columnas como si fueran de una sola tabla.

Para almacenar y accesar información también se pueden usar las tablas RMS (Record Management Services). Se pueden especificar en este tipo de tablas si son indexadas o si el acceso a los datos de la tabla será en el orden en que se hayan insertado, es decir, de forma secuencial. Dependiendo de la forma de acceso seleccionada, será el manejo y la creación del archivo. Todas las demás características se definen de la misma forma que en el otro tipo de tablas. Este tipo de tablas están fuera de la base de datos, sin embargo, se pueden accesar como si fuera prácticamente cualquier otra tabla de la base de datos.

En esta herramienta se crean los subprogramas externos que se planean para ser llamados desde programas generados, call modules. Un call module puede ser cualquier rutina que se haya escrito en cualquier lenguaje que se pueda llamar como cobol, basic, fortran, o C. Aquí se definen los campos con las características que el call module y la aplicación utilizan para pasarse datos entre ellos. Estos campos son llamados argumentos.

Los runtime parameters son campos usados para pasar valores del usuario a un programa batch (programa generado que no utiliza pantalla). Cuando se genera un programa batch, se puede especificar que un programa acepte uno o más valores del usuario cuando esta ejecutándose. De igual manera al pasar el control de un programa a otro enviando parámetros se requiere definir un runtime parameter.

5.4.2.- Herramienta para generación de formas (Form Painter tool).

Una forma es la interfase entre el usuario y la aplicación. El usuario introduce información en una aplicación através de una forma. Una vez que es definida una forma puede ser utilizada en varios programas.

Se pueden diseñar las pantallas o formas de dos maneras. Una de ellas es insertando manualmente los campos en el área de la forma y la otra es recurrir al Form Painter para que nos dibuje automáticamente los campos que están descritos de la tabla que estemos empleando. Las formas creadas automáticamente pueden dejarse así o pueden modificarse para cumplir con una necesidad específica.

Las formas pueden incluir:

- Campos- usados para desplegar los datos de la tabla que se esta accesando.
- Textos- incluye los encabezados y las etiquetas o títulos de los campo y explicaciones.
- Cajas- cajas que pueden contener nombres.
- Líneas- lineas horizontales y verticales que sirven para delinear las formas.
- Regiones tabulares- región en donde se repiten renglones de datos.

Tipicamente la región tabular se emplea cuando se hace el join de varias tablas que tienen relaciones de uno a muchos de tal manera que se pueden desplegar múltiples ocurrencias de un registro.

- Páginas- las formas pueden contener una, dos o tres páginas.

También se define el tamaño de la forma, si es de 21 ó de 24 lineas, si lleva un texto de ayuda en los campos o en la forma total, así como el poder hacer un test o prueba de la forma al terminar de diseñarla. La forma puede ser copiada de una existente. Se determina si la pantalla es borrada cada vez que se use, el tamaño de las regiones tabulares, los atributos de video, el tipo de cursor, si el

campo es protegido o requerido, si tiene máscara, si se despliegan o no los ceros a la izquierda, si se usa el separador de miles, el empleo de signos o no, los formatos de fechas y de tiempo, el orden en que el cursos pasa por cada campo,

5.4.3.- Herramientas para generación de reportes (Report Painter tool).

Esta herramienta nos sirve para diseñar reportes para los programas desarrollados.

De la misma manera que en el generador de formas, se pueden diseñar los reportes manualmente o utilizando el reporte sugerido automáticamente a partir de la tabla seleccionada.

Antes de agregar el reporte al programa, se puede validar desplegando el formato del reporte en la pantalla.

Se pueden crear dos tipo de reportes:

- Reportes tabulares: son aquellos con formato renglón y columna donde generalmente tienen encabezados en la parte superior de cada hoja seguidos por renglones de datos de las tablas seleccionadas. Este tipo de reportes consiste de líneas de detalle, encabezados de página, pies de página, grandes totales, encabezados y pies de página por grupos.
- Reportes de etiquetas: contienen bloques de información, como etiquetas de correo. Estos bloques de información se repiten por cada registro seleccionado conteniendo varios bloques a través de una página.

También se pueden incluir encabezados, pies de página, así como grandes totales.

El tratamiento para los campos en el generador de reportes es similar al que se realiza en la herramienta para generar formas.

5.4.4.- Herramienta para mantenimiento de menúes (Menu Maintenance tool)

El mantenimiento de menúes es una herramienta que se usa para definir y controlar las altas, bajas y cambios de las opciones de cada menú.

Un menú puede contener una o más opciones. Una opción de menú puede ser:

- un programa generado creado con la herramienta para generar programas
- otros menúes
- un comando de DCL

A cada menú y opción se le pueden agregar claves de seguridad (passwords) así como niveles de acceso.

Es posible adicionar información de ayuda en línea por cada opción de menú definida.

5.4.5.- Herramienta para generación de programas (Program Generator tool).

Empleamos esta herramienta para la generación de reglas y especificaciones que darán origen a programas generados.

Los tipos de programas que se pueden tener son:

- Entrada: permite introducir datos en línea que serán usadas para actualizar las tablas de la base de datos.
- Consultas: se utiliza para obtener datos de la base de datos y desplegarlos en línea.
- Batch: se utilizan para producir reportes o para realizar mantenimiento a tablas de la base de datos.

Los componentes de un programa pueden ser: tablas de entrada, tablas de salida, pantallas, reportes, módulos, parámetros, views y tablas RMS.

En las reglas del programa se especifican las relaciones existentes entre los componentes que lo integran.

Con todo esto el generador crea un archivo con las reglas de especificación del programa (program specification rules, PSR) que reflejan el conjunto de poderosas sentencias de especificaciones que definen un lenguaje de cuarta generación por medio del cual el generador creará un programa en cobol.

Cuando el archivo PSR esta listo, se puede generar el código fuente del programa. Se lee el archivo de especificaciones y se genera un lenguaje de programación adecuado (en este caso cobol) con sentencias inmersas de SQL para realizar los accesos a las tablas.

Una vez que el programa ha sido completado se procede a compilarlo y a ligarlo con sus módulos creando una imagen ejecutable.

5.5.- Pruebas.

La probabilidad de que existan errores o desviaciones en un sistema de información basado en computadoras recién desarrollado puede llegar a ser muy alta; de ahí que el proceso de pruebas a un sistema de este tipo deba ser exhaustivo y planeado con anterioridad.

Generalmente el tiempo que se asigna a probar un sistema es demasiado poco y las consecuencias pueden llegar a ser muy negativas cuando comienzan a aparecer errores durante la operación, ya que los ingenieros de software pierden credibilidad ante los usuarios y éstos a su vez dejan de confiar en el sistema.

No podemos hablar de una cantidad fija de tiempo y recursos que debe dedicarse a la prueba de un sistema, ya que dicha cantidad dependerá del tipo de sistema, pero si podemos decir que aproximadamente debe asignarse un 40% del tiempo total de un proyecto a probarlo.

5.5.1.- Objetivos de la prueba.

El objetivo fundamental de un proceso de pruebas es el de encontrar todos los posibles errores que tenga un programa de computadora en la menor cantidad de tiempo y esfuerzo. Como ventaja secundaria, la prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento.

Es importante no perder de vista el hecho de que el proceso de pruebas no puede asegurar la ausencia de errores; sólo puede demostrar que existen defectos en el software.

5.5.2.- Métodos de prueba.

Existen en esencia dos métodos que permiten llevar a cabo la prueba de un producto de software; el primero denominado "prueba de la caja negra" se fundamenta en el conocimiento de la función especifica para la que fue diseñado el producto; se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa. El segundo, "prueba de la caja blanca", parte del conocimiento que se tiene del funcionamiento del producto, se pueden desarrollar pruebas que aseguren que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

A continuación daremos una explicación más detallada de cada uno de estos métodos de prueba.

Prueba de la Caja Blanca.

La prueba de la caja blanca usa la estructura de control del diseño procedimental para determinar los casos de prueba; estos casos de prueba deben cumplir con las siguientes características:

- Garantizar que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo.
- Ejercitar todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecutar todos los ciclos en sus límites y con sus límites operacionales.
- Ejercitar las estructuras internas de datos para asegurar su validez.

Prueba de la Caja Negra.

Estas pruebas se centran en los requisitos funcionales del software y permiten obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Las pruebas de la caja negra deben satisfacer los siguientes criterios:

1.- Reducir en un coeficiente que es mayor que uno el número de casos de prueba adicionales que se deben diseñar para alcanzar una prueba razonable.

2.- Casos de prueba que nos dicen algo sobre la presencia o ausencia de clases de errores en lugar de un error asociado solamente con la prueba, en particular, que se encuentre disponible.

5.5.3.- Desarrollo del proceso de pruebas.

Queremos aclarar en este punto que los métodos de pruebas mencionados anteriormente no son excluyentes, sino por el contrario uno complementa al otro.

En principio utilizamos el método de prueba de la caja blanca mediante una integración de abajo hacia arriba (bottom-up) para verificar el correcto funcionamiento de cada parte del sistema, desde las rutinas elementales hasta cada uno de los módulos básicos. Cada rutina programada fue integrada al resto de rutinas codificadas y probadas, procediendo a verificar el correcto funcionamiento del conjunto.

Antes de cada prueba, elaboramos entradas y calculamos sus correspondientes salidas con el fin de compararlas con las salidas que nos proporcionaba el sistema; para esto, empleamos datos reales de operaciones de compra - venta pasadas.

Una vez corregidos los errores que arrojó la prueba de la caja blanca, procedimos a implementar la prueba de la caja negra; estas pruebas las llevamos a cabo con la ayuda de los usuarios a través de entrevistas y presentaciones previas del sistema. Afortunadamente los errores o desviaciones surgidas en esta etapa no fueron criticas y pudieron resolverse de manera muy sencilla; sin embargo estas actualizaciones nos obligaron a instrumentar la prueba de la caja blanca por segunda ocasión en aquellos programas que se vieron afectados en las presentaciones a los usuarios.

CAPíTULO VI

PUESTA EN MARCHA DEL SISTEMA.

El proceso de puesta en marcha del sistema es una de las etapas más importantes dentro del ciclo de vida del software. No importa qué tan bien haya sido el análisis y el diseño del sistema, si la puesta en marcha es deficiente es muy probable que el producto de software sea un total fracaso.

Hasta este punto, el sistema está totalmente desarrollado y cumple con todas las especificaciones establecidas durante la etapa de análisis; ahora, es necesario cumplir con una serie de actividades que permitan que el sistema comience a funcionar de la manera más eficiente dentro del ambiente operativo de la empresa.

Dentro de las actividades más importantes que conforman la etapa de puesta en marcha del sistema, podemos mencionar las siguientes:

- Capacitación en el sistema.

- Capacitación en nuevos procedimientos.

- Migración de información.

6.1.- Capacitación en el Sistema.

La capacitación a los usuarios en el sistema de información creemos que es la actividad primordial de la etapa de puesta en marcha, debido a todas las implicaciones que se pueden llegar a tener por una capacitación deficiente que puede ser traducida en un total fracaso del proyecto.

Es importante mencionar que en este punto del desarrollo del sistema se comienza a tener una fuerte relación con una gran cantidad de personas que de una u otra forma se verán involucradas con el nuevo sistema, y que por diversas razones, un porcentaje importante de dichas personas presentarán cierta resistencia al proyecto; por naturaleza las personas son reacias al cambio ya sea por que, en este caso particular, sienten amenazada su fuente de trabajo al creer que no podrán entender y adaptarse al nuevo sistema o simplemente por que piensan que la forma en que han venido trabajando es la correcta.

Tomando en consideración esta situación, es de vital importancia que los responsables del proyecto y especificamente los responsables de la capacitación desarrollen un plan que contemple no sólo la parte informativa y técnica del sistema sino que tenga en consideración a las personas a las que vaya a ir dirigida dicha capacitación.

Los capacitadores deben encontrar mecanismos que les permitan transmitir los conocimientos relacionados con el sistema al personal de la compañía y que además los convenzan de las mejoras que se tendrán, sobre todo a aquellas personas cuya forma de trabajo se verá alterada substancialmente.

Los empleados que trabajarán con el sistema deben conocer con detalle las funciones que desempeñarán, cómo utilizarán el sistema y los alcances de éste. La capacitación del usuario puede incluir el empleo del equipo, por ejemplo, en el caso de que se emplee una microcomputadora y el individuo sea

simultáneamente operador y usuario. En estos casos los usuarios deben recibir adiestramiento antes que nada en la forma de operar del equipo.

La mayor parte de la capacitación de los usuarios radica en la operación del sistema mismo. La capacitación en la codificación de datos destaca los métodos que han de seguirse para captación de datos, a partir de transacciones, o en la preparación de datos necesarios para las actividades de apoyo en la toma de decisiones.

Las actividades de manejo de datos que reciben la máxima atención dentro de la capacitación de usuarios son la entrada de nuevos datos (como almacenar nuevas transacciones), la edición de datos (como la actualización de información previamente almacenada) y la formulación de consultas (búsqueda de registros específicos). La parte más importante del uso de sistemas incluye este conjunto de actividades, por lo tanto, la mayor parte del tiempo de capacitación se orientará a esta área.

6.1.1.- Capacitación en el uso del sistema para la compañía.

Como se mencionó anteriormente, la capacitación debe ir orientada a las personas y por razones obvias no se puede pensar en un mismo plan de capacitación para todos los usuarios. En base a sus actividades laborales, a la capacidad de toma de decisiones y en general al puesto que ocupan, dividimos el universo de usuarios en tres grupos: capturistas, administrativos y funcionarios.

No obstante armamos un plan de capacitación diferente para cada uno de los tres grupos mencionados, existen una serie de temas generales básicos que deben ser transmitidos a todos los colaboradores de la compañía. Los temas

generales que forman parte de la capacitación fundamental para todos los usuarios de la casa de cambio son:

a) Aspectos básicos del manejo de terminales, microcomputadoras e impresoras.

b) Ciclo operativo de una operación cambiaria: control de deals, registro de la operación, posición por moneda y puntos relevantes de inversiones de saldos y de operaciones forward.

c) Aspectos generales del sistema de información de la casa de cambio: antecedentes (sistema actual), esquema básico operativo, configuración y alcances de la red de cómputo de la empresa, apoyo del sistema a la venta y al control de las actividades.

El plan de capacitación particular por tipo de usuario quedó conformado de la siguiente manera:

1.- Usuarios tipo capturista.

- Manejo de terminal e impresora.

- Actividades propias dentro del ciclo de operación diario de su oficina.

- Procedimientos administrativos a seguir y relación con el sistema de información.

- Módulo de registro de operaciones.

- Operaciones forward.

- Módulo de contabilidad.

- Mantenimiento a catálogos.

2.- Usuarios tipo administrativo.

- Manejo de terminal.

- Actividades propias dentro del ciclo de operación diaria de su oficina.

- Módulo de deals.

- Módulo de inversiones de saldos.

- Obtención de información del módulo de operaciones.

- Obtención de información de forwards.

- Obtención de información del módulo de inversiones.

- Obtención de información de contabilidad.

3.- Usuarios tipo funcionario.

- Manejo de microcomputadora como terminal del sistema.

- Ciclo de operación diario y puntos de control o supervisión.

- Obtención de información gerencial.

- Módulo de autorizaciones.

La capacitación incluye aspectos teóricos y prácticos, aunque la información con la que se trabaja es de ejemplos tipo de la operación diaria.

6.2.- Migración de información al nuevo sistema.

La migración es el proceso de copia de la información desde el sistema antiguo al nuevo sistema. En el caso de que el sistema antiguo se llevara de forma manual, este proceso se limitaría a registrar toda aquella información que fuera de utilidad para la operación del nuevo sistema.

La migración de información de un sistema basado en computadora a otro debe contemplar los formatos establecidos por el nuevo sistema. De manera general, el proceso de migración debe incluir los siguientes puntos:

1.- Selección de la información a migrar.

2.- Obtención de reportes de la información a migrar para ser comparados al terminar el proceso y asegurarse de mantener la misma información.

3.- Transferencia de la información al nuevo sistema.

4.- Procesamiento de la información para que quede en un formato establecido y pueda ser cargada a la base de datos.

5.- Carga de la información a la base de datos del nuevo sistema, procesándola y construyendo las estructuras lógicas apropiadas. En esta etapa es importante considerar la integridad referencial de los datos para evitar errores.

6.- Obtención de los reportes de la información migrada para su comparación con los obtenidos al comienzo del proceso.

6.2.1.- Proceso de migración en la casa de cambio.

La operación de la casa de cambio había estado basada parcialmente en un sistema de información automatizado desarrollado en DBASE e instalado en una microcomputadora. La creciente operación de la compañía, así como la apertura a nuevos mercados, fueron las principales razones que llevaron a la dirección a pensar en el desarrollo del nuevo sistema. El sistema se implementó en una micro-VAX de DEC en un ambiente de base de datos relacional (CA-DB).

El proceso de migración consistió entonces, en obtener toda la información necesaria de la microcomputadora copiando los archivos de dbase como archivos de sistema operativo y cargándolos en disckettes. Posteriormente, valiéndonos de una microcomputadora y de un software de emulación de terminal, se copiaron dichos disckettes al equipo VAX. Es importante mencionar en este punto que tanto la microcomputadora como la VAX utilizan código ASCII, de ahí que no fue necesario una decodificación de la información migrada.

Una vez teniendo la información en la VAX como archivos de VMS, se desarrollaron programas en COBOL que le dieron el formato necesario a dicha información, con el objeto de poder ser cargados a la base de datos CA-DB a través de la utilería "Load File".

Finalmente se obtuvieron reportes tanto de la información migrada (VAX) como de la información en dbase y se compararon con el fin de detectar cualquier posible error que se pudiera haber generado durante el proceso.

El proceso de migración se ejemplifica en las figuras 6.1 y 6.2

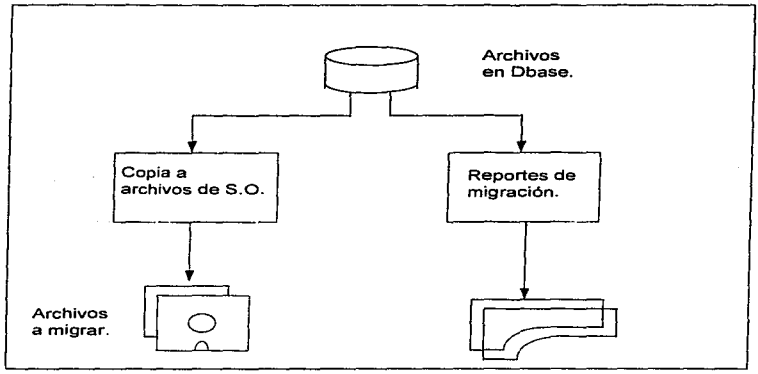


Figura 6.1 Proceso de migración de datos en microcomputadora.

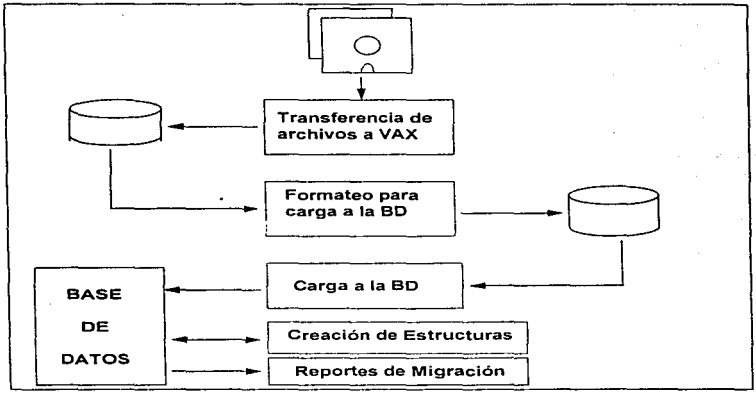


Figura 6.2 Proceso de migración de datos en VAX.

El sistema de información anterior (dbase) no contemplaba el manejo de algunos datos que por su importancia fueron considerados en el nuevo sistema; esto nos obligó a que una vez terminada la migración fuera necesario la captura de esta información.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

CAPÍTULO VII
CONCLUSIONES

Es importante mencionar que hasta antes de los años noventa, los beneficios que las tecnologías de información dieron a las organizaciones que habían invertido en ellas, no fueron como se esperaban, sino por el contrario en muchos casos provocaron un incremento en los costos:

Lester Thurow (MIT) "The Corporation of the 1990´s" afirma:

"Se pueden citar casos específicos donde las nuevas tecnologías han permitido mejorar la productividad y reducir costos, pero no se puede generalizar. No hay evidencia clara de que estas tecnologías han mejorado productividad o rentabilidad. De hecho, la afirmación contraria es cierta, al menos en los USA".

Steiner y Teixeira "Technology in Banking, 1990" afirman:

"Si bien los clientes finales se benefician de las inversiones en tecnologías de información, estas inversiones solamente han aumentado los costos de toda la industria, creando capacidad en exceso y disminuyendo la rentabilidad"

El problema grave ha sido que no se han aprovechado las oportunidades que brindan las tecnologías de información para cambiar la manera como se realiza el trabajo, esto es, en la mayoría de los casos la aplicación de dichas tecnologías se ha limitado a automatizar los procesos ya existentes de la organización sin cuestionarse si dichos procesos son los adecuados para generarle ventajas competitivas a través de reducción en costos, eficiencia en la productividad, servicios de calidad, etc.

En base a lo mencionado anteriormente, es justo mencionar que al iniciar el presente proyecto se nos otorgó la facultad de estudiar y cuestionar los procesos que conformaban la operación de la casa de cambio. Si bien la mayoría de dichos procesos venían funcionando adecuadamente y por lo tanto nos limitamos a automatizarlos, hubo uno que particularmente sufrió cambios sustanciales, nos referimos al proceso de emisión y control de deal's.

Anteriormente la empresa no prestaba importancia al manejo de deal's y en muchos casos se realizaban operaciones de compra-venta sin que éstas estuvieran soportadas por un deal, de ahí que consistentemente se tuviera el problema de omisión de operaciones por parte de empleados que se beneficiaban realizando transacciones con capital propio.

Actualmente con la implementación de un número de folio en los deal's y con el control automatizado de los mismos resulta muy fácil saber el estado del deal, su ubicación dentro de la empresa, así como la operación a la que está soportando; provocando que cualquier manejo irregular sea mucho más difícil de llevar a cabo.

El subsistema de contabilidad anteriormente se alimentaba con la recaptura de información que generaba la producción (operaciones) a través de reportes; actualmente se ha eliminado dicha recaptura y la alimentación se realiza de manera automática, lo que ha permitido eliminar fuertes cargas de trabajo al final de cada mes además de contar con información contable en cualquier momento que se desee.

En términos generales podemos afirmar que el sistema ha permitido mejorar la productividad de la organización, desde el momento de ser un sistema multiusuario en contraste con el sistema anterior que se tenía en una computadora personal.

En la práctica profesional se enfrentan problemas que por su naturaleza no pueden ser estudiados o tratados durante el desarrollo académico en la facultad, de hecho, desde nuestro punto de vista uno de los principales objetivos que se buscan con la elaboración de una tesis es precisamente que los alumnos experimenten y hagan frente a este tipo de situaciones en un plano, podríamos llamarlo, previo a la actividad profesional.

Durante el desarrollo del presente trabajo, nos vimos ante la necesidad de afrontar una serie de problemas que de alguna manera eran ajenos a nosotros pero que afectaban sustancialmente y ponían en peligro la creación del proyecto.

Poco antes de dar por terminada la fase de análisis de requerimientos, la empresa para la cual se desarrolló el sistema, sufrió un cambio de dirección a nivel corporativo; la nueva dirección congeló todos los proyectos y el nuestro no fue la excepción; ésto significó que perdiéramos el apoyo de los niveles ejecutivos para llevar a buen fin el objetivo de nuestro trabajo de tesis. Afortunadamente, contamos siempre con el apoyo del gerente del área de cambio de divisas quien nos alentó y dedicó parte de su tiempo y del de su gente para que continuáramos, aunque de manera extraoficial, con el desarrollo del proyecto.

Otro grave problema que tuvimos que enfrentar fue el hecho de que al poco tiempo de que el nuevo personal ejecutivo tomara la dirección de la empresa, se presentó un fuerte recorte de personal, lo que propició que muchas de las personas con las que habíamos venido trabajando y de las cuales esperábamos decisiones importantes que podían afectar nuestro trabajo quedaran fuera de la organización.

Si bien al cabo del tiempo pudimos salvar los problemas que se nos presentaron, éstos se vieron traducidos en un retraso importante respecto a la fecha en que pensábamos tener terminado el proyecto.

Finalmente, queremos mencionar que una vez terminada la fase de implementación del sistema, se presentó el proyecto y su avance a la dirección y ésta aprobó la implantación del mismo cumpliendose con los objetivos planteados al inicio del desarrollo del presente trabajo de tesis.

BIBLIOGRAFÍA

- Catherin Mansell Carstens.
  Nuevas Finanzas en México.
- Date, C. J.
  Introducción a los sistemas de bases de datos.
- Davo, Ramón
  Personal Computing México 1994-1996
- Facultad de Ingeniería, U.N.A.M.
  Apuntes "Diseño de bases de datos", 1992
- Fairley, Richard.
  Ingeniería de software, 1988.
- Finanzas Corporativas.
  Ross - Westerfield - Jaffe
- Hawkins, John L.
  Data Based Advisor, 1994-1995.
- H. Riehl - Rodriguez.
  Mercados de divisas y mercados de dinero.
- Pressman, Roger S.
  Ingeniería de software, 1993.
- Senn, James A.
  Análisis y diseño de sistemas de información, 1988.
- Swiss Bank Corporation.
  El cambio de divisas y operaciones del mercado monetario.

- Tsai, Alice Y.H.
  Sistemas de bases de datos, 1990.

- Yourdon, Edward.
  Análisis estructurado moderno, 1993.

ANEXO 1

PROCESO DE CREACIÓN
DE LA BASE DE DATOS.

```
$!*****************************************************************************************
$!*                                                                                       *
$!*  SISTEMA DE "CASA DE CAMBIO"                                                          *
$!*  PROCESO DE CREACIÓN DE LA BASE DE DATOS                                              *
$!*  DESARROLLARON:                    - MARTINEZ VALENZUELA HECTOR                       *
$!*                                    - TERRAZAS SOLIS VLADIMIR                          *
$!*                                                                                       *
$!*****************************************************************************************
$!
$ INICIO:
$!
$!*****************************************************************************************
$!*  DEFINICIÓN DE NOMBRES LÓGICOS PARA LOS FILES DE LA BASE DE DATOS                     *
$!*****************************************************************************************
$!
$ DBVAX_LNMS CRELNM DIVISIONES_HSH APS03:[BASE_DATOS]DIVISIONES.HASH
$ DBVAX_LNMS CRELNM SUCURSALES_HSH APS03:[BASE_DATOS]SUCURSALES.HASH
$ DBVAX_LNMS CRELNM DIVISAS_HSH APS03:[BASE_DATOS]DIVISAS.HASH
$ DBVAX_LNMS CRELNM ESPECIES_HSH APS03:[BASE_DATOS]ESPECIES.HASH
$ DBVAX_LNMS CRELNM PUESTOS_HSH APS03:[BASE_DATOS]PUESTOS.HASH
$ DBVAX_LNMS CRELNM FESTIVOS_HSH APS03:[BASE_DATOS]FESTIVOS.HASH
$!*
$ DBVAX_LNMS CRELNM USUARIOS_MIX APS03:[BASE_DATOS]USUARIOS.MIXTO
$ DBVAX_LNMS CRELNM RANGOS_MIX APS03:[BASE_DATOS]RANGOS.MIXTO
$ DBVAX_LNMS CRELNM BANCOS_MIX APS03:[BASE_DATOS]BANCOS.MIXTO
$ DBVAX_LNMS CRELNM CHEQUERAS_MIX APS03:[BASE_DATOS]CHEQUERAS.MIXTO
$ DBVAX_LNMS CRELNM POSICION_MIX APS03:[BASE_DATOS]POSICION.MIXTO
$ DBVAX_LNMS CRELNM MIG_CONTA_MIX APS03:[BASE_DATOS]MIG_CONTA.MIXTO
$ DBVAX_LNMS CRELNM CUENTAS_MIX APS03:[BASE_DATOS]CUENTAS.MIXTO
$ DBVAX_LNMS CRELNM CLIENTES_MIX APS03:[BASE_DATOS]CLIENTES.MIXTO
$ DBVAX_LNMS CRELNM FUTUROS_MIX APS03:[BASE_DATOS]FUTUROS.MIXTO
$ DBVAX_LNMS CRELNM FORMULAS_MIX APS03:[BASE_DATOS]FORMULAS.MIXTO
$ DBVAX_LNMS CRELNM SISTEMA_MIX APS03:[BASE_DATOS]SISTEMA.MIXTO
$ DBVAX_LNMS CRELNM INVERSION_MIX APS03:[BASE_DATOS]INVERSION.MIXTO
$ DBVAX_LNMS CRELNM STOCK_DIV_MIX APS03:[BASE_DATOS]STOCK_DIV.MIXTO
$ DBVAX_LNMS CRELNM AUTORIZACION_MIX APS03:[BASE_DATOS]AUTORIZACION.MIXTO
$ DBVAX_LNMS CRELNM AUTORIZADOR_MIX APS03:[BASE_DATOS]AUTORIZADOR.MIXTO
$!*
$ DBVAX_LNMS CRELNM DEALS_DAT APS03:[BASE_DATOS]DEALS.DATOS
$ DBVAX_LNMS CRELNM DEALS_NDX APS03:[BASE_DATOS]DEALS.INDICE
$ DBVAX_LNMS CRELNM OPERACION_DAT APS03:[BASE_DATOS]OPERACION.DATOS
$ DBVAX_LNMS CRELNM OPERACION_NDX APS03:[BASE_DATOS]OPERACION.INDICE
$ DBVAX_LNMS CRELNM DET_INVER_DAT APS03:[BASE_DATOS]DET_INVER.DATOS
$ DBVAX_LNMS CRELNM DET_INVER_NDX APS03:[BASE_DATOS]DET_INVER.INDICE
$!*
$!*****************************************************************************************
$!*  DEFINICIÓN DEL NOMBRE LÓGICO PARA LA BASE DE DATOS Y JOURNAL                         *
$!*****************************************************************************************
$!
$ DBVAX_LNMS CRELNM CASA_DB APS03:[BASE_DATOS]CASA_DB ROOT,-
$ APS03:[BASE_DATOS]CASA_DB.JOURNAL1
$!
$!*****************************************************************************************
$!*  CREACION DE LA BASE DE DATOS                                                         *
$!*****************************************************************************************
$!*
$ SQL
  CREATE DB CASA_DB DATA PAGES=10000 JOURNAL PAGES=30000;
  TERMINATE DB;
```

```
  START DB CASA_DB MULTI NONPAGED;
  RELEASE;
  QUIT;
$!*
$!*
$!*****************************************************************************************
$!*  CREACION DE ESPACIOS DE TABLAS                                                       *
$!*****************************************************************************************
$!*
$ SQL
  CONNECT TO CASA_DB;
  CREATE TABLESPACE DIVISIONES;
  CREATE TABLESPACE SUCURSALES;
  CREATE TABLESPACE DIVISAS;
  CREATE TABLESPACE ESPECIES;
  CREATE TABLESPACE PUESTOS;
  CREATE TABLESPACE FESTIVOS;
  CREATE TABLESPACE USUARIOS;
  CREATE TABLESPACE RANGOS;
  CREATE TABLESPACE BANCOS;
  CREATE TABLESPACE CHEQUERAS;
  CREATE TABLESPACE POSICION;
  CREATE TABLESPACE MIG_CONTA;
  CREATE TABLESPACE CUENTAS;
  CREATE TABLESPACE CLIENTES;
  CREATE TABLESPACE FUTUROS;
  CREATE TABLESPACE FORMULAS;
  CREATE TABLESPACE SISTEMA;
  CREATE TABLESPACE INVERSION;
  CREATE TABLESPACE STOCK_DIV;
  CREATE TABLESPACE AUTORIZACION;
  CREATE TABLESPACE AUTORIZADOR;
  CREATE TABLESPACE DEALS;
  CREATE TABLESPACE OPERACION;
  CREATE TABLESPACE DET_INVER;
  RELEASE;
  QUIT;
$!*
$!*****************************************************************************************
$!*  CREACION DE ARCHIVOS PARA LAS TABLAS DE LA BASE DE DATOS                             *
$!*****************************************************************************************
$!*
$ SQL
  CONNECT TO CASA_DB;
  CREATE FILE DIVISIONES_HSH    WITH PAGES=5   NAME=DIVISIONES_HSH    TYPE=TABLE;
  CREATE FILE SUCURSALES_HSH WITH PAGES=5   NAME=SUCURSALES_HSH TYPE=TABLE;
  CREATE FILE DIVISAS_HSH       WITH PAGES=7   NAME=DIVISAS_HSH       TYPE=TABLE;
  CREATE FILE ESPECIES_HSH      WITH PAGES=5   NAME=ESPECIES_HSH      TYPE=TABLE;
  CREATE FILE PUESTOS_HSH       WITH PAGES=5   NAME=PUESTOS_HSH       TYPE=TABLE;
  CREATE FILE FESTIVOS_HSH      WITH PAGES=5   NAME=FESTIVOS_HSH      TYPE=TABLE;
  CREATE FILE USUARIOS_MIX      WITH PAGES=10  NAME=USUARIOS_MIX      TYPE=MIXED;
  CREATE FILE RANGOS_MIX        WITH PAGES=100 NAME=RANGOS_MIX        TYPE=MIXED;
  CREATE FILE BANCOS_MIX        WITH PAGES=10  NAME=BANCOS_MIX        TYPE=MIXED;
  CREATE FILE CHEQUERAS_MIX     WITH PAGES=10  NAME=CHEQUERAS_MIX     TYPE=MIXED;
  CREATE FILE POSICION_MIX      WITH PAGES=50  NAME=POSICION_MIX      TYPE=MIXED;
  CREATE FILE MIG_CONTA_MIX     WITH PAGES=200 NAME=MIG_CONTA_MIX     TYPE=MIXED;
  CREATE FILE CUENTAS_MIX       WITH PAGES=20  NAME=CUENTAS_MIX       TYPE=MIXED;
  CREATE FILE CLIENTES_MIX      WITH PAGES=100 NAME=CLIENTES_MIX      TYPE=MIXED;
  CREATE FILE FUTUROS_MIX       WITH PAGES=200 NAME=FUTUROS_MIX       TYPE=MIXED;
  CREATE FILE FORMULAS_MIX      WITH PAGES=20  NAME=FORMULAS_MIX      TYPE=MIXED;
  CREATE FILE SISTEMA_MIX       WITH PAGES=5   NAME=SISTEMA_MIX       TYPE=MIXED;
```

```
  CREATE FILE INVERSION_MIX      WITH PAGES=20   NAME=INVERSION_MIX      TYPE=MIXED;
  CREATE FILE STOCK_DIV_MIX      WITH PAGES=100  NAME=STOCK_DIV_MIX      TYPE=MIXED;
  CREATE FILE AUTORIZACION_MIX WITH PAGES=50  NAME=AUTORIZACION_MIX TYPE=MIXED;
  CREATE FILE AUTORIZADOR_MIX  WITH PAGES=30  NAME=AUTORIZADOR_MIX  TYPE=MIXED;
  CREATE FILE DEALS_DAT          WITH PAGES=1500 NAME=DEALS_DAT          TYPE=TABLE;
  CREATE FILE DEALS_NDX          WITH PAGES=1000 NAME=DEALS_NDX          TYPE=INDEX;
  CREATE FILE OPERACION_DAT  WITH PAGES=3000  NAME=OPERACION_DAT     TYPE=TABLE;
  CREATE FILE OPERACION_NDX  WITH PAGES=2000 NAME=OPERACION_NDX     TYPE=INDEX;
  CREATE FILE DET_INVER_DAT    WITH PAGES=3000  NAME=DET_INVER_DAT     TYPE=TABLE;
  CREATE FILE DET_INVER_NDX    WITH PAGES=2000 NAME=DET_INVER_NDX     TYPE=INDEX;
  RELEASE;
  QUIT;
$!*
$!*****************************************************************************************
$!*    RELACION DE ARCHIVOS A LOS ESPACIOS DE TABLA DE LA BASE DE DATOS          *
$!*****************************************************************************************
$!*
$ SQL
  CONNECT TO CASA_DB;
  ADD FILE DIVISIONES_HSH TO TABLESPACE DIVISIONES;
  ADD FILE SUCURSALES_HSH TO TABLESPACE SUCURSALES;
  ADD FILE DIVISAS_HSH TO TABLESPACE DIVISAS;
  ADD FILE ESPECIES_HSH TO TABLESPACE ESPECIES;
  ADD FILE PUESTOS_HSH TO TABLESPACE PUESTOS;
  ADD FILE FESTIVOS_HSH TO TABLESPACE FESTIVOS;
  ADD FILE USUARIOS_MIX TO TABLESPACE USUARIOS;
  ADD FILE RANGOS_MIX TO TABLESPACE RANGOS;
  ADD FILE BANCOS_MIX TO TABLESPACE BANCOS;
  ADD FILE CHEQUERAS_MIX TO TABLESPACE CHEQUERAS;
  ADD FILE POSICION_MIX TO TABLESPACE POSICION;
  ADD FILE MIG_CONTA_MIX TO TABLESPACE MIG_CONTA;
  ADD FILE CUENTAS_MIX TO TABLESPACE CUENTAS;
  ADD FILE CLIENTES_MIX TO TABLESPACE CLIENTES;
  ADD FILE FUTUROS_MIX TO TABLESPACE FUTUROS;
  ADD FILE FORMULAS_MIX TO TABLESPACE FORMULAS;
  ADD FILE SISTEMA_MIX TO TABLESPACE SISTEMA;
  ADD FILE INVERSION_MIX TO TABLESPACE INVERSION;
  ADD FILE STOCK_DIV_MIX TO TABLESPACE STOCK_DIV;
  ADD FILE AUTORIZACION_MIX TO TABLESPACE AUTORIZACION;
  ADD FILE AUTORIZADOR_MIX TO TABLESPACE AUTORIZADOR;
  ADD FILE DEALS_DAT TO TABLESPACE DEALS;
  ADD FILE DEALS_NDX TO TABLESPACE DEALS;
  ADD FILE OPERACION_DAT TO TABLESPACE OPERACION;
  ADD FILE OPERACION_NDX TO TABLESPACE OPERACION;
  ADD FILE DET_INVER_DAT TO TABLESPACE DET_INVER;
  ADD FILE DET_INVER_NDX TO TABLESPACE DET_INVER;
  RELEASE;
  QUIT;
$!*
$!*****************************************************************************************
```