

6  
24

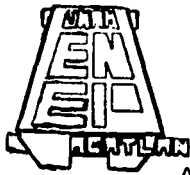


UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

-----  
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
"ACATLAN"

MANEJADOR DE BASE DE DATOS Y  
BASE DE DATOS EN EQUIPOS  
MAIN FRAME.

TESINA PROFESIONAL  
QUE PARA OBTENER EL TITULO DE  
LIC. EN MATEMATICAS APLICADAS Y  
COMPUTACION  
P R E S E N T A :  
LUIS FERNANDO CASTILLO FERNANDEZ



ACATLAN, ESTADO DE MEXICO



FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México



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

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

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

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



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "ACATLAN"

PROGRAMA DE ACTUARIA Y M.A.C.

SR. LUIS FERNANDO CASTILLO FERNANDEZ

Alumno de la carrera de Matemáticas Aplicadas  
y Computación.  
Presente.

De acuerdo a su solicitud presentada con fecha 19 de agosto de 1993, me complace notificarle que este Programa tuvo a bien asignarle el siguiente tema de tesis: "Manejador de Base de Datos y Base de Datos en Equipos Main Frame", el cual se desarrollará como sigue:

- I.- Introducción
- II.- Definiciones
- III.- Historia del DB2 y SQL
- IV.- Base de datos
- V.- Manejador de base de datos
- VI.- Cross y system product (CSP)
- VII.- Caso práctico

Así mismo fue designado como asesor de tesis el FIS. MAT. - JORGE LUIS SUAREZ MADARIAGA.

Ruego a usted tomar nota que en cumplimiento de lo especificado en la Ley de Profesiones, deberá prestar servicio social durante un tiempo mínimo de seis meses como requisito básico para sustentar examen profesional, así como de la disposición de la Coordinación de la Administración Escolar, en el sentido de que se imprimo en lugar visible de los ejemplares de su tesis el título del trabajo realizado. Esta comunicación deberá imprimirse en el interior de la tesis.

Atentamente

"POR MI RAZA HABLARA EL ESPÍRITU"  
Acatlán, Edo. de Méx., a 27 de abril de 1995.

ACT. LAURA MARIANA RIVERA BUCERRAS Y COMPUTACIÓN  
Jefe del Programa de Actuaría y M.A.C.

FALLA DE ORIGEN

---

**SEÑOR:**

Concédeme:

*Serenidad* para aceptar las cosas que  
no puedo cambiar.

*Valor* para cambiar las que si puedo y.

*Sabiduría* para discernir las diferencias.

---

---

**A Dios**

*Por darme vida, salud y  
sabiduría para verme  
realizado como persona.*

*Gracias por poner en mi  
camino personas que me  
han ayudado en mi vida  
personal y profesional.*

**A la UNAM y sus Maestros**

*Por darme la oportunidad, la formación y  
el apoyo brindado para culminar  
exitosamente mi carrera profesional.*

*Deseo fervientemente ser un egresado de  
quien estés siempre orgullosa.*

**A mi Madre**

**Gloria Fernández**

*Con amor y como un  
tributo a tu espíritu  
inquebrantable.*

**A mis Hermanos**

**Perla, Rogelio y Daniel**

*Con cariño, por su amistad y apoyo  
incondicional, que me han ofrecido hoy y  
siempre.*

---

---

***A mis familiares***

*Como reconocimiento a sus  
valiosos consejos y por  
brindarme sin reservas su  
apoyo, confianza y cariño.*

***A mis amigos***

*Por el apoyo de sus conocimientos para  
terminar satisfactoriamente mi  
licenciatura y este trabajo.*

***A mis primos y sobrino  
Arián-Karin, Viktor y Luis  
Daniel***

*Por su cariño  
incondicional en todo  
momento y por ser el futuro  
de nuestra familia.*

***Con Agradecimiento***

*A todos aquellos que de una u otra  
forma contribuyeron a mi formación  
profesional.*

---

INDICE DE CONTENIDO

	Página
<b>Introducción</b> .....	iii
<b>I. DEFINICIONES</b>	
1.1 Maxicomputadoras o main frames.....	1
1.2 Base de datos.....	3
1.2.1 Base de datos relacionales .....	3
1.3 Manejador de base de datos .....	6
1.4 Instituciones de Crédito	
1.4.1 Banco.....	9
1.4.2 Casa de Bolsa.....	10
<b>II. HISTORIA DE SQL Y DB2</b> .....	13
<b>III. BASE DE DATOS</b>	
3.1 Tipo de Base de datos	
3.1.1 Base de datos Jerárquica .....	21
3.1.2 Base de datos de red.....	23
3.1.3 Base de datos distribuidas o relacionales .....	26
3.1.4 Base de datos orientadas a objetos .....	28
3.2 Teoría del Dr. Codd .....	32
3.3 DB2 (Data Base Dos).....	38
<b>IV. MANEJADOR DE BASE DE DATOS</b>	
4.1 Características y beneficios de SQL .....	47
4.2 Futuro de SQL.....	55
<b>V. LENGUAJE DE CUARTA GENERACIÓN</b>	
5.1 Introducción de lenguaje de cuarta generación.....	56
5.2 CSP (Cross System Product) .....	57
5.2.1 Concatenación de Bibliotecas .....	59
5.2.2 Componentes de las aplicaciones.....	60
5.2.3 Definición de registros .....	61
5.2.4 Definición de tablas o working's de trabajo .....	62
5.2.5 Definición de mapas.....	62
5.2.6 Definición de la lógica de la aplicación .....	63

**VI. CASO PRACTICO**

6.1 Mercado de coberturas cambiarias.....	64
6.1.1 Participantes del mercado de coberturas cambiarias de corto plazo.....	65
6.1.2 Tipo de Cambio	
6.1.2.1 Dólar ventanilla libre bancario.....	66
6.1.2.2 Tipo de cambio para operaciones de mayoreo.....	66
6.1.2.3 Tipo de cambio libre representativo.....	67
6.1.3 ¿Cómo funciona la compra de coberturas cambiarias?.....	67
6.1.4 ¿Cómo funciona la venta de coberturas cambiarias?.....	72
6.1.5 ¿Cómo se determina el precio de una cobertura Cambiaria?.....	76
6.1.6 ¿Cuáles son los plazos posibles para celebrar operaciones de coberturas cambiarias?.....	76
6.1.7 ¿Con qué garantías debe cumplir los clientes?.....	76
6.2 Sistema de coberturas cambiarias de la casa de bolsa Pro bursa	
6.2.1 Arquitectura del equipo.....	71
6.2.2 Módulos del sistema.....	79
6.2.3 Narrativa del programa de Coberturas Cambiarias para Tasas.....	82
6.3 El futuro de los main frames y DB2.....	93
<b>Glosario.....</b>	<b>96</b>
<b>Conclusiones.....</b>	<b>113</b>
<b>Bibliografía.....</b>	<b>115</b>



---

## INTRODUCCIÓN

La invención de las computadoras revolucionó los métodos tradicionales del procesamiento de la información. A principios de los años 60, varias firmas comerciales empezaron a computarizar sus sistemas de información, en un principio fueron de gran utilidad las estructuras primitivas de almacenamiento para la manipulación de la información, pero a medida que ha pasado el tiempo se han desplazado por mecanismos más rápidos y económicos para la gestión de la información.

El procesamiento de la información es esencial para la administración de los negocios, el gobierno y la educación. En la sociedad es vital para una organización o empresa proporcionar información correcta y oportuna para apoyar la toma de decisiones y otras actividades gerenciales. Como resultado del crecimiento económico y avances tecnológicos, muchas organizaciones han crecido, tanto en tamaño, como en la sofisticación de sus funciones administrativas. Mientras el volumen de procesamiento de datos crece a una rapidez sin precedente, también crece la demanda de medios eficientes para manejarlos.

El objetivo del presente trabajo es presentar una recopilación de información, destacando la importancia que ha tenido la creación del manejador de base de datos relacional DB2 para equipos main frames, administrados por software de IBM, corriendo bajo el sistema operativo MVS (*Multiple Virtual Storage*); también se hará mención al impacto que han tenido y el futuro que les espera a los equipos main frame; ambos aspectos en base a la experiencia real de un sistema de información desarrollado para la Casa de Bolsa Probusa.

Este trabajo está enfocado para personas que se dedican al desarrollo de sistemas, por lo que se asume que los interesados, cuentan con los conocimientos básicos de computación; es por ello que algunos de los conceptos, términos y lenguajes de programación más usuales en el área informática, no son explicados con profundidad.

En la actualidad los manejadores de base de datos y las bases de datos relacionales constituyen la forma más común y económica de almacenamiento, administración y organización de grandes cantidades de información; DB2 fue el primer manejador de base

---

de datos relacional que utilizó SQL (*Structure Query Language*) como lenguaje nativo. Hoy en día SQL es el lenguaje motor de cualquier manejador de base de datos relacional, tanto para equipos grandes como pequeños. Esto se debe en gran parte a que ha sido reconocido como el estándar de los manejadores de bases de datos relacionales por las organizaciones ANSI (*American National Standard Institution*) e ISO (*International Standard Organization*).

El presente trabajo está dividido en seis capítulos de acuerdo a los siguientes temas:

*Definiciones.* En forma breve, pero completa, se presentan los principales conceptos relacionados con el tema central de mi trabajo, los cuales son: Main frame, Base de Datos, Manejador de Base de Datos, Banco y Casa de Bolsa.

*Historia de SQL y DB2.* Basado en ediciones de texto, donde se describe el surgimiento de diferentes manejadores de bases de datos, se expondrán los aspectos generales del manejador de bases de datos DB2 y del lenguaje SQL, así como los periodos en que ambos se fueron desarrollando.

*Bases de Datos.* Para comprender de una manera más clara y sencilla los conceptos de manejador de base de datos y base de datos relacional, en este capítulo se exponen las características de cada uno de los tipos de base de datos existentes, destacando lo concerniente al tipo de base de datos relacional, para lo cual se incluye la teoría del Dr. Codd: "*Un modelo relacional de datos para bancos de datos compartidos*", en donde se describen las doce reglas que una base de datos debe obedecer para ser considerada relacional. Posteriormente se describen las funciones que se realizan con la base de datos relacional DB2, haciendo especial énfasis en sus ventajas y desventajas.

*Manejador de Base de Datos.* Como se mencionó, SQL ha evolucionado a tal grado que ha sido reconocido por la ISO como un estándar de lenguaje para bases de datos relacionales, por lo que ha llegado a considerarse no solo como un lenguaje de consultas, sino como un verdadero manejador de bases de datos relacionales (RDBM's); debido a esto se considera importante mencionar los beneficios que SQL aporta al ámbito de gestión de grandes cantidades de información, así como el futuro que tal vez le depara.

*Lenguaje de Cuarta Generación.* Este capítulo inicia con una breve introducción a los lenguajes de cuarta generación (L4G), para en seguida presentar con más detalle las características y funciones de CSP (Cross System Product), L4G que realiza procesos **on-line** (ver glosario, Arquitectura de un Main frame) para equipos main frame. Se

---

mencionarán los procedimientos más importantes que se deben realizar con dicho lenguaje, ya que éste se utilizó en el desarrollo de la aplicación incluida como caso práctico.

*Caso Práctico.* En este capítulo se describen los detalles sobre el funcionamiento del Sistema de Coberturas Cambiarias de Probursa, que tiene como finalidad ver la importancia de estos equipos en la operaciones del mercado bursátil, la inclusión de este caso práctico no pretende dar a conocer el funcionamiento de una institución bancaria o de una casa de bolsa.

Para concluir, en la última sección de este capítulo se comenta un punto de vista final del futuro en computación (en lo que se refiere a equipos main frame y DB2), resumiendo los enfoques de diversas publicaciones (revistas y periódicos) especialistas en la materia.

## CAPITULO I

### DEFINICIONES

#### 1.1 MAXICOMPUTADORA O MAIN FRAME

Una maxicomputadora o main frame es un equipo con gran capacidad o almacenamiento tanto en memoria real como virtual, así como también en dispositivos auxiliares (discos magnéticos, etc.).

Puede manejar una gran cantidad de usuarios trabajando en forma simultánea soportando toda la tecnología actual de comunicaciones.

Uno de los conceptos importantes aquí es el de memoria virtual, que consiste en multiplicar la capacidad de memoria real en una cantidad considerable.

Una maxicomputadora se puede conectar a través de diversos protocolos de comunicación con diferentes equipos como maxicomputadoras, minicomputadoras, PC's, etc.; dichos equipos también son conocidos como emuladores en el ambiente de main frames.

Las características de un main frame son:

- ✓ Requiere trabajar con sistemas operativos MVS (Multiple Virtual Storage), MV (*Virtual Machine*), VSE (*Virtual Storage Extended*), etc.<sup>1</sup>
- ✓ Soporte de múltiples usuarios concurrentes.
- ✓ Manejo de varias colas de impresión.

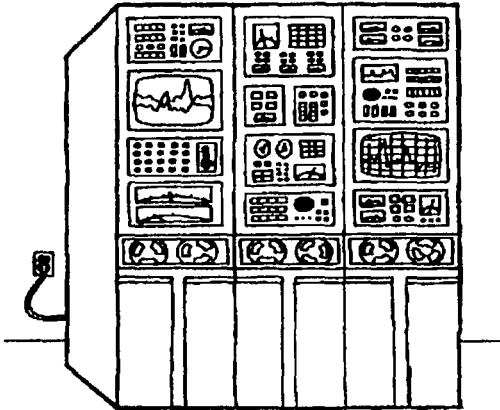
---

<sup>1</sup> Sólo se mencionan los sistemas operativos para equipos que corren en ambiente de IBM.

- ✓ Comparte dispositivos.
- ✓ Posee una memoria real desde 384 Mb hasta más de 2 Gigabytes.
- ✓ Multiprogramación.
- ✓ Multiprocesos.
- ✓ Soporta más de 8 Gigabytes, de memoria virtual.

El tipo de sistema operativo con que cuenta una maxicomputadora, le da las características a los recursos que administra (memoria, procesadores, equipos periféricos, etc. ).

Algunos de los fabricantes de este tipo de máquinas son: **DEC, IBM, NCR, DIGITAL, BURROUGHS.**



## 1.2 BASE DE DATOS

Martín<sup>2</sup> define básicamente el concepto de base de datos, como "Una colección de base de datos interrelacionados almacenados en conjunto, sin redundancias perjudiciales o innecesarias; su finalidad es la de servir al mayor número de aplicaciones, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los utilizan; se emplean métodos específicos para incluir, modificar o extraer datos. La base de datos está organizada, de modo que los datos que contiene pueden ser recuperados de muchas maneras diferentes y ser utilizados para responder a diversas interrogantes".

Date<sup>3</sup>, la define como "Un depósito de datos almacenados, y en general, es tanto integrada como compartida". Por *integrada* se entiende que la base de datos puede considerarse como una unificación de varios archivos de datos independientes, donde se eliminan parcial o totalmente cualquier redundancia entre los mismos. Por *compartida*, que partes individuales de la base de datos puede compartirse entre varios usuarios distintos, en el sentido de que cada uno de ellos puede tener acceso concurrente a la misma parte de la base de datos y utilizarla con propósitos diferentes.

### 1.2.1 Base de datos relacional

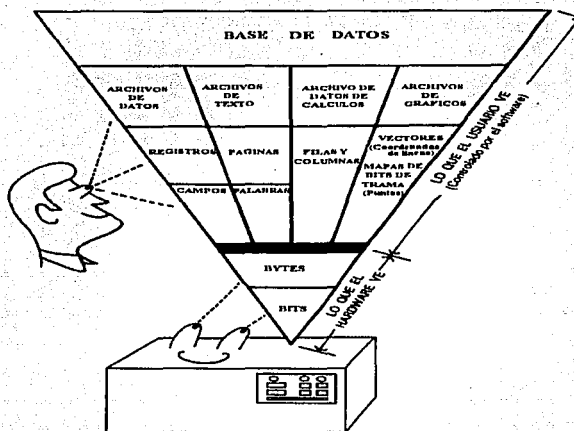
"Una base de datos es una forma de organizar la información. En ellas la información normalmente se organiza y se mantiene en una tabla compuesta por filas y columnas. Cada fila está relacionada con las otras porque todas ellas contienen el mismo tipo de información establecida en un orden determinado. A las filas en un archivo de base de datos se llaman registros y las columnas se llaman campos. Una base de datos relacional es una base de datos donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas"<sup>4</sup>.

---

<sup>2</sup> Martín, James, "Computer Data Base Organization", Prentice-Hall, 1977.

<sup>3</sup> Date, C. J., "An Introduction to Data Base System", 3th. Ed. Addison Wesley, 1986.

<sup>4</sup> Groff, James R. "Aplicase SQL", McGraw-Hill, 1992.



A continuación se describen las ventajas y desventajas de utilizar base de datos relacional, como estructuras de almacenamiento de información.

#### VENTAJAS

- ✓ Evitan la redundancia.
- ✓ Acceso rápido.

- ✓ Facilita la recuperación de datos.
- ✓ Integridad de la información.
- ✓ Optimización de hardware.
- ✓ Facilita la administración de la información.
- ✓ Facilita el desarrollo de aplicaciones.
- ✓ Existe una estandarización a nivel global.
- ✓ Ofrece seguridad de la información, ya que permite
  - Especificar niveles de acceso para los usuarios
  - Realizar respaldos

**DESVENTAJAS**

- ✓ Diseños, instalaciones y manejos complejos.
- ✓ Costos elevados.
- ✓ Mantenimiento y servicio a la Base Datos.
- ✓ Vulnerabilidad.



### 1.3 MANEJADOR DE BASE DE DATOS

Durante los inicios de la década de los 70's, tuvo lugar el desarrollo de paquetes de software por varias compañías, con el objetivo de minimizar las dificultades para integrar los datos que contiene una base de datos. Esos paquetes de software empezaron a ser conocidos como Sistemas Manejadores de Bases de Datos (DBMS).

La utilización de DBMS's representa una de las tendencias más significativas en el campo de los sistemas de información basados en computadoras. Básicamente un DBMS es un sistema de gestión de base de datos, siendo ésta su característica más importante.

El objetivo primordial de un DBMS es crear un ambiente en el cual sea posible guardar y recuperar información de la base de datos en forma conveniente y eficiente.

Un DBMS, es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Los DBMS se diseñaron para manejar grandes cantidades de información. El manejo de los datos incluye tanto la definición de la estructura para el almacenamiento de la información, como los mecanismos para el manejo de la misma.

Debido a la importancia que tiene la información en las organizaciones, la base de datos es un recurso muy valioso. Esta condujo al desarrollo de un gran número de conceptos y técnicas para manejar los datos en forma eficiente.

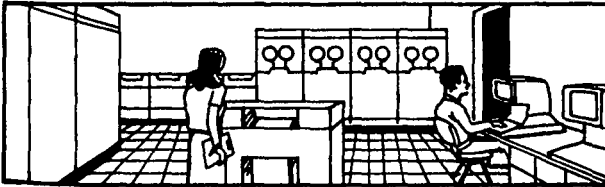
Un DBMS puede ser visualizado como una interfase entre el software de acceso a las tablas de una base de datos, y los programas de aplicaciones.

Las funciones que realiza un manejador de base de datos son:

- 1. DEFINICIÓN DE LOS DATOS** El usuario puede definir la estructura y organización de los datos almacenados y las relaciones entre ellos.
- 2. RECUPERACIÓN DE DATOS** Permite a un usuario o a un programa de aplicación recuperar los datos almacenados de la base de datos y utilizarlos.
- 3. MANIPULACIÓN DE DATOS** Facilita a un usuario o a un programa de aplicaciones actualizar la base de datos añadiendo nuevos datos, ya sea suprimiendo y/o modificando los datos.
- 4. CONTROL DE ACCESO** Puede ser utilizado para restringir la capacidad de un usuario para recuperar, añadir y modificar datos, así los datos almacenados son protegidos.
- 5. COMPARTICIÓN DE DATOS** Se utiliza para coordinar la compartición de datos por parte de usuarios concurrentes.

**6. INTEGRIDAD DE DATOS**

Define restricciones en la base de datos, protegiéndola contra corrupciones debido a actualizaciones inconsistentes o fallos del sistema.



## 1.4 INSTITUCIONES DE CREDITO

### 1.4.1 Banco

La función primordial de los bancos es captar los recursos (saldos de cuentas de cheques, certificados de depósito, pagaré con rendimiento liquidable al vencimiento, etc.) por cuenta propia.

Esto es, los bancos incorporan a su balance como deudas <<pasivos>> todos los recursos captados por ello; es decir, al llegar los fondos a un banco, independiente del canal por el que arriben, automáticamente el banco los absorbe y los incorpora a su contabilidad propia. Al otorgar préstamos sobre esos fondos, el banco asume plena responsabilidad sobre esos préstamos.



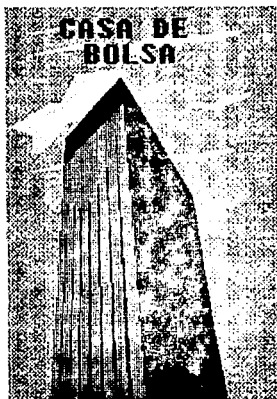
## Principales fuentes de captación y destino de los recursos

Usuarios de fondos	Excedentes de fondos captados a través de:
Préstamos personales	Cuentas de cheques
Crédito directos	Certificados de depósito
Préstamos de habilitación o avío	Pagare con rendimiento liquidable al vencimiento
Préstamos refaccionarios	Cuenta de ahorro
Préstamos hipotecarios	
Préstamos prendario	
Gobierno federal (Reservas de los bancos)	

## 1.4.2 Casa de Bolsa

La bolsa de valores es un mercado organizado de intermediarios que representan los intereses de particulares, sociedades mercantiles y del mismo estado en el libre intercambio de valores, dentro de las reglas establecidas, tanto por este último a través de sus diversas dependencias, como por los intermediarios que participan en el propio mercado.

*Agentes de valores* es el nombre con el que designa a los intermediarios en el mercado de valores. Los agentes de bolsa son personas morales que reciben el nombre de *casa de bolsa*.



Los valores que se negocian a través de bolsa son acciones, obligaciones, petrobonos, bonos de indemnización bancaria, certificados de la tesorería, papel comercial y aceptaciones bancarias.

La estructura formal sobre la que descansan y dentro de la cual se desenvuelven las operaciones bursátiles en México es la Bolsa Mexicana de valores, S.A. de C.V.

**Principales fuentes de captación y destino de los recursos**

Usuarios de fondos	Excedentes de fondos captados a través de:
El gobierno Federal Las empresas emisoras	CETES Papel comercial Aceptaciones bancarias Petrobonos Acciones Bonos de indemnización bancaria Obligaciones

## CAPITULO II

### HISTORIA DE SQL Y DB2

SQL es a la vez un lenguaje fácil de entender y una herramienta completa para la gestión de datos en computadoras personales, minicomputadoras y maxicomputadoras. A continuación se mencionan algunas de las características de SQL, que le han dado fuerza en el mercado de la computación, esta son: su independencia con los vendedores, su portabilidad a través de sistemas informáticos y el apoyo de IBM, entre otras.

DB2 es el primer manejador de base de datos relacional que utilizó como lenguaje para la manipulación de datos SQL.

Una de las principales tareas de un sistema informático es almacenar y gestionar datos. Para ocuparse de esta tarea, programas computarizados conocidos como sistemas de gestión de base de datos comenzaron a surgir a finales de los sesenta y comienzo de los setenta. Un sistema de gestión de base de datos, o **DBMS** (DataBase Management system), ayudaba a los usuarios de la computadora a organizar y estructurar sus datos, y permitía al sistema informático jugar un papel más activo en la gestión de los datos. Aunque los sistemas de gestión de base de datos se desarrollaron inicialmente en grandes sistemas de maxicomputadoras, su popularidad se ha extendido a minicomputadores, computadores personales y estaciones de trabajo.

La historia del lenguaje SQL está íntimamente relacionada con el desarrollo de las bases de datos relacionales. El concepto de base de datos relacional fue desarrollado originalmente por el Dr. E. F. <<Ted>> Codd, un investigador de IBM. En Junio de 1970, El Dr. Codd publicó un artículo titulado "**Un modelo relacional de datos para grandes bancos de datos compartidos**", que esquematizaba una teoría matemática de cómo los datos podrían ser almacenados, manipulados y utilizados por una estructura tabular. Las bases de datos y SQL tiene sus orígenes en este artículo que apareció en la revista científica *Communications of the Association for Computing Machinery*.



El artículo desencadenó una racha de investigaciones en base de datos relacionales incluyendo un importante proyecto de investigación dentro de IBM. El objetivo del proyecto, llamado System/R, fue demostrar la operatividad del concepto relacional y proporcionar alguna experiencia en la implementación efectiva de un DBMS relacional. El trabajo en el System/R comenzó a mediados de los setentas en los laboratorios de Santa Teresa de IBM en San Jose, California.



En 1974 y 1975 la primera fase del proyecto/R produjo un mínimo prototipo de un DBMS relacional. Además del propio DBMS, el proyecto System/R incluía trabajos sobre lenguajes de consultas de bases de datos. Uno de estos lenguajes fue denominado SEQUEL, un acrónimo de Structured English Query Language (*Lenguaje Estructurado de Consultas en Inglés*). En 1976 y 1977 el prototipo de investigación System/R fue reescrito desde el principio. La nueva implementación soportaba consultas multitablas y permitía que varios usuarios compartieran el acceso a los datos.

La implementación de System/R fue distribuida a una serie de instalaciones de clientes de IBM para evaluación en 1978 y 1979. Estas primeras instalaciones de usuarios proporcionaron ciertas experiencias efectivas en el uso de System/R y de su lenguaje de base de datos, que había sido renombrada como SQL, o Structured Query Language (*Lenguaje Estructurado de Consultas*), por razones legales. A pesar del cambio de nombre, la pronunciación SEQUEL permaneció, y continúa hoy en día. En 1979 el proyecto System/R llegó al final, e IBM concluyó que las base de datos relacionales no solamente era factibles, sino que podrían ser la base de un producto comercial útil.

El proyecto System/R y su lenguaje de base de datos SQL recibieron buenos comentarios en las revistas técnicas durante los setenta. Seminarios en tecnología de base de datos celebraron debates sobre los méritos del nuevo y "Herético" modelo relacional. Para 1976 era evidente que IBM se estaba entusiasmando con la tecnología de la base de datos relacional, y que estaba adquiriendo un compromiso importante con el lenguaje SQL.

La publicidad referente al System/R atrajo la atención de un grupo de ingenieros en Merlo Park, California, que decidió que la investigación de IBM presagiaba un mercado comercial para la base de datos relacional. En 1977 formaron una compañía llamada Relational Software, Inc., para construir un DBMS basado en SQL. El producto, de nombre Oracle, apareció en 1979 y se convirtió en el primer DBMS relacional comercialmente disponible. Oracle se adelantó comercialmente al primer producto de IBM en dos años completos, y se ejecutaba en minicomputadoras VAX de Digital, que eran menos caras que los maxicomputadores de IBM. Hoy la empresa, renombrada Oracle Corporation, es un vendedor importante de sistemas de gestión de bases de datos relacional, con ventas anuales por más de 1,000 millones de dólares.

Profesores en los laboratorios informáticos de Berkeley de la Universidad de California estuvieron también investigando las bases de datos relacionales a mediados de los setenta. Al igual que el equipo investigador de IBM, también ellos construyeron un prototipo de un DBMS relacional, y llamaron a su sistema Ingres. El proyecto Ingres incluía un lenguaje de consulta llamado QUEL que, aunque más "estructurado" que SQL, era menos parecido al inglés. Muchos de los expertos de bases de datos actuales comenzaron a integrarse en las bases de datos relacionales a consecuencia del proyecto Ingres de Berkeley, incluyendo entre ellos a los fundadores de Britton-Lee, un suministrador de máquinas de base de datos, y Sybase, el creador del producto SQL Server de Ashton-Tate/Microsoft.

En 1980 varios profesores abandonaron Berkeley y fundaron Relational Technology Inc. para construir una versión comercial de Ingres, anunciada en 1981. Relational Technology, renombrada Ingres Corporation en 1989, sigue siendo un importante vendedor de DBMS relacional, con ventas anuales de más de 150 millones de dólares. El lenguaje de consulta original QUEL fue efectivamente sustituido por SQL en 1986, testimonio de la potencia en el mercado del estándar SQL.

Con Oracle e Ingres en la carrera por convertirse en productos comerciales, el proyecto System/R de IBM se había transformado también en un esfuerzo por construir un producto comercial, llamado SQL/Data System (SQL/DS). IBM anunció el SQL/DS en 1981, y comenzó a entregar el producto en 1982. En 1983 IBM anunció una versión de SQL/DS para VM/CMS, un sistema operativo que se utiliza frecuentemente en maxicomputadoras IBM en aplicaciones corporativas de *centros de información*.

En 1983 IBM introdujo también Database 2 (DB2), otro DBMS relacional para sus sistemas de maxicomputadoras. DB2 operaba bajo el sistema operativo MVS de IBM, el sistema operativo principal utilizado en los centros de datos de grandes maxicomputadoras. La primera versión de DB2 comenzó en 1985, y los directivos de IBM lo celebraron como pieza estratégica en la tecnología software de IBM. DB2 se ha convertido desde entonces en el principal producto de DBMS relacional de IBM, y con el peso de IBM detrás de él, el lenguaje SQL de DB2 se ha convertido de hecho en el estándar entre los lenguajes de base de datos.

Durante la primera mitad de los ochentas, los vendedores de base de datos relacionales lucharon por la aceptación comercial de sus productos. Los productos relacionales tenían varias desventajas cuando se comparaban con las arquitecturas tradicionales de base de datos. El rendimiento de las base de datos relacionales era inferior al de las bases de datos tradicionales. Excepto en los productos IBM, las base de datos relacionales provenían de pequeños vendedores <<novatos>>. Y, excepto los productos IBM, las bases de datos relacionales tendían a ejecutarse en minicomputadoras en lugar de maxicomputadoras IBM.

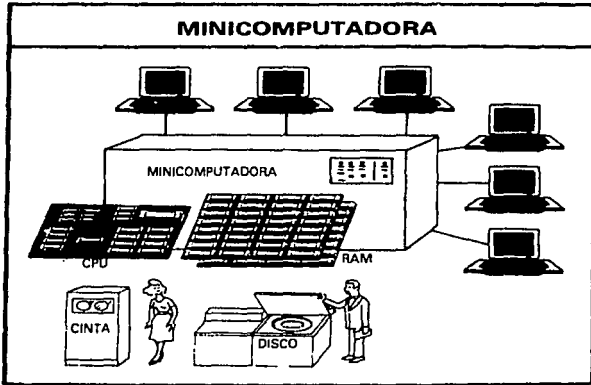
Sin embargo, los productos relacionales tenían ciertamente una ventaja importante sus lenguajes de consulta relacionales (SQL, QUEL y otros) permitía a los usuarios realizar consultas *ad hoc* a la base de datos, y obtener respuestas inmediatas, sin escribir programas. Como resultado, las bases de datos relacionales comenzaron lentamente a introducirse en las aplicaciones de centros de información como herramientas de soporte de decisiones. DB2 y SQL/DS estaban siendo lentamente aceptados, y contabilizaban sus instalaciones combinadas ligeramente por encima de las 1,000 localizaciones.

Las fuerzas del mercado también acrecentaban la popularidad de SQL a finales de los ochentas. IBM aumentaba la defensa y divulgación de SQL, posicionando DB2 como la solución de gestión de datos para los noventas. La publicación del estándar ANSI e ISO para SQL en 1986, dio estatus "oficial" a SQL como estándar. SQL también emergió como un estándar para sistemas informáticos basados en UNIX, cuya popularidad se aceleró en los ochentas. Finalmente, al hacerse las computadoras personales más potentes y conectarse en redes de área local, comenzaron a necesitar una gestión de base de datos más sofisticada. Los vendedores de base de datos para PC adoptaron SQL como la solución a sus necesidades. La tabla 2.1 muestra algunos de los sistemas de gestión de base de datos basados en SQL más populares sobre diferentes tipos de sistemas de computadores.

DMBS	EQUIPO	Sistema Operativo
DB2	Maxicomputadoras IBM	MVS
DB2	Redes de área local	OS/2
SQL/DS	Maxicomputadora IBM	VM y DOS/VSE
Rdb/VMS	Maxicomputadora VAX/VSM de Digital	
Oracle	Maxicomputadoras, minicomputadoras y PC's	
Ingres	Minicomputadoras y LAN's.	
Sybase	Minicomputadoras y LAN's	
Informix-SQL	Maxicomputadoras(Tanden), Minicomputadoras y PC's	UNIX (PC's)
Unify	Minicomputadoras	UNIX
OS/2 Extended Edition	Sistemas PS/2 de IBM	OS/2
SQL Server	PC LAN's	OS/2 y Windows NT
SQLBase	PC LAN's	DOS y OS/2
dBase IV	PC's y PC LAN's	

Tabla 2.1 Principales sistemas de gestión de base de datos basados en SQL.

Cuando la industria informática entró en los noventas, las instalaciones SQL se contaban por cientos de miles. SQL estaba claramente establecido como el lenguaje estándar de base de datos. Los vendedores de bases de datos que aún no soportaban SQL se están preocupando de hacerlo, y cualquier nuevo producto de base de datos requería a SQL como reclamo para ser tomado en serio. SQL se había convertido de hecho en el estándar oficial para las bases de datos relacionales.



Uno de los desarrollos más importantes en la aceptación del mercado de SQL es el surgimiento de los estándares SQL. Las referencias al <<estándar SQL>> significa generalmente el estándar oficial adoptado por ANSI e ISO. Sin embargo, existen otros dos estándares: el SSA (Systems Application Architecture) de IBM y el estándar X/OPEN para SQL bajo UNIX.

El trabajo en el estándar SQL oficial comenzó en 1982, cuando ANSI encargó a su comité X3H2 que definiera un lenguaje de base de datos relacional. Al principio el comité

debató los méritos de los diferentes lenguajes de base de datos propuestos. Sin embargo, cuando el compromiso de IBM con SQL se incrementó y SQL emergió como el estándar de hecho en el mercado, el comité seleccionó SQL como su lenguaje de base de datos relacional, y se aplicó a estandarizarlo.

El estándar ANSI para SQL resultante está basado en gran medida en el SQL de DB2, aunque contiene algunas diferencias importantes con respecto a él. Después de varias revisiones, el estándar fue oficialmente adoptado como estándar ANSI X3.135 en 1986, y como estándar ISO en 1987. El estándar ANSI/ISO ha sido adoptado desde entonces como estándar del Federal Information Processing Standard (FIPS) por el gobierno de los Estados Unidos.

En tabla siguiente se muestra una breve narrativa de los acontecimientos más relevantes de la historia de el primer manejador de base de datos relacional.

Fecha	Acontecimiento
1970	Codd define el modelo de base de datos relacional.
1974	Comienza el proyecto System/R de IBM.
1974	Primer artículo que describe el lenguaje SEQUEL.
1978	Test de cliente del system/R.
1979	Oracle introduce el primer RDBMS comercial.
1981	Relational Tecnology introduce Ingres.
1981	IBM anuncia SQL/DS.
1982	ANSI forma el comité de estándares SQL.
1983	IBM anuncia DB2.
1986	Se ratifica el estándar ANSI SQL.
1986	Sybase introduce RDBMS para procesamiento de transacciones.
1987	Se ratifica el estándar de ISO.
1988	Ashton-Tate y Microsoft anuncian SQL Server para OS/2.
1988	IBM anuncia la versión 2 de DB2.
1989	Primer entrega de servidores de base de datos SQL para OS/2.

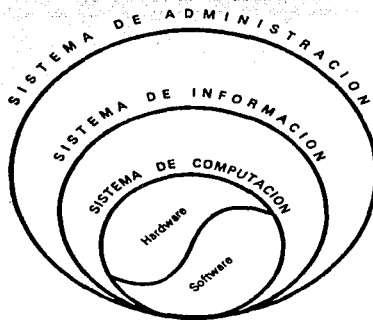
Tabla 2.2 Historia en el desarrollo de SQL.

CAPITULO III

**BASES DE DATOS**

**3.1 TIPO DE BASE DE DATOS**

Es importante conocer la evolución que han tenido los diferentes tipos de base de datos disponibles en el mercado, desde sus inicios hasta nuestros días. Es conveniente aclarar que no todos los tipos de base de datos fueron desarrollados para ser ejecutados en equipos main frame, así como para ser utilizados en cualquier tipo de aplicación o sistema de información.



**COMO SE RELACIONAN LOS SISTEMAS**

Los sistemas de gestión de base de datos organizan y estructuran los datos de tal modo que puedan ser recuperados y manipulados por usuarios y programas de aplicaciones. Las estructuras de los datos y las técnicas de acceso proporcionadas por un DBMS particular se denomina su *modelo de datos*. El modelo de datos determina la <<personalidad>> de un DBMS, y las aplicaciones para las cuales está particularmente bien conformado.

Los DBMS se clasifican de acuerdo al modelo de base de datos que administra.

### 3.1.1 Base de Datos Jerárquica

En este modelo, los registros se ligan uno a otro en una estructura de árbol, como en un organigrama. La recuperación de los datos en este tipo de base de datos requería, por tanto, navegar a través de los registros moviéndose hacia arriba, hacia abajo y hacia a los lados, para recuperar un registro a la vez.

En la figura 3.1 se muestra la estructura de una base de datos jerárquica de una lista de materiales para la construcción de un automóvil, donde cada registro de la base de datos representa una pieza específica. Los registros tenían una relaciones *padre/hijo*, que ligaba cada pieza a su subpieza, y así sucesivamente.

Uno de los sistemas de gestión de base de datos jerárquica más popular fue el Information Management System (IMS) de IBM, introducido primeramente en 1968. Las ventajas de IMS y su modelo jerárquico son las siguientes:

#### 1. ESTRUCTURAS SIMPLES

La organización de una base de datos IMS era fácil de entender. La jerarquía de la base de datos se asemejaba al diagrama de organización de una empresa o a un árbol familiar.



## 2. ORGANIZACIÓN PADRE/HIJO

Una base de datos IMS era excelente para representar relaciones padre/hijo, tales como "A es una pieza de B" o "A es propiedad de B".

## 3. RENDIMIENTO

IMS almacenaba las relaciones padre/hijo como punteros físicos de un registro de datos a otro, de modo que el movimiento a través de la base de datos era rápido. Puesto que la estructura era sencilla, IMS podía colocar los registros padre e hijo cercanos unos a otros en el disco, minimizando la entrada/salida de disco.

IMS sigue siendo el DBMS más ampliamente instalado en los maxicomputadores IBM. Se utiliza en más del 25% de las instalaciones maxicomputadoras IBM.

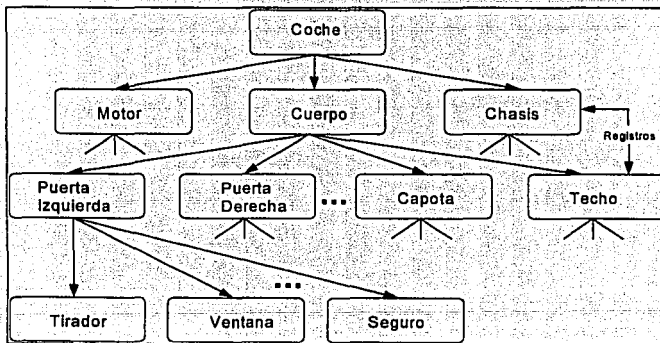


Figura 3.1 Base de datos jerárquica

### 3.1.2 Base de Datos de Red

Para manejar aplicaciones tales como el procesamiento de pedidos, se desarrolló un nuevo modelo de base de datos en red (que no tiene nada que ver con redes de área local). El modelo de base de datos en red extendía el modelo jerárquico permitiendo que un registro participara en múltiples relaciones padre/hijo, como se muestra en la figura 3.2. Estas relaciones eran conocidas como conjuntos en el modelo en red. En 1971 la conferencia sobre Lenguajes de Sistemas de Datos publicó un estándar oficial para bases de datos en red, conocido como modelo CODASYL. IBM nunca desarrolló un DBMS en red por sí mismo, eligió en su lugar extender el IMS a lo largo de los años. Pero durante los años setenta, compañías de software independientes se apresuraron a adoptar el modelo en red, creando productos tales como el IDMS de Cullinet, el Total Cincom y el DBMS Adabas que se hizo muy popular.

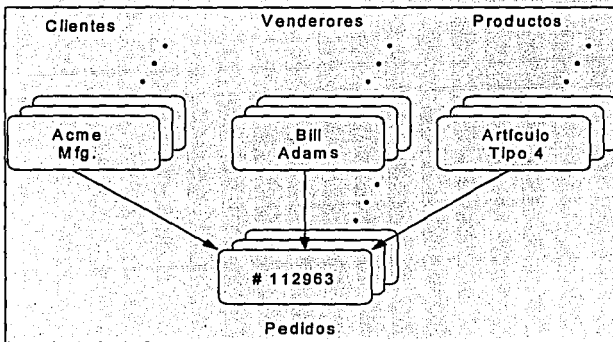


Figura 3.2 Base de datos de Red (Múltiples relaciones padre/hijo)

Para un programador, acceder a una base de datos en red era muy similar a acceder a una base de datos jerárquicos. Un programa de aplicación podía:

- ✓ Hallar un registro padre específico mediante claves (como por ejemplo un número de cliente).
- ✓ Descender al primer hijo en un conjunto particular (el primer pedido remitido por este cliente).
- ✓ Moverse lateralmente de un hijo al siguiente dentro del conjunto (el vendedor que aceptó el pedido).
- ✓ Ascender desde un hijo a su padre en otro conjunto (el vendedor que aceptó el pedido).

Una vez más el programador tenía que recorrer la base de datos registro a registro, especificando esta vez qué relación recorrer además de indicar la dirección.

Las bases de datos en red tienen varias ventajas :

### 1. FLEXIBILIDAD

Las múltiples relaciones padre/hijo permitían a una base de datos en red representar datos que no tuvieran una estructura jerárquica sencilla.

### 2. NORMALIZACIÓN

El estándar CODASYL reforzó la popularidad del modelo de red, y los vendedores de minicomputadoras tales como Digital Equipment Corporation y Data General implementaron bases de datos en red.

### 3. RENDIMIENTO

A pesar de su complejidad superior, las bases de datos en red reforzaron el rendimiento aproximándolo al de las

base de datos jerárquicos. Los conjuntos se representaron mediante punteros a registros de datos físicos, y en algunos sistemas, el administrador de la base de datos podía especificar la agrupación de datos basada en una relación de conjunto.

Las bases de datos en red tenían sus desventajas también. Igual que las bases de datos jerárquicos, resultaban muy rígidas. Las relaciones de conjunto y la estructura de los registros tenían que ser especificadas de antemano. Modificar la estructura de la base de datos requería típicamente la reconstrucción de la base de datos completa.

Tanto las bases jerárquicas como las bases en red eran herramientas para programadores. Para responder a una cuestión tal como "¿Cuál es el producto más popular ordenado por Acme Manufacturing?" un programador tenía que escribir un programa que recorriera su camino a través de la base de datos. La anotación de las peticiones para informes a medida duraba con frecuencia semanas o meses, y para el momento en que el programa estaba escrito la información que se entregaba con frecuencia ya no valía la pena.

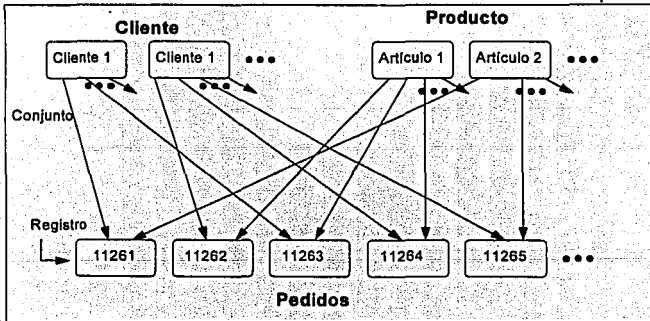


Figura 3.3 Base de datos en Red, para procesamiento de pedidos.

### 3.1.3 Base De Datos Distribuidas o Relacional

En 1970, el Dr. Codd (que trabajaba en IBM), propuso el concepto relacional. Un sistema relacional es aquel que presentan los datos a los usuarios a manera de tablas, donde no existe una relación padre-hijo. Las tablas se forman por renglones (registros) y columnas (campos de datos). Una aplicación accesa los datos a través de estructuras lógicas, llamadas vistas (views). Una vista representa registros de una tabla o varias tablas. El usuario no necesita considerar cómo se almacena los datos físicamente. Cuando se crea una vista con más de una tabla, debe existir un campo de dato común para relacionar los renglones coincidentes de las tablas participantes. Por ejemplo, en la figura 3.4 se representa una base de datos relacional, los clientes se ligan con los artículos por medio de un número de artículo.

Desgraciadamente, la definición práctica de "*¿Qué es una base de datos relacional?*" resultaba mucho menos clara que la definición matemática precisa recogida en el artículo de Codd. Los primeros sistemas de gestión de base de datos relacionales fallaron en implementar algunas partes claves del modelo del Dr. Codd, que sólo ahora están encontrando su acomodo en productos comerciales. Conforme el concepto relacional crecía en popularidad, muchas bases de datos que se llamaban a si mismas "*relacionales*" no lo eran de hecho. En respuesta a la corrupción del término "*relacional*" el Dr. Codd escribió un artículo en 1985 estableciendo doce reglas a seguir por cualquier base de datos que se llamara "*verdaderamente relacional*". Las doce reglas de Codd han sido aceptadas desde entonces como la definición de un DBMS *verdaderamente relacional*.

Los primeros productos relacionales para main frame y minis aparecieron en el mercado a fines de los setenta y principios de los ochentas. Actualmente hay más de cien productos relacionales que pueden correr sobre cualquier combinación de hardware y sistema operativos.

El concepto relacional no fue aceptado hasta que IBM saca al mercado el primer DBMS verdaderamente relacional, el DB2 y SQL/DS.

La gestión de base de datos distribuidas es una de las áreas más activas de desarrollo de DBMS hoy día, y el continuo y creciente soporte de datos distribuidos será uno de los desarrollos más importantes en la década de los noventas. Los vendedores de DBMS darán soporte de base de datos distribuidas en etapas y los

nuncios provocados por las presiones competitivas precederá típicamente a la disponibilidad efectiva de los productos en seis meses o más.

La mayoría de los vendedores de DBMS están principalmente centrados en proporcionar capacidad de base de datos distribuidas entre sus propios productos DBMS, con énfasis secundarios en pasarelas u otros entornos DBMS. Oracle, Ingres y Rdb/VMS proporcionan cada uno pasarelas a las importantes bases de datos sobre maxicomputadoras IBM. Sybase también proporciona una pasarela a DB2, y da paso más allá con su especificación Open Server, que permite a terceros construir pasarelas a otro entorno DBMS. Sin que exista ningún estándar para bases de datos distribuidas o para protocolos subyacentes, las pasarelas continuarán jugando un papel secundario en el mercado de DBMS. Las redes distribuidas que mezclan diferentes productos DBMS no pueden proporcionar el mismo tipo de transparencias e integridad de transacción ofertado por una red de un solo vendedor sin tales estándares.

Como en muchas otras áreas, las acciones de IBM podrían producir un cambio dramático en las bases de datos distribuidas. Hasta la fecha, los protocolos que IBM utiliza para comunicaciones internas entre sus productos DBMS distribuidos están cerrados y son propiedad de IBM. Sin embargo, si IBM publicase y normalizase este protocolo, los vendedores de DBMS independientes se apresurarían a soportarlos como modelo de proporcionar acceso distribuido a DB2 y SQL/DS. Los vendedores de DBMS podrían también utilizar estos protocolos IBM como un método <<neutro>> de comunicación entre sus sistemas DBMS, proporcionado, efectivamente un estándar para la implementación de una base de datos distribuida heterogéneamente. Si IBM va a adoptar tal paso dramático, o si verá una ventaja competitiva en mantener un DBMS distribuido cerrado y propietario, está por verse.

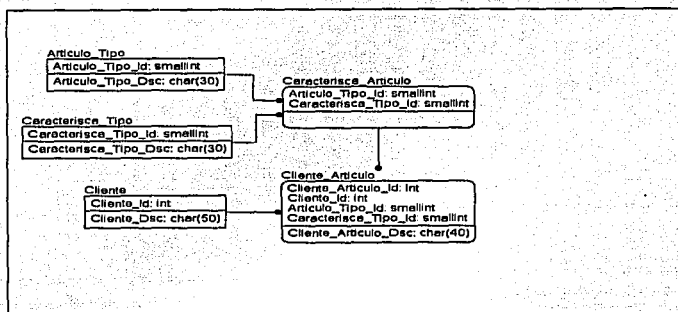


Figura 3.4 Base de datos relacional, para artículos de un almacén.

### 3.1.4 Base De Datos Orientada a Objetos

Gran parte de la investigación académica en el área de las bases de datos durante los últimos años se ha concentrado en nuevos modelos de datos <<posrelacionales>>. Al igual que el modelo relacional proporcionó claras ventajas sobre los primitivos modelos jerárquico y de red, el objetivo de esta investigación es desarrollar nuevos modelos de datos que puedan superar algunas de las desventajas del modelo relacional. Hoy en día, la mayoría de las investigaciones sobre el nuevo modelo de datos están centradas en las llamadas base de datos <<orientadas a objetos>>. Los entusiastas defensores de base de datos orientadas a objetos proclaman que son la siguiente generación de los sistemas de base de datos y predicen que será un serio reto a las base de datos relacionales hacia el final del siglo.

A diferencia del modelo de datos relacional, donde el artículo de Codd de 1970 proporcionó una clara definición matemática de una base de datos relacional, no existe acuerdo sobre qué es una base de datos orientada a objetos. Cuando los investigadores utilizan el término, están generalmente describiendo una base de datos que utiliza los mismos principios organizativos que la programación orientada a objetos.

Estos principios incluyen :

### 1. OBJETOS

En una base de datos orientada a objetos cualquier cosa es un objeto y se manipula como un objeto. La organización tabular de filas/columnas de una base de datos relacional es sustituida por la noción de colecciones de objetos. En general, una colección de objetos es ella misma un objeto y puede ser manipulada del mismo modo que se manipulan los restantes objetos.

### 2. CLASES

Las bases de datos orientadas a objetos sustituyen la noción relacional de tipo de datos o valores atómicos, con una noción jerárquica de clases y subclases. Por ejemplo, VEHÍCULOS podría ser una clase de objeto, y los miembros individuales (<<instancias>>) de esa clase incluirían un coche, una bicicleta, un tren o un bote. La clase VEHÍCULOS podría incluir subclases denominadas COCHES y BOTES, respetando una forma más especializada de vehículo. Análogamente, la clase de COCHES podría incluir una subclase denominada CONVERTIBLE, etc.

### 3. HERENCIA

Los objetos heredan las características de su clase y de todas las clases de niveles superiores a la que pertenecen. Por ejemplo, una de las características de un vehículo podría ser el <<número de pasajeros>>. Todos los



miembros de las clases COCHES, BOTES y CONVERTIBLES tendrían también este atributo, ya que son subclases de VEHÍCULOS. La clase COCHES podría también tener el atributo << número de puertas >>, y la clase CONVERTIBLES heredaría este atributo. Sin embargo, la clase BOTES no heredaría el atributo.

#### 4. MENSAJE Y MÉTODOS

Los objetos se comunican uno con otros mediante el envío y recepción de mensajes. Cuando recibe un mensaje, un objeto responde ejecutando un método, un programa almacenado dentro del objeto que determina cómo se procesa el mensaje. Por tanto un objeto incluye un conjunto de comportamientos descritos por sus métodos. Generalmente un objeto comparte mucho de los mismos métodos con otros objetos de su clase.

Estos principios y técnicas hacen que las bases de datos orientadas a objetos estén bien adecuadas a aplicaciones que implican tipos de datos complejos, tales como documentos compuestos o de diseños asistidos por computadora que combinan texto, gráficos y hoja de cálculo. La base de datos proporciona un modo natural de representar las jerarquías que aparecen en los datos complejos.

Por ejemplo, un documento entero puede presentarse como un único objeto, compuesto de objetos más pequeños (secciones), compuestos de objetos aún más pequeños (párrafos, gráficos, etc.). La jerarquía de clases permite a la base de datos seguir la pista del <<tipo>> de cada objeto en el documento (párrafos, gráficos, ilustraciones, títulos, pies de página, etc.). Finalmente, el mecanismo de mensaje ofrece soporte natural para una interfaz de usuario gráfica. El programa de aplicación puede enviar un mensaje <<extráigalo usted mismo>> a cada parte del documento, pidiendo

que se extraiga de la pantalla. Si el usuario cambia la forma de la ventana que visualiza el documento, el programa de aplicación puede responder enviando un mensaje <<modifica tu tamaño>> a cada parte del documento, etc. Cada objeto del documento tiene la responsabilidad de su propia visualización, por lo que los nuevos objetos pueden ser añadidos fácilmente a la arquitectura del documento.

Existen varias bases de datos comerciales orientadas a objetos actualmente bajo desarrollo, pero sólo un puñado de ellas están disponibles hoy en día como productos comerciales. Sin embargo, las bases de datos orientadas a objetos ya han provocado una tormenta de controversias en la comunidad de las bases de datos. Los proponentes proclaman que las bases de datos orientadas a objetos reemplazarán en un futuro al modelo relacional. Los criterios arguyen que las bases de datos orientadas a objetos son simplemente una reformulación de las base de datos jerárquicas de los sesentas, con una nueva terminología y un nuevo giro en conceptos antiguos. Los vendedores de base de datos relacionales proclaman que la mayoría, si no todos, los beneficios de las bases de datos orientadas a objetos pueden lograrse añadiendo soporte de "blobs" a una base de datos relacional.

Las bases de datos orientadas a objetos jugarán probablemente un papel creciente en segmentos de mercados especializados, tales como diseño teórico, procesamiento de documentos compuestos e interfaces de usuario gráficas. Sin embargo, es poco probable que puedan tener un impacto importante en la corriente principal de aplicaciones de procesamiento de datos durante los próximos años. Con el soporte de un estándar oficial, el apoyo al mercado de IBM y el soporte de varios vendedores DBMS independientes importantes, el vagón de SQL es simplemente demasiado potente para que descarrile debido a un puñado de vendedores de DBMS orientados a objetos en cualquier momento en un futuro cercano. Sin embargo, hace tan sólo una década, la misma observación podría haberse efectuado referente a las bases de datos relacionales y a SQL. De todas las tendencias del mercado de las bases de datos, el desarrollo de bases de datos orientadas a objetos es la única que amenaza el dominio continuo de SQL en las próximas décadas.

Se continuará con una reseña de la teoría del Dr. Codd, que no es otra cosa más que las características que debe cubrir un manejador de base datos relacional.

### 3.2 TEORÍA DEL DR. CODD

En su artículo de 1985 en "*Computerworld*", Ted Codd presentó doce reglas que una base de datos debe obedecer para que sea considerada verdaderamente relacional. Las reglas se derivan del trabajo teórico de Codd sobre el modelo relacional y representan realmente más un objetivo ideal que una definición de una base de datos relacional.

Las doce reglas de Dr. Codd son:

#### 1. LA REGLA DE INFORMACIÓN.

Toda la información de una base de datos relacional está representada explícitamente a nivel lógico y exactamente de un modo, mediante valores de tablas. Es básicamente la definición informal de una base de datos relacional.

#### 2. REGLAS DE ACCESO GARANTIZADO

Todos y cada uno de los datos (valores atómicos) de una base de datos relacional se garantiza que sea lógicamente accesibles recurriendo a una combinación de nombre de tabla, valor de clave primaria y nombre de la columna. Esta regla refuerza la importancia de las llaves primarias para localizar datos en la base de datos.

**3. TRATAMIENTO SISTEMÁTICO DE VALORES NULOS** Los valores nulos (distintos de la cadena de caracteres vacíos o de una cadena de caracteres en blanco y distinta del cero o de cualquier otro número) se soportan en los DBMS completamente relacionales, representan la falta de información y la información inaplicable de un modo sistemático e independiente del tipo de datos. Requieren soportar una instrucción que indique la falta de un dato.

**4. CATÁLOGO EN LÍNEA DINÁMICO BASADO EN EL MODELO RELACIONAL.** La descripción de la base de datos se representa a nivel lógico del mismo modo que los datos ordinarios, de modo que los usuarios autorizados puedan aplicar a su interrogación el mismo lenguaje relacional que aplican a los datos regulares. La base de datos debe contener ciertas tablas de sistema cuyas columnas describan la estructura de la propia base de datos.

**5. REGLAS DE SUBLENGUAJE COMPLETO DE DATOS.** Un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal (por ejemplo, el modo de rellenar con blancos). Sin embargo, debe haber al menos un lenguaje cuyas sentencias sean expresables, mediante algunas sintaxis bien definida, como cadenas de caracteres y que sea completa en cuanto al soporte de todos los puntos siguientes:

1. Definición de datos.
2. Definición de vistas.

3. Manipulación de datos (interactiva y por programa).
4. Restricciones de integridad.
5. Autorización.
6. Fronteras de transacciones (comienzo, cumplimiento y vuelta atrás).

Utilizar un lenguaje de base de datos relacional, tal como SQL, aunque no se requiera específicamente SQL. El lenguaje debe ser capaz de soportar todas las funciones básicas de un DBMS, como son : creación de una base de datos, recuperación y entrada de datos, implementación de la seguridad de la base de datos, etc.

**6. REGLA DE ACTUALIZACIÓN DE VISTAS** Todas las vista que sean teóricas actualizables son también actualizables por el sistema. Las vistas son tablas virtuales utilizadas para dar a diferentes usuarios de una base de datos diferentes vistas de su estructura. Es una de las reglas más difíciles de implementar en la práctica, y ningún producto comercial la satisface totalmente hoy en día.

**7. INSERCIÓN, ACTUALIZACIÓN Y SUPRESIÓN DE ALTO NIVEL.** La capacidad de manejar una relación de base de datos o una relación derivada como un único operando se aplica no solamente a la recuperación de datos, sino también a la inserción, actualización y supresión de los datos.

La naturaleza orientada a conjuntos de una base de datos relacional, requiere que las filas sean tratadas como un conjunto en operaciones de inserción, supresión y actualización. La regla está diseñada para prohibir implementaciones que sólo sopórtan la modificación o recorrido fila a fila de la base de datos.

**8. INDEPENDENCIA FÍSICA DE LOS DATOS** Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cualquiera que sean los cambios efectuados ya sea a las representaciones de almacenamiento o a los métodos de acceso.

**9. INDEPENDENCIA LÓGICA DE LOS DATOS.** Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cuando se efectúen sobre las tablas de base de cambios preservadores de la información de cualquier tipo que teóricamente permita alteraciones. Las reglas 8 y 9 aíslan al usuario o al programa de aplicaciones de la implementación de bajo nivel de la base de datos. Especifican las técnicas de acceso y almacenamiento utilizadas por el DBMS, e incluso los cambios a la estructura de las tablas en la base de datos, no deberían afectar a la capacidad del usuario de trabajar con los datos.

**10. INDEPENDENCIA DE INTEGRIDAD.** Las restricciones de integridad específica para una base de datos relacional particular deben ser definibles en el sublenguaje de datos relacional y almacenables en el catálogo, no en los programas de aplicación. El lenguaje de base de datos debería soportar las restricciones de integridad que restringen los datos que puede ser introducidos en la base de datos y las modificaciones que puedan ser efectuadas en ésta. Esta es otra de las reglas que no soportan la mayoría de los productos comerciales DBMS.

**11. INDEPENDENCIA DE DISTRIBUCIÓN.** Un DBMS relacional tiene independencia de distribución. El lenguaje de base de datos debe ser capaz de manipular datos distribuidos localizados en otros sistemas informáticos.

**12. REGLA DE NO SUBVERSIÓN.** Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro cada vez), ese no puede ser utilizado para subvertir o suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez). Esta regla impide "otros caminos", en la base de datos que pudiera subvertir su estructura relacional y su integridad.

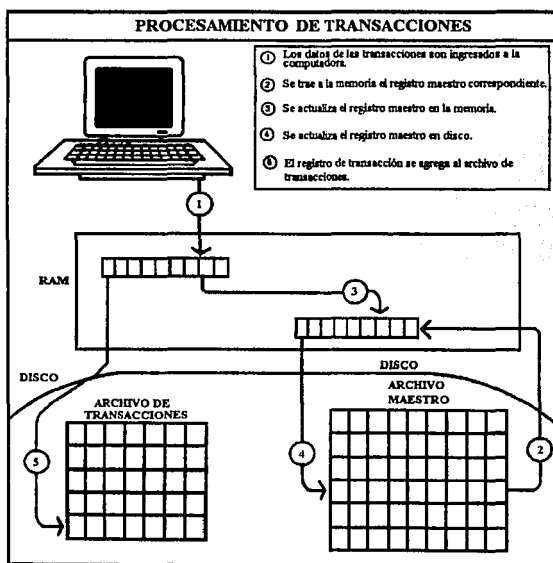


Por último se incluye las características del Manejador de Base de Datos Relacional DB2.



## 3.3 DB2 (Data Base Dos)

DB2 es un manejador de base de datos relacional (DBMSR) ofrecido por IBM para ambientes MVS. DB2 soporta acceso a datos desde CICS (Monitor de teleprocesos), IMS/DC (base de datos jerárquica) y TSO (Medio interactivo que pueden ser compartido por múltiples usuarios y está orientado a la submisión y control de procesos), "en línea o batch" simultáneamente. Tales accesos pueden ser logrados también mediante aplicaciones escritas en COBOL, PL/I, ENSAMBLADOR o FORTRAN, a través de un lenguaje de interface con el usuario de DB2.



Una de las recomendaciones que hace IBM a los usuarios que desean instalar DB2 en su ambiente de trabajo es, que se realice una afinación al sistema operativo así como a los sistemas de interfaces para manejar las comunicaciones (CICS, TSO, IMS/DC), antes de instalar DB2, de lo contrario se tendrán problemas serios de desempeño de la máquina.

La teoría relacional basa su significado en la distinción entre las vistas de datos de los usuarios y el almacenamiento físico de las mismas. Mientras que la mayoría de los manejadores no relacionales requieren de cierto conocimiento y manipulación de datos basados en conceptos de almacenamiento físico, la teoría relacional presenta los datos para su manejo en forma de tabla sin limitaciones respecto al almacenamiento.

DB2 se divide en 2 grandes categorías :

1. Servicio de la base de datos.                      Básicamente maneja los datos del usuario, esencialmente a través de SQL.
2. Servicio del sistema.                                Se usan más funciones internas tales como órdenes de DB2, estadísticas, almacenamiento virtual, procesos de recuperación y enlaces así como funciones de inicio y terminación.

Como hemos dicho DB2 presenta los datos en forma de tablas. El manejo, definición, manipulación, acceso y control de datos son manejados a través de SQL formado por 3 lenguajes :

1. Lenguaje de definición de datos (DDL)
2. Lenguaje de manipulación de datos (DML)
3. Lenguaje de control de datos (DCL)

El primero de ellos (DDL) se refiere a las instrucciones utilizadas para crear elementos de DB2; el segundo de ellos (DML), como su nombre lo indica, es útil para manejar los datos, es un lenguaje de formato libre, que puede ejecutarse tanto de forma interactiva como de modo Batch. Un poderoso elemento de SQL, es DML el puede operar bloques completos de registros (*filas*) al mismo tiempo, de diferentes formas a la mayoría de los lenguajes de alto nivel. Finalmente, el lenguaje de control (DCL) de datos explícitamente otorga privilegios a usuarios potenciales de elementos de DB2 a través de 2 órdenes principales, GRANT Y REVOKE.

Existen muchas discrepancias que necesitan ser rectificadas entre el modelo relacional de datos de Codd y su representación por DB2. De acuerdo con el Dr. Codd, el DB2 de IBM necesita mejorar su soporte de dominios y capacidades para actualizar, sin embargo sigue superando cualquier producto en el mercado en sus aplicaciones del modelo relacional.

SQL se apega estrictamente al manejo de la base de datos sin tomar en cuenta los elementos de control de programas. Consecuentemente, su mayor impacto es que libera a los lenguajes de desarrollo de aplicaciones de la carga que representa la estructura de la base de datos.

La utilería de manejo de datos (QMF) ayuda a los usuarios finales por medio de dos métodos alternativos de SQL a realizar reportes que van desde reportes de formato libre hasta presentaciones tan complicadas, como lo sean los requerimientos del usuario.

El almacenamiento de las definiciones y las autorizaciones sobre los elementos de DB2 están contenidos en un conjunto de tablas especiales de DB2 llamadas colectivamente catálogos de DB2; el cual es controlado por el mismo DB2 y contiene información que puede ser accedida por cualquier usuario autorizado como si fuera cualquier tabla de DB2.

Una base de datos está compuesta por una o varias tablas y cada una de ellas está asociada a dos tipos de tablas, una tabla que contiene los índices de los registros para hacer los accesos a la base de datos y otra que contiene los registros con la información. Ambas están estrechamente interrelacionadas.

Como es sabido todos los DBMSR presentan ventajas y desventajas con respecto a otros productos en el mercado. A continuación se describirán algunas de las ventajas y desventajas más importantes de DB2.

## VENTAJAS

### A) SINÓNIMOS (Create/Drop SYNONYM)

El administrador proporciona a otros usuarios permiso para acceder a las tablas, sin embargo, deben utilizarse nombres de tablas calificados para referirse a las tablas de otros usuarios. En la práctica, esto significa que "todas" las consultas que referencian las principales tablas de un sistema deben utilizarse nombres de tablas cualificados, lo cual hace que las consultas sean largas y tediosas de escribir.

Para tratar este problema, los productos SQL de IBM proporcionan una capacidad de SINÓNIMOS. Un sinónimo es un nombre definido para representar el nombre de alguna tabla. Un sinónimo se crea utilizando la sentencia CREATE SYNONYM. Una vez que se ha definido un sinónimo, puede utilizarse lo mismo que un nombre de tabla en las consultas SQL.

El uso de sinónimos no cambia el significado de la consulta y sigue siendo necesario tener permiso para acceder a la tabla de los demás usuarios. No obstante, los sinónimos simplifican las sentencias SQL utilizadas y aparentan que las tablas son propiedad del usuario. Si posteriormente se decide que ya no se desea utilizar los sinónimos puede suprimirse con la sentencia DROP SYNONYM.

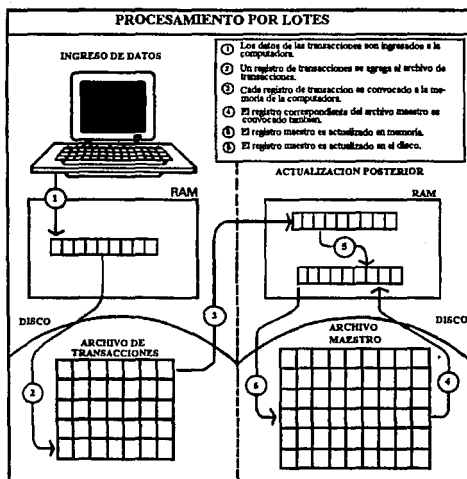
### B) MECANISMOS DE CONTROL Y/O HERRAMIENTAS DE CONTROL.

Como mecanismo de control, la estructura de catálogo del DB2 actual es capaz de soportar alrededor del 95 % de los requerimientos de control de una instalación. Las instrucciones LABEL Y COMMENT ON, en conjunto con las tablas que manejan la información, a través de mecanismo de control (PLAN).

La finalidad de que una aplicación esté asociada con un PLAN es hacer el acceso a la base de datos en forma DINÁMICA de lo contrario, el acceso se realizará en forma ESTÁTICA, lo cual hace el acceso más lento.

### C) AMBIENTE DE APLICACIONES

DB2 existe como un subsistema en el ambiente de MAS que tiene otros tres subsistemas MIS, CISCA y TOS. DB2 está conectada a estos subsistemas por sus respectivas facilidades que los unen. La facilidad de unión con TOS es requerida para ejecutar aplicaciones TOS o BATCH (Procesos que son enviados a ejecución y que no requieren liberar los recursos para que puedan ser utilizados por aplicaciones o programas de otros usuarios) que accesen datos de DB2, para utilizar las pantallas interactivas provistas por DB2 o para ejecutar los programas desarrollados.



## D) SEGURIDAD

El DB2 emplea tres mecanismos principales de seguridad para controlar la autorización e integridad del acceso a la información de una aplicación dada: un subsistema DB2 de AUTORIZACIÓN PRIVILEGIADA, un sofisticado MECANISMO DE CIERRE Y TRAYECTORIA DE TRANSACCIONES, y una característica menos tangible en las vistas.

Los privilegios son diferidos a un usuario por medio del comando GRANT y para desactivar los privilegios de un usuario tan sólo hay que ejecutar el comando REVOKE.

## DESVENTAJAS

### A) FALTA DE UN DICCIONARIO DE DATOS

Para los usuarios habituados a las estructuras convencionales de base de datos, el elemento más obvio que le falta al DB2 es la capacidad del diccionario de datos. Si el usuario examina la funcionalidad del DB2, terminaría prefiriendo DB2 por encima de cualquier DBMSR. Los diccionarios de datos son empleados como y/o una herramienta de desarrollo.

### B) EXCESO DE INPUT/OUTPUT

Uno de los aspectos de DB2 debe cuidar para versiones posteriores es el exceso de I/O que consume para mantener la integridad referencial de la base de datos. Ejemplo, para relaciones de un empleado con el departamento (Uno a muchos), la versión 2 de BD2 es adecuada. Sin embargo en una relación de proyectos hacia empleados (muchos a muchos) requiere de una tabla de conexión con el hardware actual, esto daría como resultado un I/O desmesurado.

## FUTURO DE DB2

En este punto, no hay duda de que el DB2 es un producto estratégico de IBM. su futuro tendrá ramificación de largo alcance en todos los niveles de usuarios. El DB2 es una gran promesa para la promoción del modelo relacional de Codd, un modelo que puede crecer para establecer nuevos tipos de datos y curso accesible de selección de criterios. No obstante, el DB2 es un producto muy joven con experiencia en evolución de diseño y realización.

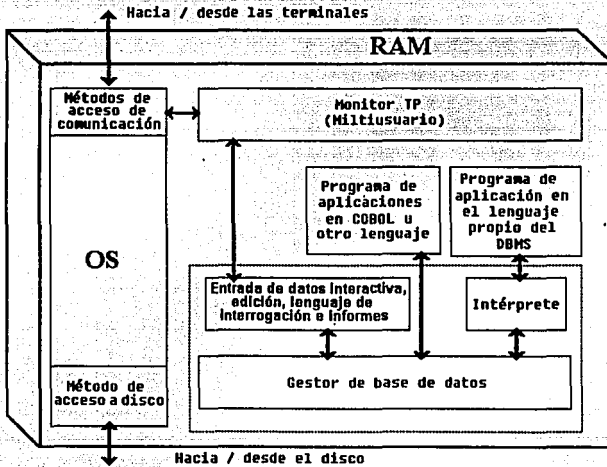
El área que más concierne y preocupa a los usuarios respecto al DB2 es el exceso de INPUT/OUTPUT (I/O) necesario para sostener la integridad referencial.

A principios de 1992 IBM lanzó al mercado DB2 para PC el cual trabaja bajo sistema operativo OS/2, éste será un gran competidor de base de datos no sólo para ambientes main frame si no en muy poco tiempo en ambientes de redes.

CAPITULO IV

MANEJADOR DE BASE DE DATOS

Como se mencionó en la introducción, SQL es lenguaje nativo que utiliza DB2; por lo que se de vital importancia conocer las características primordiales el mismo, así como la sintaxis de las instrucciones más comunes en el manejador de base de datos relacional SQL.





SQL "*Structured Query Language*" no es propiamente un lenguaje de programación, más bien es sublenguaje de base de datos que consiste en treinta sentencias especializadas para organizar gestionar y recuperar datos almacenados en una base de datos relacional. Estas sentencias SQL las incorpora a otros lenguajes como **COBOL**, **FORTRAN**, **ENSAMBLADOR** o **C** para extender a estos últimos y permitirles utilizar el acceso a la base de datos. SQL puede ejecutarse en sistemas informáticos que van desde computadoras personales hasta maxicomputadoras.

Un sistema de gestión de base de datos o DBMS (*Database Management System*), ayuda a los usuarios de la computadora a organizar y estructurar sus datos y permite al sistema informático jugar un papel más activo en la gestión de los datos.

El DBMS procesa la petición SQL, recupera los datos solicitados y los devuelve. Este proceso de solicitar datos de la base de datos y de recibir los resultados se denomina consulta *Query* a la base de datos; de aquí el nombre *Structured Query Language*.

#### 4.1 CARACTERÍSTICAS Y BENEFICIOS DE SQL

SQL es a la vez un lenguaje fácil de entender y una herramienta completa para gestionar datos. Algunas de las principales características de SQL que han logrado tener éxito en el mercado son :

1. Independencia de los vendedores.
2. Portabilidad a través de sistemas informáticos.
3. Estándares SQL.
4. Apoyo de IBM.
5. Fundamento relacional.
6. Estructura de alto nivel semejante al inglés.
7. La consulta interactivas ad hoc.
8. Acceso a la base de datos mediante programas.
9. Vistas múltiples de datos.
10. Lenguaje de base de datos.
11. Definición dinámica de datos.
12. Arquitectura cliente/servidor.

Estas son las razones por las que SQL ha emergido como la herramienta estándar para gestionar datos en computadores personales, microcomputadoras y maxicomputadores.

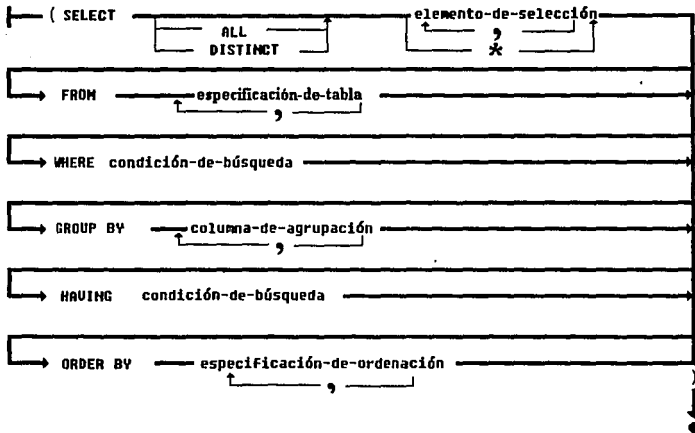
SQL fue inventado originalmente por investigadores de IBM y desde entonces se ha convertido en un producto estratégico para IBM. SQL es un componente esencial de la Systems Application Architecture SAA, la marca de IBM para la compatibilidad de sus diversas líneas de productos. El soporte de SQL está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

SQL es un lenguaje para base de datos relacional y se ha popularizado conjuntamente con el modelo de base de datos relacional. La estructura tabular, de filas y columnas de una base de datos relacional, es intuitiva para los usuarios, y hace que el lenguaje SQL se

mantenga simple y fácil de entender. El modelo relacional tiene también un fuerte fundamento teórico que ha guiado la evolución y la implementación de las bases de datos relacionales. SQL se ha convertido en manejador de base datos para las base de datos relacionales y sus tareas principales son :

1. SELECT

SQL se utiliza para recuperar datos de la base de datos, utilizando la sentencia SELECT. Se pueden recuperar todos o parte de los datos almacenados, ordenarlos y solicitar a SQL que sumarice los datos utilizados.



La sentencia **SELECT** consta de seis cláusulas, las cláusulas **select** y **from** de la sentencia son necesarias. Las cuatro cláusulas restantes son opcionales

La cláusula **SELECT** lista los datos a recuperar por la sentencia **SELECT**. Los campos pueden ser columnas de la base de datos o columnas a calcular por **SQL** cuando efectúe la consulta.

**FROM** lista las tablas que contiene los datos a recuperar por la consulta.

**WHERE** dice a **SQL** que incluya solo ciertas filas de datos en los resultados de la consulta. Se utiliza una *condición de búsqueda* para especificar las filas deseadas.

**GROUP BY** especifica una consulta sumaria, en vez de producir una fila de resultados por cada fila de datos de la base de datos, una consulta sumaria agrupa todas las filas similares y luego produce una fila sumaria de resultados para cada grupo.

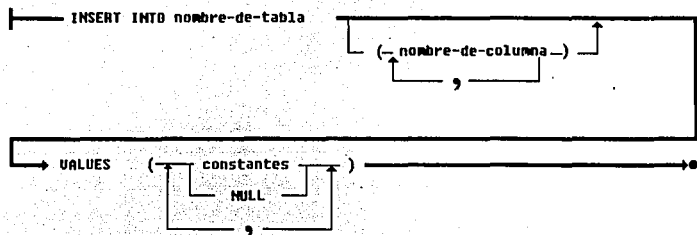
**HAVING** dice a **SQL** que incluya sólo ciertos grupos producidos por la cláusula **GROUP BY** en los resultados de la consulta. Al igual que la cláusula **WHERE**, utiliza una *condición de búsqueda* para especificar los grupos deseados.

**ORDER BY**, ordena los resultados de la consulta de una o más columnas de la base de datos. Si se omite, los resultados de la consulta no se muestran ordenados.

## 2. INSERT, DELETE Y UPDATE

**SQL** se utiliza para actualizar la base de datos, añadiendo nuevos datos con la sentencia **INSERT**, suprimiendo datos con la sentencia **DELETE** y modificando datos existentes con la sentencia **UPDATE**.

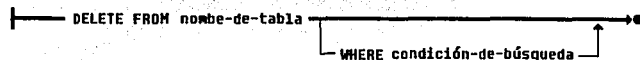
La sentencia **INSERT** de una fila, añade una nueva fila a una tabla.



La cláusula **INTO** especifica la tabla que recibirá la nueva fila y la cláusula

**VALUES** especifica los valores de datos que la nueva fila contendrá. La lista de columnas indica qué valor va a qué columna de la nueva fila.

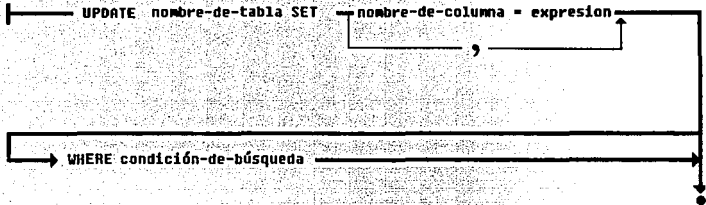
La sentencia **DELETE**, elimina filas seleccionadas de datos de una única tabla.



**FROM** especifica la tabla destino que contiene la o las filas a eliminar

**WHERE** especifica la *condición de búsqueda* de las filas de la tabla van a ser suprimidas. La sentencia **WHERE** es opcional, pero casi siempre está presente. Si se omite la cláusula **WHERE** de una sentencia **DELETE**, se borran *todos* los registros de la tabla destino.

La sentencia UPDATE, modifica los valores de una o más columnas en las filas seleccionadas de una tabla única.



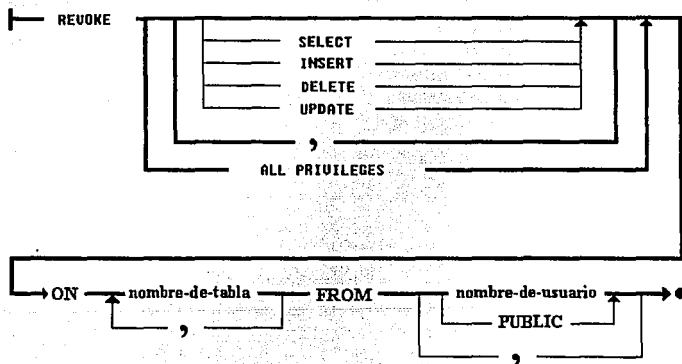
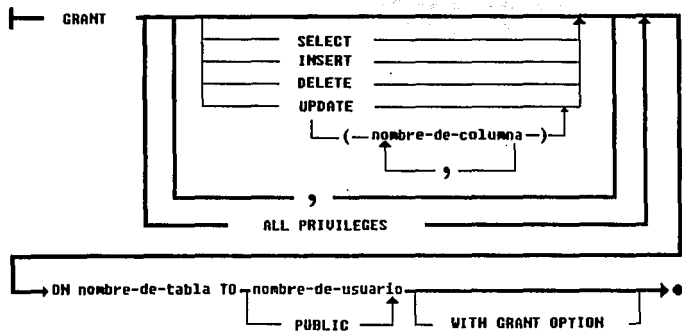
La tabla destino a actualizarse indica en la sentencia, y es necesario disponer de permiso para actualizar la tabla además de cada una de las columnas individuales que serán modificadas.

**WHERE** selecciona las filas de la tabla a modificar.

**SET** especifica qué columnas se van actualizar y calcula los nuevos valores.

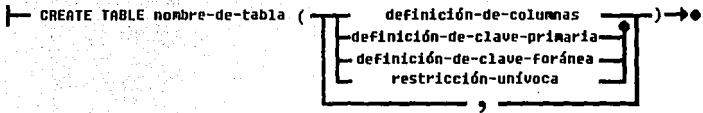
### 3. GRANT Y REVOKE

Se utiliza para controlar el acceso a la base de datos concediendo y revocando privilegios a usuarios específicos.

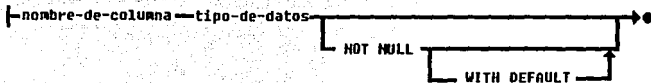


4. CREATE TABLE Y DROP TABLE

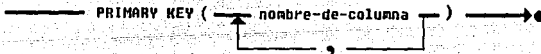
Se utiliza para crear la base de datos definiendo la estructura de nuevas tablas y borrando tablas cuando ya no se van a necesitar.



definición-de-columna:

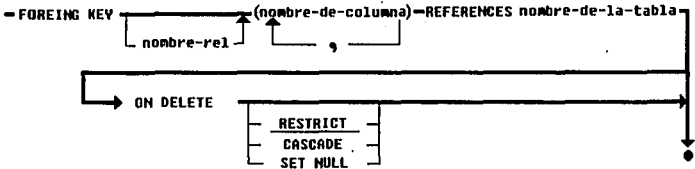


definición-de-clave-primaria:

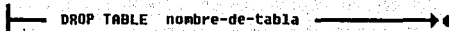
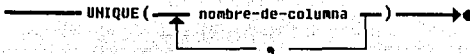




definición-de-clave-foránea:



restricción-unívoca:



Las bases de datos relacionales organizan los datos en formas tabulares sencillas y proporcionan muchas ventajas sobre los tipos anteriores de base de datos. SQL es específicamente un lenguaje relacional de base de datos utilizado para trabajar con base de datos relacionales.

## 4.2 FUTURO DE SQL

SQL es una de las tecnologías más importantes que conducen y conforman el mercado informático hoy día. Desde sus primeras implementaciones comerciales hace poco más de una década, SQL ha crecido hasta convertirse en el lenguaje estándar de base de datos. Con el respaldo de IBM la bendición de las corporaciones de estándares y el soporte entusiasta de los vendedores de DBMS, la evidencia muestra claramente la importancia de SQL :

1. Las bases de datos insignia de IBM, en sistemas de procesadores maxicomputadores hasta computadores personales, están todas en SQL.
2. Los principales vendedores de DBMS independientes y la mayoría de los fabricantes de hardware de computadoras ofrecen o planean ofrecer productos basados en SQL.
3. Las bases de datos basadas en SQL están comenzando a dominar virtualmente todos los segmentos de mercado, desde las computadoras personales a los sistemas OLPT.
4. El más importante vendedor de bases de datos para PC, Ashton-Tate, se ha inclinado ante la presión del mercado y ha injertado SQL en su producto dBASE IV.

Al comienzo de la década de los noventas, el tamaño del mercado para los productos de gestión de bases de datos traspasó la marca de los tres millones de dólares, y las predicciones dicen que continuará este crecimiento a lo largo de la década. La normalización de SQL y del modelo de base de datos relacional creó un nuevo mercado de base de datos más unificado.

Uno de los contribuidores más importantes a la elevación de SQL durante los ochentas fue un incremento dramático en el rendimiento de hardware, utilizando como estructuras de almacenamiento información de las bases de datos relacionales. Parte de este incremento de rendimiento fue debido a avances en la tecnología de las bases de datos y la optimización de las consultas.

CAPITULO V

**LENGUAJE DE CUARTA GENERACIÓN**

**5.1 INTRODUCCIÓN AL LENGUAJE DE CUARTA GENERACIÓN**

Los lenguajes de cuarta generación representan una clase de software poderoso, muy diverso y variable en cuanto a sus áreas de aplicación, capacidades, estructuras y cualidades, pero sus finalidades convergen hacia una meta común: hacer eficiente el proceso de desarrollo y mantenimiento de sistemas de información *a menudo de naturaleza comercial*, en términos de productividad y eficiencia, así como extender la utilización directa de las computadoras hacia el usuario final, al grado de que se encuentre en posibilidad de satisfacer, de manera independiente sus requerimientos de información.

Un lenguaje de cuarta generación, no representa necesariamente la idea convencional de un lenguaje de programación, es decir, un lenguaje basado en estructuras y reglas sintácticas de cierta complejidad para el usuario (programador), como **COBOL**, **PL/I** o **FORTRAN**. Ciertamente existen componentes de un lenguaje de cuarta generación que obedecen a estas características, sin embargo lo que entendemos hoy por un lenguaje de cuarta generación, bien puede ser un lenguaje estilo natural, lenguaje gráfico o un lenguaje interactivo entre una pantalla y un usuario final donde construye sus aplicaciones llenando formas, seleccionando opciones, modificando especificaciones predefinidas, entre otras cosas, orientado a qué se desea obtener y no al cómo obtenerlo.

Así un lenguaje de cuarta generación, no representa ya la idea de escribir con gran esfuerzo una enorme cantidad de líneas de código complejo, para obtener aplicaciones de limitado alcance y utilidad, sino complejos sistemas de software creados para distintas finalidades, que proveen tanto a un especialista en computación como a un usuario final, de una herramienta dinámica, ágil y práctica para la solución de una amplia variedad de problemas en el ámbito de la información.

## 5.2 CSP (Cross System Product)

Anteriormente los programadores para desarrollar un programa on-line en una maxicomputadora, utilizaba herramientas como CICS donde el usuario codificaba sus pantallas, instrucción por instrucción y una vez que ésta había sido terminada, se compilaba dejando el programa objeto en lenguaje ensamblador, para poder ser utilizada por un lenguaje de programación como PL/I, COBOL, etc.

CSP, es una herramienta de programación on-line que ha venido desplazando a CICS, siendo ésta, una herramienta que proporciona el poderío y la flexibilidad de un lenguaje de cuarta generación.

La codificación de un programa en CSP, puede realizarse desde una red de área local hasta en equipos main frames. Una vez concluida la programación de la aplicación, ésta puede ser ejecutada en el mismo ambiente o migrala a otro ambiente y ejecutarla en éste. Esta flexibilidad sólo es ofrecida por la versión 3.4.\*

Esto sin duda ha dado una gran facilidad a los programadores de aplicaciones en main frame de realizar la codificación en equipos pequeños y baratos, para posteriormente migrar sus aplicaciones a equipos grandes y caros. Esto sin duda ha traído una disminución del costo de producción de los sistemas, además proporciona a los programadores las ventajas del tiempo de respuesta de los equipos pequeños.

### **CSP, internamente realiza dos grandes procesos :**

El Cross System Product/Application Development (CSP/AD), utilizado por los programadores de aplicaciones para definir, probar y generar aplicaciones.

El Cross System Product/Application Execution (CSP/AE), que permite la ejecución de las aplicaciones.

---

\* Mucha gente experimentada en el manejo de esta herramienta, asegura que aún no pueden ejecutarse aplicaciones en equipos que no sean main frames.

**Las funciones del CSP/AD son :**

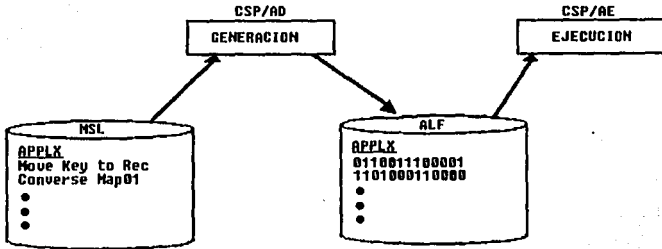
**1. Desarrollar las aplicaciones**

Una aplicación de CSP, está formada por <<n>> procedimientos, los cuales son almacenados en una biblioteca fuente, llamada Member Specification Library (MSL). Para desarrollar una aplicación, es utilizado un editor inteligente, el cual verifica la sintaxis de las instrucciones. Una MSL tiene las propiedades de un archivo VSAM.

**2. Probar las aplicaciones**

Las aplicaciones pueden ser probadas desde el ambiente fuente, es decir, sin tener que catalogarlas con anterioridad (generadas) ; incluyendo una opción de rastreo (TEST), donde el usuario puede elegir, los procesos que desea ejecutar, instrucción por instrucción, o ejecutar todos los procedimientos que forman la aplicación.

Es necesario generar las aplicaciones (compilar), para poder utilizarlas en producción. Las versiones de CSP anteriores a la 3.4, no dejaban programas ejecutables, lo que sin duda representaba una de las más grandes desventajas de este producto.



Las funciones del CSP/AE son :

1. Generación de Aplicaciones

Cargar a memoria las aplicaciones generadas en la ALF (biblioteca donde deja el código de los programas generados) y controlar su ejecución, puesto que el CSP/AE.

2. Debug

Controlar la ejecución del CSP/AD en el ambiente de desarrollo.

### 5.2.1 Concatenación de Bibliotecas

La definición de las aplicaciones se hace en una MSL de Lectura/Escritura, la cual no puede ser accesada para escritura por otros programadores al mismo tiempo. Se puede acceder hasta 5 MSL's sólo de lectura, lo cual permite compartir los componentes de las aplicaciones.

Mientras que se concatenen las MSL's durante la definición, existe un diccionario común a todas ellas, que no permite la duplicidad de nombres con características diferentes.

El acceso independiente de una MSL, abre la posibilidad de definición de nombres duplicados con diferentes características.

```

EZEM01          CROSS SYSTEM PRODUCT
EZEM0217A No MSL selected
== >

ENTER = Continuar          PF3 = File and Exit
----- MSL -----
MSL FILE NAME          MSL ACCESS          CONNECT
                        ORDER              STATUS

Read/Write MSL:
-> usr1msl             -> 1

Read Only MSLs:
-> usr2msl             -> 2
->                     -> 3
->                     -> 4
->                     -> 5
->                     -> 6

                In all facilities:
PF1 = Help          PA2 = Cancel
    
```

### 5.2.2 Componentes de las Aplicaciones

A diferencia de los lenguajes de tercera generación, que en sus programas forman una sola entidad, las aplicaciones de CSP están formadas por diferentes componentes, los cuales pueden estar guardados en varias MSL's.

Las aplicaciones de CSP están formadas por 3 grandes componentes:

- Datos : Información de archivos o bases de datos, campos de trabajo o tablas.  
Un campo es usualmente definido como parte de una working o definición de una tabla.
- Mapas : Pantallas que se presentan en la terminal o reportes enviados a impresoras.
- Lógica : Grupo de sentencias que permiten ligar los datos con los mapas, acceder archivos o bases de datos y desplegar o recibir pantallas.

CSP es un lenguaje de programación que fomenta la programación modular y la compartición de código, pero exige estandarización en la programación.

### 5.2.3 Definición de Registros

CSP permite reservar espacio en las aplicaciones para las áreas de Entrada/Salida de los archivos o bases de datos, o los campos de trabajo.

Se define el tipo de registro, el archivo o base de datos al que está asociado y los campos que lo forman. Sólo contiene apuntadores a los nombres de los campos, ya que estos forman una entidad separada.

Soporta archivos VSAM KSDS, RRDS, ESDS e índices alternos; archivos SAM; áreas de trabajo o redefinición de registros; bases de datos jerárquicas (DL/I) o relacionales (SQL o DB2).



#### 5.2.4 Definición de Tablas o Working's de Trabajo

Las tablas o working's de trabajo son de gran utilidad para una aplicación, ya que facilitan la tarea de validación de campos de entrada de mapa, que ahorran lógica de programación la validación puede ser por valores válidos, valores inválidos o rangos válidos.

Una tabla es una colección estática de datos y procesos con un formato predefinido de columnas y renglones.

Las tablas o working's también se utilizan para guardar datos que podrán ser adaptados a un programa COBOL, PL/I, etc., el cual puede a su vez actualizar la base de datos.

No existe restricción en cuanto al número de tablas a utilizar en una aplicación, aunque se recomienda, declarar las tablas que sólo son utilizadas por la aplicación.

Las tablas pueden compartirse entre diferentes terminales, usuarios, etc., ya sean de lectura y/o escritura.

#### 5.2.5 Definición de Mapas

CSP permite definir las pantallas de entrada y salida para tener una comunicación usuario/aplicación (on-line), así como definir los reportes finales para el usuario. Esta facilidad le permite al usuario (programador) ahorrar líneas de codificación.

CSP realiza una validación automática de campos de entrada, como tipo de campo (numérico o alfanumérico), campo obligatorio, valores máximo y/o mínimo, verificación contra los valores de una tabla.

Permite la edición de campos de salida, como separación con comas, posiciones decimales, signos de pesos, formato de fecha, etc.

Se pueden definir hasta 5 teclas de función (PF's), además de PA1, PA2 y PA3, que le permitan al usuario de la terminal excluir la validación.

Especificar los atributos físicos de los campos, como brillantez, color, cursor, etc.

### 5.2.6 Definición de la Lógica de la Aplicación

Los procesos están formados obligatoriamente por una y sólo una opción de proceso (entrada/salida contra terminal o archivo, o estatutos del lenguaje) y, opcionalmente, instrucciones del lenguaje. Existen dos tipos de procesos :

1. Procesos principales

Define donde inicia la aplicación y además contiene un flujo que los une entre si. Este proceso es único, ya que la programación en CSP está modularizada en procesos y en grupos de estatutos.

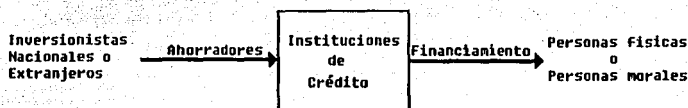
2. Procesos llamados

La diferencia con los procesos principales, es que son llamados con la instrucción PERFORM y además no contienen flujo.

CASO PRACTICO

6.1 MERCADO DE COBERTURAS CAMBIARIAS

El objetivo primordial del mercado de Coberturas Cambiarias contra riesgos de tasas de intereses nominales y del nivel de índices Nacional de Precios al Consumidor, es atraer grandes ahorradores nacionales y extranjeros, para las instituciones de crédito.



Las ventajas que ofrece invertir en el mercado de Cobertura Cambiaria, es que, el capital del inversionista no esta en riesgo, lo que esta en riesgo son los intereses que genera el capital del inversionista, dependiendo del tipo de instrumento crediticio seleccionado por el ahorrador.

El mercado de coberturas cambiarias es el mecanismo puesto en marcha por el Banco de México mediante el cual es factible cubrir las posibles fluctuaciones del peso mexicano frente al dólar de los Estados Unidos de América, lo anterior es importante cuando se realiza transacciones comerciales o financieras de carácter internacional.

Este mecanismo permite al comerciante o al inversionista establecer los flujos, los costos o los ingresos a futuro sin que en ningún momento una vez tomada la opción de coberturas cambiarias éstos varíen.

En la actualidad solamente existen coberturas cambiarias en México para dólares americanos.

La fluctuación cambiaria implica la depreciación o apreciación de una moneda con respecto a otra.

La depreciación de una moneda implica la pérdida de poder adquisitivo frente a una segunda moneda y la apreciación de una moneda implica la ganancia de poder adquisitivo frente a una segunda moneda.

### **6.1.1 Participantes del mercado de coberturas cambiarias de corto plazo**

#### ***Intermediarios***

Solo las casas de bolsa e instituciones de crédito del país autorizadas por el Banco de México podrán operar como intermediarios en el mercado de coberturas cambiarias de corto plazo.

La autorización dependerá de la calificación técnica (aplicación de exámenes) y de la capacidad de los solicitantes en términos de capital contable y global, así mismo, se tomará en cuenta la opinión de la Comisión Nacional de Valores, los intermediarios autorizados se darán a conocer en el diario oficial de la federación.

#### ***Clientes***

Los intermediarios celebrarán operaciones de coberturas con personas físicas o morales ya sean nacionales o extranjeras, que no tengan impedimento jurídico para realizarlas y que satisfagan los requisitos crediticios necesarios.

Los organismos públicos descentralizados, las instituciones de crédito, o los bancos, almacenes generales de depósito, uniones de crédito podrán realizar operaciones de coberturas siempre y cuando cuenten con la autorización por escrito del Banco de México para actuar como participante en el mercado de coberturas cambiarias.

Las casas de cambio en ningún caso celebran operaciones de coberturas cambiarias.

Los tipos de cobertura cambiaria son:

Compra de cobertura cambiaria	----- >	Protección Vs. Depreciación
Venta de coberturas cambiaria	----- >	Protección Vs. Apreciación

### 6.1.2 Tipo de cambio

6.1.2.1 Dólar ventanilla libre bancario (Tipo de cambio mínimo a la compra y máximo a la venta)

Es el tipo de cambio que surge a partir del decreto por el cual abroga el control de cambio publicado en el diario oficial de la federación el 10 de Noviembre de 1991, por medio de este desliz cambiario se reduce de cuarenta centavos diarios a veinte centavos.

Cabe notar que el tipo de cambio de ventanilla que actualmente rige es fijado por cada una de las instituciones de acuerdo a la oferta y demanda que tengan de la divisa.

#### 6.1.2.2 Tipo de Cambio para operaciones de Mayorco (Interbancario)

Este tipo de cambio es el que se utiliza para operaciones en dólares que sean mayores a cien mil dólares, dicho tipo de cambio se rige por la oferta y demanda del dólar en un momento determinado llegando a un punto de equilibrio entre la fuerzas de demanda y oferta, el tipo de cambio será libremente pactado entre las partes siempre y cuando éste no esté por debajo del nivel del tipo de cambio libre bancario a la compra ni por encima del tipo de cambio libre bancario a la venta.

6.1.2.3 Tipo de cambio libre representativo o tipo de cambio para solventar obligaciones denominadas en moneda extranjera pagaderas en la República Mexicana. (*Dólar fix*).

Es el tipo de cambio que resulta del siguiente procedimiento:

El Banco de México obtiene cada día hábil bancario de cuando menos seis intermediarios autorizados para operar en el mercado de cambio una muestra de las cotizaciones del tipo de cambio de mayoreo, la postura a la venta en la *fecha valor de 48 o spot*; dicha postura deberá reflejar las condiciones predominantes en el mercado de cambio al mayoreo.

Una vez obtenida dicha cotización, elimina la postura más baja y la más alta calculando con las restantes un promedio aritmético.

El tipo de cambio a la venta que resulta del cálculo anterior lo da a conocer la Bolsa Mexicana de Valores en esa misma fecha por medio de la publicación diaria que emite llamada *Movimiento diario del mercado de valores*.

El Banco de México lo publica al día siguiente en el diario oficial de la federación, este tipo de cambio entra en vigencia al día siguiente hábil bancario de su publicación en la gaceta oficial. *Este es el tipo de cambio que marca el vencimiento de toda cobertura cambiaria.*

El tipo de cambio de inicio para operaciones de cobertura cambiaria: será el que se pacte libremente entre el intermediario y el cliente, siempre y cuando refleje el nivel real del dólar *spot*.

6.1.3 ¿Cómo funciona la compra de coberturas cambiarias ?

*Fecha de inicio*

El banco y el cliente acuerdan libremente el precio, el plazo, el tipo de cambio de inicio y el monto en dólares a cubrir.

El comprador deberá pagar a más tardar el segundo día hábil posterior a la fecha de concertación a el banco la cantidad que resulte de multiplicar el precio por dólar cubierto por el monto establecido.

El cliente deberá tomar en cuenta cuatro fechas importantes de su operación de cobertura.

Fecha de inicio o concertación	Es aquella en la cual se pactó la cobertura cambiaria.
Fecha de liquidación de inicio	Fecha en que se deberá pagar la primera, cantidad que resulte de multiplicar el precio por dólar cubierto por el monto.
Fecha de vencimiento	Es aquella que resulta de sumar los días del plazo establecido a la fecha de concertación o de inicio. El plazo estará siempre expresado en días naturales, en esta fecha vence la cobertura, es decir, que el cliente por su lado ya no esta cubierto a partir de esta fecha contra fluctuaciones del peso mexicano frente al dólar americano, por su parte el intermediario ya no tiene a partir de esta fecha la contingencia cambiaria que adquirió al vender cobertura, esta fecha siempre será dos días hábiles bancarios antes de la fecha de liquidación.
Fecha de liquidación del vencimiento	En esta fecha se paga o se cobra según sea el caso el diferencial entre el tipo de cambio de inicio y el tipo de cambio <i>fix</i> , que rija para los contratos indexados a dólares en la fecha de vencimiento. La liquidación siempre será dos días hábiles posteriores a la fecha de vencimiento de las coberturas cambiarias. Como se puede apreciar en los puntos anteriores la fecha de vencimiento afecta la posición de

riesgo cambiario para el participante y en sentido inverso para el intermediario. La fecha de liquidación afecta la tesorería del intermediario y del participante, una más en sentido opuesto.

$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
0	2	28	30
Concertación		Vecimiento	
Liquidación de inicio		Liquidación	

#### ¿ Qué pasa al vencimiento ?

El cliente deberá tomar en cuenta que hasta este día la cobertura cambiaria lo protege contra variaciones en el tipo de cambio del peso frente al dólar de los Estados Unidos de América por lo que deberá decidir si le es conveniente optar por una compra de dólares americanos en el mercado libre de *mayoreo spot o fecha valor 48 horas*, ya que la liquidación de la cobertura cambiaria en cuestión se realizará en esa misma fecha valor.

Debemos recordar que es la cotización de venta valor *spot* la que va a determinar el tipo de cambio libre representativo *fix* mismo que es usado para determinar el nivel de liquidación de la cobertura cambiaria, dicho lo anterior debemos entender que si el cliente opta por la compra de dólares en el mercado libre destinará un monto moneda nacional muy cercano al que recibirá por el concepto de la liquidación de cobertura cambiaria, hay que hacer hincapié en que la cantidad podrá variar ya que no podemos asegurar el nivel exacto al que saldrá publicado el tipo de cambio libre representativo *fix*.

Resumiendo, si el cliente opta por la compra de dólares, estará cubierto en todo momento de la operación de cobertura.



### ¿ Qué pasa a la liquidación ?

El cliente deberá junto con el intermediario calcular el monto a liquidar por concepto de la cobertura cambiaria, dicho cálculo se determina de la siguiente manera:

$$(\text{TC. Final} - \text{TC. Inicial}) * \text{Monto Cobertura} = \text{Liquidación}$$

#### Ejemplo 1:

Compra de cobertura a 28 días	
Monto	US\$1,000,000.00
T.C. Inicial	3.100
T.C. Final	3.125

En este ejemplo el tipo de cambio libre representativo *fix* a la fecha de inicio o concentración fue de 3.100 y el tipo de cambio *fix* en vigor para la fecha de liquidación es de 3.125

Precio de la cobertura: N\$ 0.020

El cliente aseguró que el compraría dólares a 3.120. El día en que pacto el desembolso N\$0.020 por cada dólar cubierto

$$\begin{array}{rcl} \text{T.C. Inicial} & + & \text{Precio} & = & \text{Tipo de Cambio Asegurado} \\ 3.100 & + & 0.020 & = & 3.120 \end{array}$$

Obsérvese que el tipo de cambio *fix* al final de la operación fue más alto por N\$ 0.005, el intermediario deberá pagar la diferencia ya que el cliente aseguró un precio menor.

#### Cálculo de la liquidación:

(T.C. Final - T.C. Inicial) * Monto Cobertura	= Liquidación
(3.125 - 3.100) * 1,000,000.00	= Liquidación
0.025 * 1,000,000.00	= Liquidación
25,000	= Liquidación

El cliente recibirá del intermediario N\$ 25,000 por concepto de la cobertura cambiaria

Flujo del cliente:

	Paga	Cobra
Día 1(Concentración)	20,000	
Día 30 (Liquidación)	0	25,000
NETO		5,000

Ejemplo 2:

Compra de cobertura a 28 días	
Monto	US\$1,000,000.00
T.C. Inicial	3.100
T.C. Final	3.120
Precio de la cobertura	N\$ 0.025/dólar cubierto

El cliente aseguró que el compraría dólares a 3.125, el día en que pacto desembolso N\$ 0.025 pesos por cada dólar cubierto.

T.C. Inicial	+	Precio	=	Tipo de Cambio Asegurado
3.100	+	0.025	=	3.125

Obsérvese que el tipo de cambio *fix* final de la operación fue menor por N\$0.005 pesos. El participante deberá pagar la diferencia ya que el cliente aseguró un precio mayor.

Cálculo de la liquidación:

$$\begin{array}{rcl}
 (\text{T.C. Final} - \text{T.C. Inicial}) * \text{Monto cobertura} & = & \text{Liquidación} \\
 (3.120 - 3.100) * 1,000,000.00 & = & \text{Liquidación} \\
 0.020 * 1,000,000.00 & = & \text{Liquidación} \\
 & & 20,000.00
 \end{array}$$

El cliente recibirá del intermediario N\$20,000.00 por concepto de la cobertura cambiaria.

Flujo del cliente:

	Paga	Cobra
Día 1 (Concentración)	25,000	
Día 30 (Liquidación)	0	20,000
NETO	5,000	

#### 6.1.4 ¿ Cómo funciona la venta de coberturas cambiarias ?

Fecha de inicio

El banco y el cliente acuerdan libremente el precio, el plazo, el tipo de cambio de inicio y el monto en dólares a vender en cobertura.

El vendedor recibirá de el banco la cantidad que resulte de multiplicar el precio por dólar vendido en cobertura por el monto establecido; a más tardar el segundo día hábil posterior a la concentración de la cobertura.

Fecha de concertación

En dicha fecha se pacta la operación

Fecha de liquidación

Fecha en que se paga el monto equivalente en moneda nacional por la cobertura

Prima = (Precio/dólar vendido en cobertura) \* Monto

### ¿Qué pasa al vencimiento ?

El cliente deberá tomar en cuenta que este día de vencimiento ya no tiene la contingencia contra la fluctuación del peso frente al dólar, dicho lo anterior deberá determinar si es necesario pactar la venta de dólares que había vendido en cobertura, de ser afirmativa dicha necesidad, deberá pactar una venta de dólares americanos a 48 horas (dos días hábiles bancarios posteriores), debemos recordar que es la cotización a la venta de esta fecha, es la que va a determinar el tipo de cambio libre representativo o *fix* a la liquidación de la operación de cobertura, por lo que deberá pagar por concepto de la cobertura cambiaria una cantidad muy cercana a la que recibirá por la venta de los dólares, hay que hacer hincapié en que la cantidad podrá variar ya que no podemos asegurar el nivel exacto al que saldrá el tipo de cambio libre representativo.

### ¿Qué pasa al liquidar ?

El cliente deberá junto con el intermediario calcular el monto a liquidar por concepto de la cobertura cambiaria, dicho cálculo se determina de la siguiente manera:

$$(T.C. \text{ final} - T.C. \text{ Inicial}) * \text{Monto Cobertura} = \text{Liquidación}$$

Ejemplo 1:

Venta de cobertura	30 Días
Por	1,000,000.00
T.C. Inicial	3.100
T.C. Final	3.125

En este ejemplo el tipo de cambio de inicio o concertación fue de 3.100 y el tipo de cambio *fix* en vigor para la fecha de liquidación es de 3.125

Precio de la cobertura 0.020

El cliente asegura que el vendería dólares a 3.120, el día en que pactó desembolso N\$0.020 por cada dólar cubierto.

$$\begin{array}{rclclcl} \text{T.C. Inicial} & + & \text{Precio} & = & \text{Tipo de cambio Asegurado} \\ 3.100 & + & 0.020 & = & 3.120 \end{array}$$

Obsérvese que el tipo de cambio *fix* al final de la operación fue más alto por N\$0.005, el cliente deberá pagar la diferencia ya que el intermediario aseguró un precio menor.

Cálculo de la liquidación

$$\begin{array}{rclclcl} (\text{T.C. Final} - \text{T.C. Inicial}) * & \text{Monto Cobertura} & = & \text{Liquidación} \\ ( 3.125 - 3.100 ) * & 1,000,000.00 & = & \text{Liquidación} \\ 0.025 * & 1,000,000.00 & = & \text{Liquidación} \\ & \text{N\$ 25,000} & = & \text{Liquidación} \end{array}$$

El cliente pagará al intermediario N\$25,000 por concepto de la cobertura cambiaria

Flujo del cliente:

	Paga	Cobra
Día 1(Concentración)		20,000
Día 30 (Liquidación)	25,000	
NETO	5,000	

## Ejemplo 2:

Venta de cobertura a 30 días	
Monto	US\$1,000,000.00
T.C. Inicial	3.100
T.C. Final	3.120
Precio de la cobertura :	N\$ 0.025

El cliente asegura que el vendería dólares a 3.125, el día en que pacto desembolso N\$0.020 por cada dólar cubierto.

$$\begin{array}{rclcl} \text{T.C. Inicial} & + & \text{Precio} & = & \text{Tipo de cambio Asegurado} \\ 3.100 & + & 0.025 & = & 3.125 \end{array}$$

Obsérvese que el tipo de cambio *fix* al final de la operación fue menor por N\$0.005, el participante deberá cobrar la diferencia ya que el cliente aseguró un precio mayor.

## Cálculo de la liquidación

(T.C. Final - T.C. Inicial) *	Monto Cobertura	=	Liquidación
( 3.120 - 3.100 ) *	1,000,000.00	=	Liquidación
0.020 *	1,000,000.00	=	Liquidación
	N\$ 20,000	=	Liquidación

El cliente pagará al intermediario N\$20,000 por concepto de la cobertura cambiaria

## Flujo del cliente:

	Paga	Cobra
Día 1 (Concentración)		25,000
Día 30 (Liquidación)	20,000	
<b>NETO</b>		<b>5,000</b>

**6.1.5 ¿Cómo se determina el precio de una cobertura cambiaria ?**

El precio se acuerda libremente entre el participante y el intermediario el cual se expresará en pesos mexicanos por dólares cubiertos.

Los intermediarios no cargarán comisiones por concepto de coberturas cambiarias ya que estos a su vez realizan operaciones de coberturas por cuenta propia.

**6.1.6 ¿ Cuáles son los plazos posibles para celebrar operaciones de coberturas cambiarias ?**

El plazo de las operaciones deberá ser entre 1 día natural y 363 días naturales, éste será libremente establecido por el participante.

**6.1.7. ¿ Con qué garantías deben cumplir los clientes ?**

El cliente deberá constituir una prenda en caso de que le sea requerido entre 1 y los 5 días posteriores al cierre de la operación, dichos títulos deberán depositarse en la administración del banco y así mismo, tanto a la fecha de la concertación, tanto como a la fecha de la firma del contrato correspondiente, deberán tener un valor nominal que represente el margen de garantía especificado.

**Tratamiento fiscal**

En el diario oficial del 28 de diciembre de 1993, se especifica que las operaciones de coberturas cambiarias quedan exentas de impuestos para clientes extranjeros, mientras que para personas físicas y personas morales nacionales siguen en vigencia el establecido por el Banco de México.

En seguida se presenta un desglose de las principales funciones de la aplicación desarrollada.

## 6.2 SISTEMA DE COBERTURAS CAMBIARIAS DE LA CASA DE BOLSA DE PROBURSA

### 6.2.1 Arquitectura del equipo

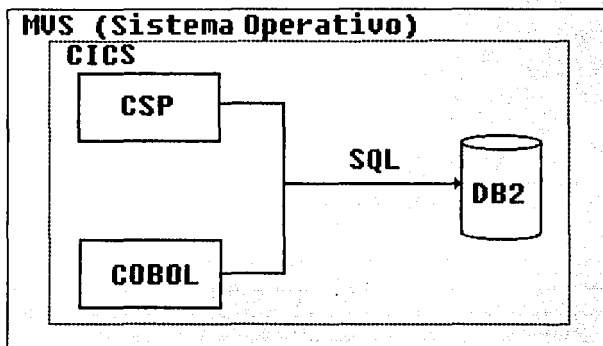


Figura 6.2 Arquitectura de la máquina 9121 de la casa de bolsa Probursa

El sistema de Cobertura Cambiarias de la casa de bolsa Probursa, se desarrolló en un equipo main frame IBM-9121, administrado por software de IBM. El sistema contiene proceso on-line y procesos batch's.

En un proceso o programa *on-line*, obtiene los resultados en forma instantánea (Existe una comunicación entre usuario aplicación). Además el programa no tiene que abrir archivos, el volumen de registros a procesar es mínimo, acepta y envía datos a terminales.

Un ejemplo común de un proceso *on-line*, un retiro o depósito de efectivo en un cajero automático, como se muestra en la figura 6.3

CSP, se ocupó como la herramienta para la parte on-line en este sistema.



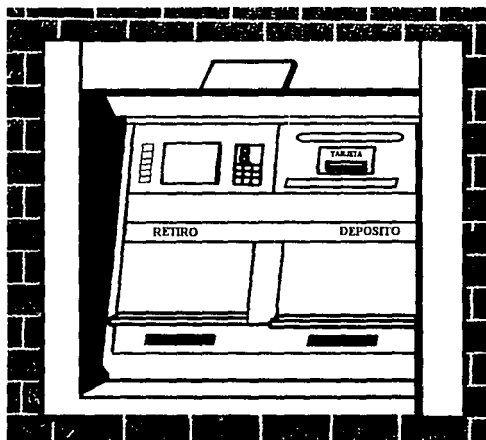


Figura 6.3 Cajero automático

Un proceso *batch*, es un programa que tiene codificadas las instrucciones tales como: **READ**, **OPEN**, **CLOSE**, **WRITE**. (no existe una comunicación entre usuario terminal). Además:

- Las áreas de *Entrada/Salida* se construyen en el propio programa.
- Se procesan archivos completos.
- La impresión de resultados es a veces de horas y a veces de días.

Un ejemplo real de estos tipos de proceso son los reportes, respaldos de base de datos, etc.



Figura 6.4 Proceso Batch

El sistema utilizó **COBOL-CICS**, **COBOL-DB2** para la generación de reportes y para dar de altas las operaciones pactadas con los clientes.

### 6.2.2 Módulos del sistema

En la figura 6.4 se puede ver el diagrama de bloques, que define cada uno de los módulos que contiene el sistema.

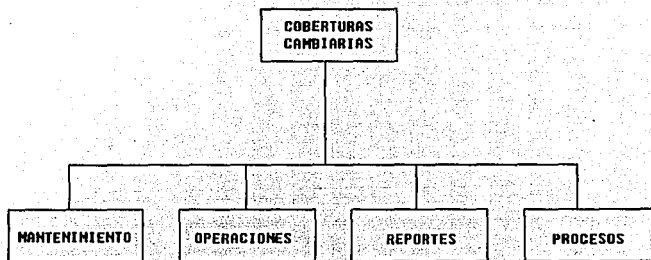


Figura 6.4 Diagrama del sistema de coberturas cambiarias

A continuación se describe cada uno de los procesos y/o programas que tienen contemplados cada módulo

#### MANTENIMIENTO

- 1 Porcentaje de aplicación de garantía
- 2 TIP
- 3 Tasa parámetro
- 4 Índice parámetro

#### OPERACIONES

- 5 Compra y venta de coberturas de cambio de tasas
- 6 Compra y venta de coberturas de cambio de índices
- 7 Actualización de garantías

## REPORTES

- 8 Utilidades
- 9 Valuación
- 10 Operación del día
- 11 Ponderado de posición
- 12 Vencimientos
- 13 Posición global
- 14 Mantenimiento de garantías
- 15 Resumen contable
- 16 Póliza contable
- 17 Operaciones vigentes

## PROCESOS

- 18 Vencimientos de coberturas
- 19 Actualización de garantías

En seguida se presenta una descripción del funcionamiento del programa que realiza la compra y venta de coberturas cambiarias para tasas, este programa fue seleccionado debido a que es la parte medular del sistema juntamente con el programa de coberturas cambiarias para índices.

### 6.2.3 Narrativa del Programa de Coberturas Cambiarias para Tasas

#### SISTEMA DE COBERTURAS CAMBIARIAS DE TASAS E INDICES DETALLE DE ESPECIFICACIONES DE USUARIO

##### MODULO : 5 COBERTURAS DE TASAS

##### ACTIVIDAD: COBERTURAS CAMBIARIAS DE RIESGO DE TASAS (COMPRA/VENTA)

El objetivo de esta función es capturar las coberturas cambiarias de riesgo de tasas para compra y/o venta.

##### DATOS DE ENTRADA

##### FUENTE

Tipo de Operación

Fecha

Anexo(control)

Contrato y/o clave

##### PROCESO/FUNCIONAMIENTO

Para referirse a los mensajes de errores ó informativos existe el catálogo de ellos, ver el anexo 5.

##### 1. Inicialización de variables.

- ✓ Limpiar mapa(pantalla)
- ✓ Limpiar working's de trabajo

##### 2. Llenar campos de default de la pantalla, utilizando programas generales (Ver anexo 4).

- ✓ Traer el usuario firmado
- ✓ Obtener el nombre de la compañía
- ✓ Centrar el texto de la compañía

##### 3. Presentar la pantalla para capturar la información requerida.

Si <enter> entonces

- ✓ Validar la información necesaria para lectura de la cuenta (Ver anexo 1).
- ✓ Lectura de la cuenta y tipo de persona (Ver anexo 2).
- ✓ Validar información necesaria para el cálculo del porcentaje de garantía de la cobertura cambiaria, llenar working's de trabajo (Ver anexo 3).
- ✓ Proteger campos del mapa sobre escritura

Finsi

Si <PF4> entonces

- ✓ Llenar working's para llamar a programas COBOL-CICS, de compra/venta.
- ✓ Llamar programas para efectuar la operaciones de compra/venta en la base de datos.

Si COW10507.WCODRET = 0 entonces

Presentar el número de folio con que se realizó la operación.

Limpiar pantalla

Desproteger campos de la pantalla

Sino

Move 362 a EZEMNO

Finsi

Poner el cursor en MO01.Operac

Sino

Mover 109 a EZEMNO

Finsi

Si diferente <enter> o <PF4> entonces

Mover 584 a EZEMNO

Finsi

---

**FRECUENCIA : Diaria**

---

**FECHA DE ELABORACIÓN :**

---

**SISTEMA DE COBERTURAS CAMBIARIAS DE TASAS E INDICES  
DETALLE DE ESPECIFICACIONES DE USUARIO**

**MODULO : 5 COBERTURAS DE TASAS (ANEXO 1)**

**ACTIVIDAD : VALIDA INFORMACION**

El objetivo de está función es verificar que la información capturada por el usuario sea correcta para obtener el número de cuenta del contrato deseado.

**PROCESO/FUNCIONAMIENTO**

1. Validar el tipo de operación

Si MO01.OPERAC = "C" ó  
MO01.OPERAC = "V" entonces

Continuar

Sino

Mover 1 a SEXIERR

Mover "El Tipo de Operación sólo puede ser C o V"

a EZEMSG

Poner cursor y abrillantar el campo MO01.OPERAC

Terminar

Finsi

2. Validar el Anexo o Control

Si MO01.ANEXO < 0 entonces

Continuar

Sino

Mover 1 a SEXIERR

Mover 3 a EZEMNO

Poner cursor y abrillantar el campo MO01.ANEXO

Terminar

Finsi

3. Validar el Contrato y/o Clave

Si MO01.CONTRATO < 0 y  
MO01.CLAVE < 0 entonces

Continuar

Sino

Mover 1 a SEXIERR

Mover 222 a EZEMNO  
Poner cursor y abrillantar el campo MO01.CONTRATO  
Terminar

Finsi

---

**FECHA DE ELABORACIÓN :**

---



**SISTEMA DE COBERTURAS CAMBIARIAS DE TASAS E INDICES  
DETALLE DE ESPECIFICACIONES DE USUARIO****MODULO : 5 COBERTURAS DE TASAS (ANEXO 2)****ACTIVIDAD : LECTURA DE CUENTA Y DEL TIPO DE PERSONA**

La finalidad de esta función es obtener de la base de datos el número de cuenta del contrato, así como todos los datos del detalle del mismo que son requeridos.

**PROCESO/FUNCIONAMIENTO**

## 1. Lectura de cuenta por contrato o por la clave del mismo

Si MO01.CONTRATO <> 0 entonces  
Lectura de la cuenta por contrato (Paso 2)  
Sino  
Lectura de la cuenta por clave (Paso 3)  
Finsi

## 2. Lectura de la cuenta por contrato

Leer ICUENTA, PORGTIA  
De CTAMERC  
Cuando IMARCO = MO01.Contrato  
Y ITIPOOPE = 'T'

Si Lectura = Correcta y  
MO01.APLCPOLT = 'P' entonces  
Mover CTAMERC.PORGTIA a MO01.PORCGARAN  
Sino  
Mover 1 a SEXIERR  
Mover 18 a EZEMNO  
Poner cursor y abrillantar el campo MO01.CONTRATO  
Terminar  
Finsi

## 3. Lectura de la cuenta por clave

Leer ICUENTA, PORGTIA  
De CTAMERC  
Cuando NCTABXI = MO01.CLAVE  
Y ITIPOOPE = 'T'

Si Lectura = Correcta y  
MO01.APLCPOLT = 'P' entonces  
Mover CTAMERC.PORGTIA a MO01.PORCGARAN

Sino  
Mover 1 a SEXIERR  
Mover 677 a EZEMNO  
Poner cursor y abrillantar el campo MO01.CONTRATO  
Terminar

Finsi

4. Lectura de la cuenta

Leer NAPELL1, NOMBRE, NABREV, IRFC,  
IPROM, IPERJUR, ICTAGLO  
De CUENTA  
Cuando CTAMARC.CUENTA = CUENTA.CUENTA

Si lectura = Correcta entonces  
Mover CUENTA.NOMBRE a MO01.NOMBRE  
Mover CUENTA.NAPPELL1 a MO01.APELL1  
Mover CUENTA.NABREV a MO01.CLAVE  
Mover CUENTA.IRFC a MO01.IRFC  
Mover CUENTA.IPROM a MO01.PROMOTOR

Sino  
Mover 1 a SEXIERR  
Mover 18 a EZEMNO  
Poner cursor y abrillantar el campo MO01.CONTRATO  
Terminar

Finsi

5. Lectura del tipo de persona de la cuenta

Si CUENTA.IPERJUR  $\in$  10,11,12 entonces

Si CUENTA.ICTAGO  $\in$  1 entonces

Move 1

a COW10507.ISR (*Persona Física*)

Finsi

Finsi

Si CUENTA.IPERJUR = 12 entonces

Si CUENTA.ICTAGO = 1 entonces

Move 1

a COW10507.ISR (*Persona Física*)

Finsi

Finsi

---

**FECHA DE ELABORACIÓN :**

---

---

**SISTEMA DE COBERTURAS CAMBIARIAS DE TASAS E INDICES  
DETALLE DE ESPECIFICACIONES DE USUARIO**

---

**MODULO : 5 COBERTURAS DE TASAS (ANEXO 3)**

---

**ACTIVIDAD :VALIDAR INFORMACIÓN PARA EL CALCULO DEL PORCENTAJE DE  
GARANTÍA**

La propósito es obtener el monto total de garantía para la cobertura.

---

**PROCESO/FUNCIONAMIENTO**

---

1. Validar el número de la Operación

Si MO01.OPER <= 0 entonces  
Mover 1 a SEXIERR  
Mover 651 a EZEMNO  
Poner cursor y abrillantar el campo MO01.OPER  
Terminar

Sino

Continuar

Finsi

2. Validar el índice nacional de precios al consumidor

Si MO01.INPC <= 0 entonces  
Mover 1 a SEXIERR  
Mover 651 a EZEMNO  
Poner cursor y abrillantar el campo MO01.INPC  
Terminar

Sino

Continuar

Finsi

3. Validar la fecha de vencimiento

Si MO01.FECHVEND = 0 ó  
MO01.FECHVENC = 0 ó  
MO01.FECHVENCA = 0  
Mover 1 a SEXIERR  
Mover 3 a EZEMNO  
Poner cursor y abrillantar el campo MO01.FECHVENC  
Terminar

Finsi

Validar que la fecha de vencimiento sea miércoles y que sea un día laborable

Si MO01.FECHVENC = Correcta

Continuar

Sino

Mover 1

a SEXIERR

Mover 1557

a EZEMNO

Poner cursor y abrillantar el campo MO01.CONTRATO

Terminar

Finsi

#### 4. Validar el monto

Si MO01.MONTO <= 0 entonces

Mover 1

a SEXIERR

Mover 651

a EZEMNO

Poner cursor y abrillantar el campo MO01.MONTO

Terminar

Sino

Continuar

Finsi

#### 5. Validar la política de aplicación de la garantía

Si MO01.APLCPOLT = 'P' ó

MO01.APLCPOLT = 'O' entonces

MO01.MONTGAR = MO01.PORCGARAN \* MO01.MONTO / 100;

Mover 1

a SEXIERR

Mover "Valor invalido en el campo. Debe ser P u O "

a EZEMSG

Poner cursor y abrillantar el campo MO01.MONTO

Terminar

Sino

Continuar

Finsi

---

**FECHA DE ELABORACIÓN :**

---

---

**SISTEMA DE COBERTURAS CAMBIARIAS DE TASAS E ÍNDICES  
DETALLE DE ESPECIFICACIONES DE USUARIO**

---

**MODULO : 5 COBERTURAS DE TASAS (ANEXO 4)**

---

**ACTIVIDAD : COBERTURAS CAMBIARIAS DE RIESGO DE TASAS (COMPRA/VENTA)**

El finalidad de este documento es establecer la relación entre los programas generales y la aplicación.

---

**RUTINAS GENERALES**

---

<b>RUTINA</b>	<b>WORKING</b>	<b>DESCRIPCIÓN</b>
VRB403O	VRW403	Obtiene nombre de la compañía
VRB045O	VXW004	Obtiene usuario firmado
AHR009	VXW108	Centrar títulos
VRC980O	VRW980	Rutina de abortar
VRC451O	---	Manejo de errores
VRB468D	VRW468	Autorización del usuario
VRB654O	VRW654	Autorización del contrato servicio
VRB4469D	VRW469	Autorización del contrato operación

---

**FECHA DE ELABORACIÓN :**

---

---

**SISTEMA DE COBERTURAS CAMBIARIAS DE TASAS E INDICES  
DETALLE DE ESPECIFICACIONES DE USUARIO**

---

**MODULO : 5 COBERTURAS DE TASAS (ANEXO 5)**

**ACTIVIDAD : COBERTURAS CAMBIARIAS DE RIESGO DE TASAS (COMPRA/VENTA)**

Catálogo de Errores

---

**RUTINAS GENERALES**

---

<b>ERROR</b>	<b>DESCRIPCIÓN</b>
3	El dato solicitado es requerido
18	El contrato no existe
109	Tecla de función invalida
362	Operación cancelada
420	El dato esta fuera de rango
584	Tecla invalida de transferencia

---

**FECHA DE ELABORACIÓN :**

---

### 6.3 EL FUTURO DE LOS MAINFRAMES Y RMBD'S

Pensar en el futuro, imaginarlo, resulta un buen punto de partida para tomar decisiones. ¿Cómo se ven las empresas usuarias para el año 2000? Para muchos en ese entonces, los main frames serán piezas de museo y si acaso, ocuparán el papel de un superservidor.

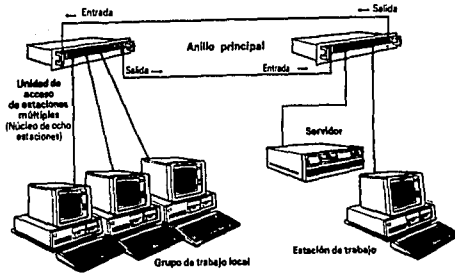
En nuestro país las tendencias son claras, -como se verá en seguida- a pesar de las opiniones encontradas que manifiestan tanto usuarios como directivos de diversas empresas.

Para muchos el precio cada vez más bajo de los avances tecnológicos es lo que determinará la desaparición de estos y del concepto de centro de cómputo como el sitio de concentración de la información.

Creo que el continuar operando una empresa con equipo main frame es una decisión difícil pues las tendencias mundiales se dirigen hacia el *downsizing o rightsizing* como también se le conoce; el significado de dichos conceptos en el ámbito informático está orientado a establecer la decisión de una empresa para adoptar una o varias plataformas de cómputo que en congruencia con su equipo base instalado, responda de una manera eficiente y eficaz a los requerimientos de los usuarios finales y del crecimiento del negocio.

Hay quienes consideran el downsizing, de manera elemental, como el cambio de main frames a soluciones basadas en redes de área local con computadoras personales o estaciones de trabajo que funcionan como servidores, sin embargo el main frame ocupa un lugar que otros equipos no pueden desplazar debido a sus características de seguridad, gran capacidad de almacenamiento y velocidad de transacciones.



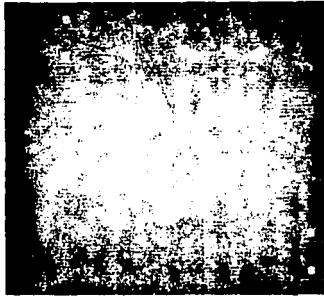


Entonces la correcta selección de equipo se orienta a adoptar o mantener un main frame cuando es necesario, orientando su uso en esencia al almacenamiento de datos y aprovechando sus capacidades para cumplir las veces de un *cliente* y un *servidor* a la vez, e incluso poder contener con seguridad las aplicaciones, protocolos, lenguajes, etc., presentes en cualquier compañía.

La tecnología cliente/servidor es una de las opciones tecnológicas que más énfasis ha cobrado desde principios de los 90's, ya que mediante su aplicación se resuelve que varios clientes incluso de distintas plataformas accedan a un servidor, y que como puede ser un main frame, puede ser un equipo mini, la idea es posibilitar un entorno con capacidad de crecimiento para que evolucione conforme se desenvuelve un negocio, aprovechando las ventajas que cada plataforma ofrece de tal forma que se complementen y realmente resulten en una solución a las empresas.

El efecto que provocó la acogida del downsizing en otros países, principalmente en Estados Unidos, en México no tuvo ese impacto debido a que la evolución de los equipos de gran tamaño no ha tenido un crecimiento masivo como en otros lugares, lo cual permitió que pocas empresas incursionaran en experimentos que los llevara a reemplazar sus equipos main frames.

Sin embargo el crecimiento de redes de área local, así como la tendencia en el uso de sistemas operativos abiertos, como es el caso de UNIX, hace que el concepto de cliente/servidor sea una alternativa con muchas posibilidades de éxito, ya que la idea fundamental radica en establecer una infraestructura de cómputo que evolucione junto con el negocio y que aproveche la plataforma y aplicaciones existentes.



En cuanto al futuro de DB2 no cabe duda que es un producto estratégico de IBM. Constituye la promoción del modelo relacional de Codd, un modelo que puede crecer para establecer nuevos tipos de datos. No obstante, el DB2 es un producto muy joven con experiencia en evolución de diseño y realización.

El área que más concierne y preocupa a los usuarios respecto al DB2 es el exceso de INPUT/OUTPUT (I/O) necesario para sostener la integridad referencial.

A principios del año 1992 IBM lanzó al mercado DB2 para PC el cual trabaja bajo sistema operativo OS/2, éste será un gran competidor de base de datos no sólo para ambientes main frame si no en muy poco tiempo en ambientes de redes.

## GLOSARIO

- Aplicación**                      Sinónimo de programa.
- Arquitectura de un Mainframe**      Básicamente la arquitectura de un mainframe divide una aplicación en dos partes. La primera son los procesos *on-line* y la segunda los procesos *batch*.
- Los procesos *on-line* son los procesos donde existe una continua comunicación entre el usuario y la máquina, estos procesos se usan de manera rutinaria y comprenden aplicaciones que se ejecutan repetitivamente para soportar las operaciones diarias de la empresa. Por ejemplo, una aplicación que soporta el depósito o retiro de efectivo en una institución bancaria, a este tipo de procesos también se les conoce como producción. En la actualidad para realizar este tipo de procesos en línea para equipos mainframes, las herramientas más usuales son CICS y CSP.
- Los procesos *batchs*, son aquellos que nos sirven como soporte para la toma de decisiones, se obtienen periódicamente de la(s) base(s) de datos de producción. Estos procesos son ejecutados por lo regular, una vez al día, mensualmente, etc. no requiere de una comunicación con el usuario. Para llevar a cabo este tipo de procesos es utilizado herramientas y lenguajes de programación complementarias como son QMF, SPUI, COBOL.
- Base de datos (Data Base)**      Un conjunto de archivos interrelacionados que es creado y manejado por un sistema de gestión o de administración de bases de datos (DBMS).
- Cualquier conjunto de datos almacenado electrónicamente.

- Banco de México** El Banco de México es el organismo que regula y estipula la forma de operar de los bancos que operan en territorio nacional, así como a las casas de bolsa del país. El Banco de México, controla las operaciones de la Casa de Bolsa de México.
- Bolsa de Valores** La bolsa de valores es un mercado organizado de intermediarios que representan los intereses de particulares, sociedades mercantiles y del mismo estado en el libre intercambio de valores, dentro de las reglas establecidas, tanto por este último o a través de sus diversas dependencias, como por los intermediarios que participan en el propio mercado.
- CICS** Es un conjunto de programas; producto de IBM que es utilizado para manejar aplicaciones en línea. También pudiésemos decir, que CICS es un lenguaje para comunicación entre terminales.
- COBOL (Common Business Oriented Language)** Lenguaje Común Orientado a los Negocios. Un programa de programación de alto nivel orientado a los negocios, que ha sido el principal lenguaje de aplicaciones comerciales en mini y macrocomputadoras. Su adaptación a las computadoras personales está aumentando continuamente. El COBOL es un lenguaje compilado y fue uno de los primeros lenguajes de alto nivel en ser desarrollado. Adaptado formalmente en 1960, deriva de un lenguaje llamado Flowmatic, de mediados de los 50. El COBOL requiere una escritura más extensa que otros lenguajes, pero el resultado es una mayor legibilidad. Por ejemplo, la sentencia multiply hourly-rate by hours-worked giving gross-pay (multiplique tarifa-horaria por horas-trabajadas dando sueldo-bruto) se explica por sí misma. El COBOL está estructurado en las siguientes divisiones:

IDENTIFICATION Identificación del programa  
ENVIRONMENT Tipo de equipamiento utilizado  
DATA Buffers, constantes y áreas  
de trabajo  
PROCEDURE El procesamiento (lógica del programa)

**CSP** (Proceso Secuencial de Comunicación), es una herramienta de cuarta generación, que nos facilita la programación de aplicaciones en línea. CSP, puede acceder directamente a las base de datos DL/I y DB2.

**DEC (Digital Equipment Corporation)** Una marca registrada de la Digital Equipment Corporation que es utilizada en muchos de sus productos; por ejemplo, DECmate, DECnet y DECsystem. En el pasado, todo el mundo se refería a la compañía por ésta sigla; no obstante, hoy la compañía incita al uso de Digital en su lugar.

**DBMS ( DataBase Management System)** Sistema de administración o gestión de bases de datos Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos. Acepta pedidos de datos desde un programa de aplicación y le ordena al sistema operativo transferir los datos apropiados.

Cuando se usa un sistema de gestión de bases de datos (SGBD, en inglés DBMS) los sistemas de información pueden ser modificados más fácilmente a medida que cambien los requerimientos de la organización. Nuevas categorías de datos pueden agregarse a la base de datos sin dañar el sistema existente.

**DB2** ( Data Base Dos ), en un manejador de base de datos relacional. DB2 fue la primera base de datos relacional reconocida mundialmente como tal. DB2 es la base de datos relacional, mientras SQL es el manejador de la misma.

- 
- DL/I** (Data Lenguaje/ I), es un sistema destinado al manejo de base de datos que sirve de interfaces entre el sistema operativo y los programas de aplicaciones batch y on-line desarrollados para dar mantenimiento a la información contenida en la base de datos jerárquica.
- DRDBMS (Distributed Relational DataBase Management System)** Sistema de gestión de bases de datos relacionales distribuidas.  
Un sistema de gestión de bases de datos relacionales que administra bases de datos distribuidas. Véase database management system.
- Gateway (pasarela, puerta de acceso)** Una computadora que conecta dos tipos diferentes de redes de comunicaciones. Realiza la conversión de protocolos de una red a otra. Por ejemplo, una puerta de acceso podría conectar una red LAN de computadoras. Nótese la diferencia con bridge, el cual conecta redes similares.
- IBM (International Business Machines Corporation)** Corporación Internacional de Máquinas para Negocios. La compañía informática más grande del mundo. Comenzó en 1911 en la ciudad de Nueva York cuando se creó la Computing-Tabulating-Recording Co. (CTR) gracias a la fusión de The Tabulating Machine Co. (la compañía de tarjetas perforadas de Hollerith en Washington), la International Time Recording Co. (fabricante de relojes marcadores de tiempo en el estado de Nueva York), la Computing Scale Co. (fabricante de balanzas y cortadoras de alimentos en Dayton, Ohio), y la Bundy Manufacturing (fabricante de relojes marcadores de tiempo en Poughkeepsie, N.Y.). La CTR comenzó con 1200 empleados y un capital evaluado en US\$ 17,5 millones.  
En 1914, Thomas J. Watson, Sr., accedió al cargo de gerente general. En los 10 años siguientes, se deshizo de las actividades que no tuvieran que ver con la tabulación, y la transformó en una empresa internacional denominada IBM, en 1924, Watson fue el responsable de inculcar en sus

empleados un comportamiento profesional muy estricto que distinguió a la gente de IBM del resto del ambiente.

IBM logró un éxito espectacular fabricando máquinas tabuladoras y las tarjetas perforadas con las que se las alimentaba. Desde la década del 20 hasta la del 60, desarrolló una inmensa clientela que era ideal para su conversión a las computadoras.

IBM lanzó su negocio informático en 1953 con el modelo 701 e introdujo el 650 un año más tarde. Para fines de los años 50, la 650 era la computadora más ampliamente utilizada en el mundo, con más de 1.800 sistemas instalados. La 1401 anunciada en 1959, fue un segundo éxito en computadoras, y para mediados de la década del 60, se estimaba que unas 18.000 se encontraban en uso.

En abril de 1964, se anunció el System/360, que fue la primera familia de computadoras compatibles desarrollada hasta entonces. La 360 tuvieron un éxito enorme y sentaron un estándar para todas las macrocomputadoras IBM hasta hoy en día.

A pesar de que IBM logró sus éxitos más resonantes gracias a la compatibilidad, construyó también una variedad de minicomputadoras incompatibles a lo largo de la década del 70, incluyendo los System/3, System/34, System/38, Series de 1 y 8100.

En 1981, introdujo la PC. El caótico mundo de las computadoras personales necesitaba estabilidad y la PC se convirtió rápida, en el estándar de la industria.

Para la década del 90, la meta de IBM es integrar sus diferentes líneas de productos mediante un plan maestro denominado SAA (Systems Application Architecture - arquitectura de aplicación de sistemas). SAA intenta proveer de interfaces comunes, de modo tal que los usuarios puedan interactuar con, e intercambiar información fácilmente entre todas las máquinas de IBM desde micro hasta macrocomputadoras.

**IBM Mainframes** La serie de computadoras siguientes proceden todas de la (sistema de gran arquitectura de System/360 presentada en 1964. tamaño de IBM)

<u>Fecha</u>	<u>Nombre de serie y modelos</u>
1964	System/360 (Modelos 20 al 195)
1970	System/370 (Modelos 115 al 168)
1977	Serie 303x (3031, 3032, 3033)
1979	Serie 43xx (4300 al 4381, ES/4381)
1980	Serie 308x (3081, 3083, 3084)
1986	Serie 3090 (Modelos 120 al 600, ES/3090)
1990	System/390 (Modelos 120 al 900)

**IMS (Information Management System)** Un sistema de administración de bases de datos jerárquicos de IBM que se emplea en sus grandes macrocomputadoras. El IMS es un sistema de primera generación que fue ampliamente usado durante la década del 70. Como el CICS, el IMS provee la capacidad de procesamiento de transacciones y manipula automáticamente los detalles de las comunicaciones y redes SNA.

**Intel Corporation)** (Intel) Una fábrica líder de dispositivos semiconductores que fue fundada en 1968 por Bob Noyce y Gordon Moore en Mountain View, California. Un año más tarde introdujo su primer producto, un chip de RAM bipolar estático de 64 bits. Para 1971, sus exitosos chips de memoria comenzaron a reemplazar los núcleos magnéticos para almacenamiento de memoria.

A pesar de ser conocida principalmente por su familia de microprocesadores 8086, Intel inventó realmente el microprocesador en 1971. Como respuesta a un pedido de chips para calculadoras de fabricante japonés Basicom, el ingeniero de Intel Marcian E. "Ted" Hoff decidió que tenía más sentido diseñar una máquina de propósito general. Como resultado de ello nació el chip 4004, que fue el primer microprocesador del mundo.



A través de los años, Intel ha desarrollado una amplia variedad de chips y otros productos a nivel de tarjeta, así como su propia arquitectura de bus MULTIBUS que se utiliza en numerosas aplicaciones industriales. Intel comenzó con 12 personas e ingresos de \$ 2.672 para el primer año. En 20 años, ha crecido hasta alcanzar ingresos superiores a los tres mil millones de dólares anuales, con más de 20.000 empleados.

**ISAM (Indexed Sequential Access Method)**

Método de acceso secuencial indexado. Un método de acceso a disco impliamente usado que almacena los datos en forma secuencial, al tiempo que mantiene un índice de campos clave para todos los registros en el archivo, lo que le otorga capacidad de acceso directo. El orden secuencial del archivo es el más comunmente usado para el procesamiento por lotes y la impresión, tal como por número de cuenta o nombre.

**ISO (International Standards Organization)**

Organización Internacional de Estándares. Una organización que establece estándares (normas) internacionales, fundada en 1946, con sede en Ginebra. Se ocupa de todos los campos, excepto la electricidad y la electrónica, las cuales están ya desde antes bajo la jurisdicción de la IEC (International Electrotechnical Commission - Comisión Electrotécnica Internacional), también radicada en Ginebra. Con respecto a los estándares de procesamiento de la información, la ISO y la IEC crearon recientemente la JTC1 (Joint Technical Committee - Comité Técnico Conjunto) para la tecnología informática.

La ISO desarrolla su trabajo a través de más de 160 comités técnicos y 2300 subcomités y grupos de trabajo, y está constituida por las organizaciones de estándares de más de 75 países, algunas de las cuales sirven como secretariados para estos cuerpos técnicos. En los EE.UU., la ANSI es miembro de la ISO.

Para mayor información, dirigirse a American National Standards Institute, 1430 Broadway, New York, NY 10018.

**Jerárquico**

Estructura compuesta por diferentes niveles, como un diagrama de organización de una compañía. Los niveles más altos tienen control o prioridad sobre los niveles más bajos. Las estructuras jerárquicas son una relación de uno a muchos, ya que cada elemento tiene uno o más elementos debajo de él.

En comunicaciones, una red jerárquica se refiere a una sola computadora que tiene el control sobre todo los nodos conectados a ella.

**Join (unir)**

En administración de bases de datos relacionales, aparear un archivo con otro sobre la base de alguna condición, creando un tercer archivo con datos de los archivos apareados. Por ejemplo, un archivo de clientes puede ser unido con un archivo de pedidos creando un archivo con registros para todos los clientes que compraron un producto particular.

**LAN**

1) (Local Area Network) (Red de Area Local) Red de comunicaciones que sirve a usuarios dentro de un área geográficamente limitada.

2) (Local Area Network) (Red de Area Local) Red de computadoras personales dentro de un área geográficamente confinado que se compone de servidores, estaciones de trabajo, sistemas operativos de redes y un enlace de comunicaciones.

Los servidores son máquinas de alta velocidad que pueden contener programas y datos que todos los usuarios de redes puedan compartir. Una estación de trabajo es una máquina de usuario, que puede funcionar como una computadora personal autónoma. Las estaciones de trabajo sin disquete o las estaciones de trabajo sólo disco flexible recuperan todo el software y todos los datos del servidor. Como con cualquier computadora personal, una impresora se puede unir a una estación de trabajo o a un servidor y puede ser compartida por los usuarios de redes.

Las LAN's pequeñas pueden permitir que cada estación de trabajo funcione como un servidor y que todos los usuarios accedan a los datos en todas las máquinas. Estas redes por a generalmente son más simples de instalar y de dirigir, pero un servidor dedicado siempre podrá manejar más transacciones por segundo. En redes de alto volúmen de transacciones se utilizan muchos servidores.

Para conectar redes iguales se utilizan puentes y gateways, los cuales conectan diferentes tipos de red, permitiendo que una red de computadoras personales, por ejemplo, se interconecten una red de minicomputadoras o con una computadora de gran tamaño.

El software de control en una LAN es el sistema operativo de la red, como NetWare, Lantastic y Appletalk, que reside en el servidor. En cada estación de trabajo reside un componente del software y permite que una aplicación lea y escriba datos de un servidor como si estuviera en la máquina local.

La transferencia física de datos la lleva a cabo el método de acceso, como Ethernet o Token Ring, que aparecen en forma de adaptadores de red (NICs) y se conectan a cada computadora. El enlace real, o la vía de acceso de las comunicaciones es el cable (un par trenzado, un cable coaxial, fibra óptica..) que se conecta a cada adaptador de red y a su vez conecta estaciones de trabajo y servidores juntos.

**Medios de Almacenamiento**

Hay muchas variedades de paquetes de discos, cartuchos de disco, discos flexibles, cintas de dos bobinas, cartuchos de cinta y cassettes de cinta. Cada uno posee su propia forma y tamaño y puede utilizarse sólo en unidades diseñadas para adaptarse a ellas.

En los modelos removibles, el estándar físico es sólo la mitad de la cuestión de la compatibilidad. La otra mitad la constituyen los patrones de grabación, que son invisibles

para el ojo humano. Las cintas y discos magnéticos vírgenes, recién retirados de su envase, son superficies de grabación en blanco. Las pistas reales de grabación son dispuestas sobre la superficie por la cabeza de lectura/escritura de la unidad de almacenamiento. De esta forma, el mismo disco flexible que almacena 720.000 bytes en una unidad de disco, puede contener 800.000 bytes si es formateado por otro. Si la computadora lee una cinta incompatible, o lee y escribe un disco incompatible, señalará un error de lectura/escritura.

Para las minicomputadoras y macrocomputadoras, es común, como medio intercambiable de almacenamiento, el carrete de cinta magnética de media pulgada. Para las computadoras personales, se utilizan comúnmente el minifloppy de 5,25 pulgadas y el microfloppy de 3,5 pulgadas.

**MVS (Multiple Virtual Storage)** Almacenamiento Virtual Múltiple Presentado en 1974, el sistema operativo primario utilizado en grandes sistemas de computadoras de IBM (los otros son VM y DOS/VSE). MVS es un sistema operativo orientado al procesamiento por lotes que maneja una gran cantidad de memoria y espacio de disco. Las operaciones en línea se obtienen con CICS, TSO y otros programas de sistemas.

**MVS/XA (MVS/eXtended Architecture)** (Arquitectura Ampliada), presentado en 1981 con la arquitectura 370/XA de IBM, gestiona las mejoras con 2GB de memoria virtual.

**MVS/ESA (MVS/Enterprise Systems Architecture)** (Arquitectura de Sistemas de Empresa) presentado en 1988 con la arquitectura ESA/370 de IBM, gestiona las mejoras realizadas en los sistemas grandes de computadoras a gran escala, con 16 TB de memoria virtual. MVS/ESA presentada en 1990, se ejecuta en todos los modelos de la línea de productos System/390 ES/9000.

---

**NCR Corporation)** (NCR) Un gran fabricante de computadoras y terminales financieras. En 1884, John Henry Patterson adquirió la National Manufacturing Company (Compañía Nacional de Manufactura) de la ciudad de Dayton, Ohio y le cambió el nombre por National Cash Register (Registradora de Efectivo Nacional). Se convirtió en la compañía líder en cajas registradoras del país y, para 1911, había vendido su millonésima caja registradora.

A través de las décadas del 30 y del 40, la NCR ganó una reputación en las industrias bancarias y de ventas minoristas como fabricante de máquinas de contabilidad que fueron usadas para registrar los resúmenes de cuenta de los clientes. NCR se ha especializado en ambas industrias desde entonces.

En 1952, adquirió la Computer Research Corporation (Corporación para la Investigación en Computación), una firma electrónica recién iniciada, que para 1955 había construido y vendido 30 sistemas de procesamiento de datos orientados a la ciencia, denominada CRC 102. Reorientando su nueva adquisición hacia el mundo de los negocios y sorteando la era de las válvulas de vacío, NCR introdujo una computadora transistorizada en 1957, denominada 304. Esta aceptaba datos desde sus cajas registradoras y terminales bancarias por medio de cintas de papel. Gracias a su gran fiabilidad fué ampliamente aceptada en las industrias bancarias y de ventas minoristas.

Uno de los dispositivos más novedosos de NCR fué su unidad de almacenamiento CRAM (Card Random Access Memory - Memoria de Acceso Aleatorio por Tarjetas), introducida en 1961. Estaba compuesta por cartuchos removibles que contenían tiras de cinta magnética, ofreciendo gran capacidad de almacenamiento periférico para la época. Las unidades CRAM eran más fiables que los dispositivos de tiras magnéticas de los fabricantes de la competencia.

Desde la serie Century en los años 60 y la serie Criterion de los años 70, hasta los macroordenadores 9300 de los 80, la

NCR se ha mantenido al corriente de los tiempos, proveyendo una completa línea de sistemas integrados de punto de venta y computación financiera.

**Network red**

- 1) Una disposición de objetos que están interconectados. Véase local area network y network database.
- 2) En comunicaciones, los canales de transmisión y el soporte de hardware y software.

**Online (en línea)**

1) Un dispositivo periférico, tal como una terminal o una impresora, que está listo para operar. Una impresora puede estar conectada y encendida, y aún estar fuera de línea, si el indicador ONLINE o SEL está apagado. Pulsando el botón de ONLINE por lo general se la colocará en línea nuevamente.

2) Un sistema informático en línea se refiere a un sistema con terminales, pero no implica el modo de operación del sistema. Por ejemplo, un sistema colector de datos es un sistema en línea que acepta y almacena datos de varias terminales, pero no actualiza ningún archivo maestro.

Un sistema interactivo es un sistema en línea que implica entrada y actualización de datos.

Un sistema de procesamiento de transacciones es un sistema en línea que actualiza los archivos necesarios a medida que entra el trabajo, tal como en un sistema de procesamiento de órdenes de compra.

Un sistema en tiempo real es un sistema en línea que provee respuesta inmediata a un requerimiento.

Aunque completamente redundante, no es incorrecto decir que se posee un "sistema de procesamiento de transacciones interactivo, en línea y en tiempo real", especialmente si se desea impresionar a alguien. Sólo es de esperar que ese alguien no sea un analista de sistemas experimentado.

---

**Organizaciones de Estandares.** de Las siguientes organizaciones establecen los estándares (normas) para computadoras, comunicaciones y productos relacionados en todo el mundo

**ANSI** - American National Standards Institute (Instituto americano de estándares nacionales)

**EIA** - Electronic Industries Association (Asociación de industrias electrónicas)

**IEEE** - Institute of Electrical and Electronics Engineers (Instituto de ingenieros electricistas y electrónicos)

**NIST** - National Institute of Standards & Technology (Instituto nacional de estándares y tecnología, anteriormente National Bureau of Standards - Oficina Nacional de Estándares).

#### **Internacionales**

**CCITT** - Consultative Committee for International Telephony & Telegraphy (Comisión consultiva para telefonía y telegrafía internacional).

**ISO** - International Standards Organization (Organización internacional de estándares)

**IEC** - International Electrotechnical Commission (Comisión electrotécnica internacional)

#### **Estandares de hecho**

Cuando un producto de un fabricante se utiliza ampliamente, se convierte en un estándar de hecho. Apple, Ashton-Tate, Digital, HP, IBM, Intel, Lotus, Microsoft, Motorola y muchos otros proveedores de hardware y de software han establecido estándares de hecho.

---

## EL FUTURO

La cuestión de los estándares y la compatibilidad es un dilema que nunca termina. Sin embargo, el hecho es que podrían crearse estándares que abarcaran el futuro y permitieran una expandibilidad mucho mayor que la que permiten actualmente. Irónicamente, en un campo que se encuentra en plena vanguardia del futuro, esta industria tiene una versión muy miope del mismo.

Algún día, deberá implementarse un estándar para definir el estándar a fin de que un programa pueda preguntar a otro que lenguaje habla. Un programa podría también interrogar un archivo de datos y determinar su formato. Si el programa no puede entender el lenguaje del otro programa o el formato del archivo, seguiría existiendo el problema de intermediación, como existe actualmente. Sin embargo, al hacerse más multilingües los programas, un protocolo de identificación estándar sería un gran avance hacia el establecimiento de un enlace de inteligencia artificial entre todas las computadoras en el futuro.

## OS/2

Sistema operativo multitarea de un solo usuario para PCs 286 y superiores. Las versiones de 16 bits se han desarrollado conjuntamente por IBM y Microsoft. Las versiones de 32 bits se han desarrollado independientemente.

Es un sistema avanzado que no está confinado al límite inferior de 1 MB de DOS. Aunque se han añadido órdenes nuevas, muchas de OS/2 son las mismas que las de DOS. Las versiones de 16 bits pueden direccionar 16MB de RAM y 1GB de memoria virtual. OS/2 requiere 4MB de RAM (excepto la versión 1.3), aunque se encuentra ejecutándose en computadoras con 8MB y más de 8.

El Gestor de Presentación (Presentation Manager) de OS/2 (OS/2 PM), que también forma parte del SAA de IBM, proporciona una interfaz de usuario similar al Windows 3.x.

IBM tiene en propiedad una Edición Extendida que incluye comunicaciones y gestión de bases de datos.



**Procesos batch** Un proceso batch, es un programa que tiene codificadas las instrucciones tales como: READ, OPEN, CLOSE, WRITE. (no existe una comunicación entre usuario terminal). Además:

- Las áreas de Entrada/Salida se construyen en el propio programa.
- Se procesan archivos completos.
- La impresión de resultados es a veces de horas y a veces de días.

**Procesos on-line** En un proceso o programa on-line, obtiene los resultados en forma instantánea, claro en una terminal, (Existe una comunicación entre usuario aplicación ). Además

- El programa no tiene que abrir archivos, el volumen de registros a procesar es mínimo, acepta y envía datos a terminales.

**PL/I (Programming Language 1)** Un lenguaje de programación de alto nivel de IBM que fue presentado en 1964 con su familia de computadoras System/360. Fue diseñado para combinar características del COBOL y del FORTRAN y finalmente suplantarlos, lo cual por supuesto nunca sucedió. Un programa PL/I está formado por módulos llamados procedures (procedimientos) que pueden ser compilados independientemente. Hay siempre un procedimiento principal con o sin procedimientos adicionales. También se proveen funciones, que pasan argumentos de envío y de regreso.

**SQL (Structured Query Language)** es un Lenguaje Estructurado de Consultas. SQL en muy poco tiempo se ha convertido en el manejador de base de datos relacional más utilizado en el mundo de la informática, debido a que las organizaciones mundiales

---

ISO y ANSI, declaron a SQL como el estándar para manejadores de bases de datos relacional.

Lenguaje de Consulta Estructurado Lenguaje utilizado para interrogar y procesar datos en una base de datos relacional. Desarrollado originalmente por IBM para sus macrocomputadoras, han habido muchas implementaciones creadas para aplicaciones de base de datos en mini y microcomputadoras. Las órdenes (mandatos) de SQL se pueden utilizar para trabajar interactivamente con una base de datos, o pueden incluirse en un lenguaje de programación para servir de interfaz a una base de datos.

La siguiente consulta SQL selecciona clientes con límites en sus créditos de al menos 5.000 dólares y los introduce en una secuencia desde el límite de crédito más alto al más bajo.

```
SELECT NOMBRE, CIUDAD, ESTADO, CODIGO ZIP
FROM CLIENTE
WHERE LIMITE DE CREDITO > 4999
ORDER BY LIMITE DE CREDITO DESC
```

**Transaction  
(transacción)**

Actividad o solicitud. Ordenes de compra, ventas, cambios, altas y bajas son ejemplos de transacciones que se registran en un entorno de información para negocios. Las consultas y demás solicitudes son también transacciones para la computadora, pero normalmente se las procesa sin registrarlas en el sistema. El volumen de transacciones es un factor preponderante en la determinación del tamaño y la velocidad de un sistema informático.

**Transaction  
processing  
(Procesamiento  
de Transacciones)**

Procesamiento de las transacciones en el momento de ser recibidas por el sistema. Los sistemas de procesamiento de transacciones, también llamados "en línea" (online) o "de tiempo real" (realtime), actualizan los archivos maestros tan pronto como las transacciones se introducen en las terminales o llegan por las líneas de comunicaciones. Nótese la diferencia con batch processing, el cual almacena las

transacciones para actualizar posteriormente los archivos necesarios.

Si usted guarda los recibos en una caja de zapatos, y los suma a fin de año por razones impositivas, eso es procesamiento por lotes. Sin embargo, si usted compra algo e inmediatamente añade la suma abonada al total corriente, eso es procesamiento de transacciones.

---

## CONCLUSIONES

Es claro el cambio tecnológico tan fuerte que se ha dado, redes de area local, tendencia hacia soluciones cliente/servidor, orientación a sistemas abiertos, downsizing y lo más reciente multimedia; todo ello ha sembrado la duda con respecto a la factibilidad de seguir trabajando con mainframe.

Aún cuando el precio es uno de los inhibidores actuales para la venta de mainframes pienso que los mainframes sacarán partido de sus cualidades más relevantes: su capacidad de almacenamiento y la superior velocidad de procesamiento que ofrecen con respecto a equipos pequeños. IBM, empresa líder en nuestro país en cuanto a mainframes de refiere, ha expresado su intención de continuar su inversión en tecnología y desarrollo de estos equipos, así como dando soporte a su clientes; y consideran que para 1996 el procesamiento y capacidad de almacenamiento mejorará por lo menos en un 25% además entre sistemas abiertos y cada vez mayores mejoras tecnológicas, el precio disminuirá.

No obstante que IBM instaló en México en 1991 la mayor cantidad de sistemas empresariales hechas en un solo año y que continúa invirtiendo en dicha tecnología, también esta enfatizando sus soluciones hacia cliente/servidor.

Por lo anterior, tal vez en un futuro, el mainframe se convierta en un super-servidor, de tal forma que las empresas lo vean como un elemento más de su ambiente de cómputo cliente/servidor, en donde necesidades de seguridad, control de información, administración de redes, y almacenamiento masivo de información no dejarán de ser indispensables para diversas empresas o instituciones. Por otra parte ofrecerán a sus usuarios acceso transparente a los datos de la empresa sin importar su origen, plataforma o marca, respaldo más eficiente de su información y comunicaciones a cualquier elemento informático de su compañía.

Hay que aceptar la idea de que en el futuro, los centros de datos no se estructurarán sólo con mainframes; contendrán procesadores especializados y servidores en donde funcionen bases de datos y procesos de transacciones en línea (*OLTP*), todo ello conectado seguramente a través de canales de fibra óptica.

---

En cuanto a México, existe una gran ventaja, el tener por vecino a un mercado de primer mundo, permite observar que sucedió, ante diversas circunstancias, sin tener que vivirlo, lo cual da la oportunidad de reaccionar antes. Con esto no quiero decir que el denominado downsizing -que tiene tanto auge en Estados Unidos desde principios de los 90's- sea malo sino que hay que saber cómo y cuándo hacerlo.

Creo que el mercado de mainframes no está extinto, aunque muchos los consideren ya como piezas de museo; algunas empresas lo adaptarán con diversos equipos más modernos y otras seguirán perteneciendo a un grupo, tal vez selecto, de usuarios ya sean grandes corporaciones o empresas que llegan a crecer tan rápido y con grandes necesidades de asegurar y almacenar su información, que un upsizing probablemente sea conveniente.

En cuanto a DB2 y CSP, probablemente algunas empresas ha adquirido estas herramientas sin conocer las ventajas y desventajas de éstas, tal vez se han dejado llevar tan solo porque son las más novedosas en el mercado o debido a que los competidores las tienen, el hecho es que la combinación de ambas ofrece grandes ventajas en el mercado, ya que se basa en una de las mejores teorías para la manipulación de la información, además de estar estrechamente ligado a SQL, estándar reconocido por la ISO y la ANSI (en cuanto a manejadores de bases de datos relacionales se trata).

La inversión de capital en la casa de bolsa es de suma importancia para la economía de un país, debido a que el dinero que ingresa por la compra de acciones de una empresa inyecta dinero fresco a la misma para la apertura de nuevos proyectos o la ampliación de su mercado.

Las coberturas cambiarias ofrecen al inversionista una opción de inversión, ya que invierte su dinero en la compra o venta de coberturas cambiarias, pudiendo fijar un rendimiento o el interés del monto con algún instrumento de la federación como son los CETES, PETROBONOS, etc.

Las ventajas y riesgos que tiene el inversionista es la fluctuación del peso con el dólar, debido a que el inversionista invierte su dinero en moneda nacional y una vez que se cumple el plazo de vencimiento, el monto puede ser cambiado a dólares de acuerdo al tipo de cambio que riega para operaciones interbancarias *f.v.*

---

## BIBLIOGRAFIA

**Groff, James R. y Weinberg, Paul N.**

Aplique SQL

Ed. Mc. Graw Hill

México D.F., 1992.

p.p. 619.

**Heydt, Robert y Heydt, Diane**

DB2 versión 2

Ed. Wiley

U.S.A., 1989.

p.p. 242

**Evan, Jorge I. y Cordero, Luis G.**

Estructura de datos

Ed. UNAM

México D.F., 1984.

p.p. 219.

**Writh, Niklaus**

Algoritmos y estructuras de datos

Ed. Prentice Hall

México, D.F., 1987.

p.p. 305.

**Wolf de Cordoba, Heidi**

Introducción a las bases de datos

Red

Año III, núm. 26

México D.F., Octubre de 1992.

p.p. 57-60.

---

**Wolf de Cordoba, Heidi**  
*Introducción a las bases de datos*  
Red  
Año III, núm. 27  
México D.F., Noviembre de 1992.  
p.p. 64-67.

**Wolf de Cordoba, Heidi**  
*Introducción a las bases de datos*  
Red  
Año III, núm. 27  
México D.F., Diciembre de 1992.  
p.p. 51-55.

**Tsai, Alice Y. H.**  
*Sistemas de bases de datos Administración y Uso*  
Ed. Prentice Hall  
México D.F., 1992.  
p.p. 607.

**Marmolejo, Martín**  
*Inversiones*  
Ed. Instituto Mexicano de Ejecutivos de Finanzas  
México D.F., 1992.  
p.p. 509.