



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

SISTEMAS DE INFORMACION:

"UNIFICACION INSTITUCIONAL DE LAS BASES DE DATOS DE LA BOLSA MEXICANA DE VALORES".

293018

TRABAJO DE SEMINARIO QUE PARA OBTENER EL TITULO DE LICENCIADA EN INFORMATICA PRESENTA: ALICIA CARRERA SILVA

ASESOR: M. en C. VALENTIN ROLDAN VAZQUEZ.

CUAUTITLAN IZCALLI, EDO. DE MEXICO. 2001.



Universidad Nacional  
Autónoma de México



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

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

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

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



FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN  
 UNIDAD DE LA ADMINISTRACION ESCOLAR  
 DEPARTAMENTO DE EXAMENES PROFESIONALES

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

DEPARTAMENTO DE EXAMENES PROFESIONALES

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

DR. JUAN ANTONIO MONTARAZ CRESPO  
 DIRECTOR DE LA FES CUAUTITLAN  
 PRESENTE

ATN: Q. Ma. del Carmen García Mijares  
 Jefe del Departamento de Exámenes Profesionales de la FES Cuautitlán

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario

Sistemas de Información: "Unificación institucional de las bases de datos

de la Bolsa Mexicana de Valores".

que presenta la pasante: Alicia Carrera Silva

con número de cuenta: 9656785-2 para obtener el título de

Licenciada en Informática

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXÁMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO

ATENTAMENTE  
 "POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán Izcalli, Méx. a 11 de Septiembre de 2001

MODULO	PROFESOR	FIRMA
I	M.C.C. Araceli Nivón Zaghi	
II	M. en C. Valentín Roldán Vázquez	
III	Inq. Miquel Álvarez Pasaye	

**A Dios**

Por haberme permitido vivir este momento y  
por llenar mi vida de bendiciones

**A mis papás**

Por todo el esfuerzo que pusieron en mi  
educación, por su comprensión y apoyo, pero  
sobre todo por su infinito amor.

**A mis hermanos**

Por todo su cariño y solidaridad

**A mi Paco**

Por su apoyo y amor incondicional, porque  
esta meta es el resultado de nuestros  
sacrificios y esfuerzos

**A mis amigos y amigas**

de la preparatoria y de mi querida  
Universidad, porque han formado parte  
importante de mi desarrollo personal y  
profesional.

**A la UNAM**

Por permitirme formar parte de su comunidad  
estudiantil; porque me esforzaré siempre en  
dejar su nombre muy en alto

## INDICE

<b>TEMA</b>	<b>PÁG.</b>
<b>Introducción.....</b>	<b>6</b>
<b>Objetivos.....</b>	<b>10</b>
Objetivo General.....	10
Objetivos Secundarios.....	11
<b>Hipótesis.....</b>	<b>12</b>
<b>Capítulo 1. Sistemas de Información.....</b>	<b>13</b>
1.1 Antecedentes históricos de la BMV.....	14
1.2 Sistemas de Información.....	20
Concepto.....	20
Elementos de un sistema de información.....	21
Principales funciones de un sistema de información.....	22
Ciclo de vida de un sistema de información.....	23
Tipos de sistemas de información.....	25
Usuarios de los sistemas de información.....	29
Evolución de los sistemas de información.....	30
1.3 La evolución de la informática en la BMV.....	34

<b>Capítulo 2. Bases de Datos</b> .....	36
2.1 Importancia de las bases de datos.....	37
2.2 Antecedentes de los sistemas de bases de datos.....	38
2.3 Sistemas de administración de bases de datos.....	39
Componentes principales de un sistema de bases de	41
Datos.....	
Clasificación de los sistemas de bases de datos.....	44
2.4 Modelo de Datos.....	46
Definición.....	46
Estructura del modelo de datos.....	47
Clasificación de los modelos de datos.....	50
2.5 Modelo entidad – relación.....	52
Entidad.....	54
Relación.....	56
Atributo.....	60
Representación del modelo entidad - relación.....	62
2.6 Modelo Relacional.....	64
Terminología Relacional.....	65
Álgebra Relacional.....	71
Reglas de las bases de datos relacionales.....	76
2.7 Modelo de datos orientados a objetos.....	78
2.8 Modelo Jerárquico.....	81
2.9 Modelo de red.....	83

2.10 Normalización de bases de datos relacionales.....	84
Primera forma normal.....	86
Segunda forma normal.....	87
Tercera forma normal.....	88
Forma normal Boyce-Codd (FNBC) .....	88
Cuarta forma normal.....	89
Quinta forma normal.....	90
2.11 Integridad de las bases de datos .....	91
2.12 Seguridad de las bases de datos.....	95
Autorizaciones y vistas.....	96
2.13 Lenguajes de manipulación de datos.....	99
SQL (Structured Query Language) .....	102
2.14 Actores en un sistema de bases de datos.....	113
Administrador de base de datos (DBA) .....	113
Diseñadores de la base de datos.....	114
Usuarios Finales.....	115
<b>Capítulo 3. Proceso de Creación de Bases de Datos.....</b>	<b>117</b>
3.1 Ciclo de vida de un sistema de información orientado a bases de datos.....	118
Fases del ciclo de vida de un sistema de información orientado a bases de datos.....	119
3.2 El proceso de Diseño de bases de datos. ....	120
Fases del diseño de bases de datos.....	121

Diseño del contenido de datos y de la estructura de la base de datos.....	123
Diseño del procesamiento de la base de datos y de las aplicaciones de software.....	124
3.3 Métodos de diseño de bases de datos.....	126
Concepto y fases de un método de diseño.....	126
<b>Capítulo 4. Caso Práctico. Unificación Institucional de las bases de datos de La BMV.....</b>	<b>137</b>
4.1 Planificación.....	138
Antecedentes.....	138
Objetivos del proyecto.....	139
Alcance y condiciones generales.....	139
Plan de proyecto.....	140
Equipos de trabajo.....	142
4.2 Recopilación de requisitos de usuario.....	142
4.3 Análisis y diseño de la base de datos unificada.....	143
Análisis funcional del sistema.....	144
Diseño de programas de aplicación.....	150
Implementación.....	152
4.4 Implementación, carga o conversión de datos y aplicaciones.....	153
4.5 Pruebas de validación.....	153
4.6 Operación.....	153
4.7 Supervisión y mantenimiento.....	153

<b>Conclusiones.....</b>	<b>154</b>
<b>Bibliografía.....</b>	<b>156</b>

## INTRODUCCIÓN

El presente trabajo de seminario desarrolla el proyecto de Unificación de Bases de Datos de la Bolsa Mexicana de Valores.

Partimos de la premisa de que la información es el activo más importante de cualquier empresa u organización; dentro de la Bolsa Mexicana de Valores (BMV) juega un papel estratégico para la toma de decisiones en el ámbito financiero y bursátil, tanto para las empresas del Centro Bursátil como para aquellas que cotizan y forman parte del Mercado de Valores.

Vivimos la era de la información, la forma en que una organización la administra y explota trae consecuencias directas en el desempeño competitivo que tenga en el mercado. De la misma forma la BMV se encuentra en un ámbito competitivo internacional en el que se debe contar con información clara, veraz y oportuna, para poder mantenerse firme en el mercado internacional.

Actualmente la BMV cuenta con una extensa gama de aplicaciones y procesos operativos, que proveen, procesan, generan y difunden información hacia otras áreas internas de la institución, así como a entidades externas. Todo el conjunto de dicha información se almacena en dos bases de datos principales, CONTEN y TANDEM, a las cuales se encuentran conectadas las aplicaciones y procesos que las generan o utilizan.

Dada la diversidad de tipos de información y a la compleja labor de administración de la misma, se han generado problemas en torno a la veracidad, consistencia y explotación de esta información.

En ambas plataformas se guarda casi la misma información provocando duplicidad e inconsistencia en los datos debido a que las bases de datos soportadas por estas dos plataformas cuentan con estructuras totalmente diferentes.

La base de datos soportada en Tandem es solo un contenedor de datos organizados en tablas, pero que no tiene relaciones establecidas entre ellas, la estructura de la base de datos y de las tablas no se encuentra normalizada y tampoco cuentan con ningún tipo de estándar de diseño. Las tablas fueron diseñadas para satisfacer necesidades muy específicas y no han sufrido modificaciones al mismo ritmo que las necesidades de la empresa, por lo que existen muchas entidades y atributos que ya no son utilizados y otros más que necesitan ser creados o redefinidos.

La base de datos soportada en SQL Server cuenta solo con una pequeña parte de la información que se genera en la BMV, contiene la información más importante como catálogos principales y algunas tablas históricas. El diseño de esta base de datos fue sistematizado y estandarizado, por lo que ésta cubre todos los requisitos de una base de datos eficiente y eficaz.

El objetivo principal de este trabajo de seminario es establecer las bases para el diseño de una base de datos unificada que le permita a la BMV llevar a cabo este proyecto para el beneficio de la organización y del mercado bursátil y de valores.

El presente trabajo se organiza en 4 capítulos:

En el capítulo No. 1 se hablará de los sistemas de información, sus conceptos y características principales, tipos, los diferentes niveles de organizaciones de acuerdo a su

evolución en sistemas de información, etc. Tomando como base la teoría estudiada en este capítulo, se ubicará a la BMV en un nivel de desarrollo de sistemas de información, esto con el objetivo de tener un objetivo claro de hacia dónde se quiere llegar con el desarrollo del proyecto y encaminar los esfuerzos a satisfacer las necesidades propias de una organización con esas características.

En el capítulo No. 2 se manejarán los conceptos y características principales de las Bases de Datos en general. Se estudiarán conceptos básicos de bases de datos, el modelo entidad-relación, y el modelo relacional. Se mencionarán las características generales del modelo de red, el jerárquico y el orientado a objetos. Dentro de este capítulo se tratará la normalización, integridad y seguridad de las bases de datos. En la parte final del capítulo se hablará de los lenguajes de manipulación de datos y de los actores que participan dentro de un sistema de base de datos, sus características y funciones principales.

En el capítulo No. 3 se tratará el proceso de creación de bases de datos, basado en el ciclo de vida de un sistema de información orientado a base de datos. Se estudiarán sus fases, el proceso de diseño de bases de datos en donde se estudiarán las dos vertientes del diseño - el diseño del contenido de datos y de la estructura de la base de datos, y el diseño del procesamiento de la base de datos y de las aplicaciones de software,

En este capítulo se definirá el proceso a seguir para el diseño de la unificación de las bases de datos de la BMV.

En el capítulo No. 4 se desarrolla el caso práctico que se titula: " Unificación Institucional de Base de Datos en la BMV.", basado en el proceso de creación de bases de datos estudiado en el capítulo III. El capítulo IV se definen las actividades a realizar

organizadas por fases para el diseño de la Base de Datos Unificada. En cada una de esas actividades se generan productos (que pueden ser documentos, diagramas, modelos, etc) resultado del análisis realizado.

Posteriormente se encuentran las conclusiones del propio trabajo de seminario y del proyecto específicamente, sección con la que se concluye este trabajo.

## **OBJETIVO GENERAL**

**Describir un proceso de unificación de bases de datos contenidas en dos plataformas diferentes.**

## OBJETIVOS SECUNDARIOS

- Estudiar los diferentes enfoques del diseño de sistemas de información orientado a bases de datos.
- Ubicar el nivel de la función informática en la BMV.
- Plantear un proceso de unificación de acuerdo al nivel de la función informática para que la base de datos diseñada satisfaga sus necesidades de información.
- Analizar el proceso de Unificación de las bases de datos en SQL Server y TANDEM de la Bolsa Mexicana de Valores.
- Ofrecer a la empresa una base de datos diseñada para soportar sistemas de información gerenciales y estratégicos y por lo tanto promover el desarrollo de los mismos.

## HIPÓTESIS

Si la BMV tiene una base de datos consistente, que aumente la seguridad, veracidad y el control en la administración de la información, entonces tendrá una participación más competitiva a nivel internacional proporcionando a los inversionistas nacionales y extranjeros solidez y confianza para la toma de decisiones que les permita incorporarse al mercado bursátil mexicano.

CAPÍTULO 1

SISTEMAS DE INFORMACIÓN

## 1.1 ANTECEDENTES HISTÓRICOS DE LA BOLSA MEXICANA DE VALORES

La Bolsa Mexicana de Valores (BMV) es el foro en el que se llevan a cabo las operaciones del mercado de valores organizado en México, cumple, entre otras, las siguientes funciones: proporcionar la infraestructura, la supervisión y los servicios necesarios para la realización de los procesos de emisión, colocación e intercambio de valores y títulos inscritos en el Registro Nacional de Valores e Intermediarios (RNVI), y de otros instrumentos financieros;

- Hacer pública la información bursátil;
- Realizar el manejo administrativo de las operaciones y transmitir la información respectiva a SD Indeval;
- Supervisar las actividades de las empresas emisoras y casas de bolsa, en cuanto al estricto apego a las disposiciones aplicables; y fomentar la expansión y competitividad del mercado de valores mexicano.

Las empresas que requieren recursos (dinero) para financiar su operación o proyectos de expansión, pueden obtenerlo a través del mercado bursátil, mediante la emisión de valores (acciones, obligaciones, papel comercial, etc.) que son puestos a disposición de los inversionistas (colocados) e intercambiados (comprados y vendidos) en la BMV, en un mercado de libre competencia y con igualdad de oportunidades para todos sus participantes.

La Bolsa Mexicana de Valores ha tenido una larga historia desde su constitución, las fechas más importantes de su evolución son las siguientes:

1850. - Negociación de primeros títulos accionarios de empresas mineras.

1867. - Se promulga la Ley Reglamentaria del Corretaje de Valores.

1880-1900. - Las calles de Plateros y Cadena, en el centro de la Ciudad de México, atestiguan reuniones en las que corredores y empresarios buscan realizar compraventas de todo tipo de bienes y valores en la vía pública. Posteriormente se van conformando grupos cerrados de accionistas y emisores, que se reúnen a negociar a puerta cerrada, en diferentes puntos de la ciudad.

1886. - Se constituye la Bolsa Mercantil de México.

1895. - Se inaugura en la calle de Plateros (hoy Madero) el centro de operaciones bursátiles Bolsa de México, S.A.

1908. - Luego de periodos de inactividad bursátil, provocados por crisis económicas y en los precios internacionales de los metales, se inaugura la Bolsa de Valores de México, SCL, en el Callejón de 5 de Mayo.

1920. - La Bolsa de Valores de México, S.C.L. adquiere un predio en Uruguay 68, que operará como sede bursátil hasta 1957.

1933. - Comienza la vida bursátil del México moderno. Se promulga la Ley Reglamentaria de Bolsas y se constituye la Bolsa de Valores de México, S.A., supervisada por la Comisión Nacional de Valores (hoy Comisión Nacional Bancaria y de Valores).

1975. - Entra en vigor la Ley del Mercado de Valores, y la Bolsa cambia su denominación a Bolsa Mexicana de Valores, e incorpora en su seno a las bolsas que operaban en Guadalajara y Monterrey.

1995. - Introducción del BMV-SENTRA Títulos de Deuda. La totalidad de este mercado es operado por este medio electrónico.

1996. - Inicio de operaciones de BMV-SENTRA Capitales.

1998. - Constitución de la empresa Servicios de Integración Financiera (SIF), para la operación del sistema de negociación de instrumentos del mercado de títulos de deuda (BMV-SENTRA Títulos de Deuda). También en este año surge Bursatec como una de las empresas más importantes de desarrollo de sistemas para el sector financiero.

11 de enero de 1999. - La totalidad de la negociación accionaria se incorporó al sistema electrónico. A partir de entonces, el mercado de capitales de la Bolsa opera completamente a través del sistema electrónico de negociación BMV-SENTRA Capitales.

En este año también se listaron los contratos de futuros sobre el IPC en MexDer y el principal indicador alcanzó un máximo histórico de 7,129.88 puntos el 30 de diciembre.

2001. - El día 17 de Mayo del 2001 se registró la jornada más activa en la historia de la Bolsa Mexicana de Valores estableciendo niveles récord de operatividad.

El número de operaciones ascendió a 11,031, cifra superior en 516 operaciones (+4.91%) respecto al nivel máximo anterior registrado el día tres de Marzo del 2000.

El desarrollo de la BMV es determinante para el crecimiento económico del país, esto implica que la BMV debe contar con tecnología de punta y con sistemas robustos que soporten la operación diaria e histórica, además de contar con la infraestructura adecuada para el desarrollo de sistemas de apoyo a las decisiones, determinantes dentro de la empresa, además de mantener a la BMV como una empresa sólida y atractiva de inversión nacional y extranjera, capaz de estar a la altura de las bolsas de valores internacionales más importantes.

La empresa encargada de los sistemas de información de la Bolsa Mexicana de Valores es Bursatec, S.A de C.V.

Bursatec nace con la necesidad de unificar esfuerzos de todos los departamentos de sistemas de las empresas que laboran en el centro bursátil, debido a la magnitud e importancia de los sistemas desarrollados por cada una de ellas; de esta forma en 1998 surge Bursatec como una subsidiaria de la Bolsa Mexicana de Valores y el S.D. Indeval. La meta principal de la empresa es brindar servicios y sistemas confiables de la más alta calidad para asegurar la continuidad operativa de los clientes a través del uso de tecnología avanzada, impulsando el crecimiento de los mercados financieros en un marco de competitividad internacional.

Bursatec diseña, desarrolla, implementa, integra y opera una gran variedad de sistemas de información y comunicación automatizados para el sector financiero, además de brindar otros servicios de valor agregado para sus clientes.

La empresa cuenta con una fábrica de software. La fábrica de software es un nuevo concepto que se está incorporando para facilitar el desarrollo de sistemas de información. Este nuevo concepto requiere de nuevas metodologías de desarrollo de sistemas de información. En Bursatec se está empezando a implementar Rational Unified Process como metodología para el desarrollo de sistemas, con el objetivo de explotar al máximo las utilidades y beneficios que trae la implementación de una fábrica de software.

Anteriormente, los sistemas que se desarrollaban dentro de la empresa carecían de un proceso específico y de estándares de documentación y desarrollo, tanto por parte de usuarios como por parte de los analistas y programadores, creando conflictos en el mantenimiento de los sistemas. Con Rational Unified Process se pretende tener cada una de las etapas del desarrollo de sistemas controladas con el fin de producir software de más alta calidad y por lo tanto un mantenimiento eficiente del mismo.

De esta forma, la empresa evoluciona en el desarrollo de sistemas. Debido a este desarrollo, se hace necesario contar con una base sólida para que los sistemas desarrollados puedan funcionar de la manera más eficiente posible. La base de datos que soporte a las aplicaciones de la Bolsa Mexicana de Valores debe ser lo más segura y confiable posible para terminar de reunir las piezas claves del desarrollo organizacional.

Actualmente la información que se genera en la Bolsa Mexicana de Valores es almacenada en dos bases de datos, una que se encuentra en la plataforma Compaq NonStop (Tandem) y otra que se encuentra en SQL Server (Conten).

En Tandem se almacena la información que se genera diariamente dentro de la Bolsa Mexicana de Valores, así como las tablas históricas correspondientes; en SQL Server se

almacena parte de la información histórica y los datos concernientes a los sistemas desarrollados en aplicaciones como Visual Basic, C++, etc. Esto implica que se esté duplicando información, debido a que muchos de los sistemas que utilizan la base de datos de SQL requieren información que se encuentra en Tandem, para lo cual se desarrollan interfaces que extraigan la información necesaria de Tandem y la inserten en la base de SQL Server.

El mantenimiento de la base de datos, el control y la seguridad de la información se vuelve de esta forma compleja y el mantenimiento del equipo Tandem resulta bastante costoso por la cantidad de información que almacena.

Con el crecimiento de la empresa en cuanto a sistemas de información, es indispensable que la base de datos que soporta las aplicaciones de la Bolsa Mexicana de Valores, sea lo suficientemente confiable en cuanto a integridad y seguridad de los datos, además de soportar aplicaciones innovadoras tales como sistemas de apoyo a la toma de decisiones y sistemas de información estratégica.

Para poder llegar a desarrollar este tipo de aplicaciones, se requiere tener una base de datos consistente que pueda soportar estos sistemas, este proyecto de seminario se centra en diseñar una base de datos para la Bolsa Mexicana de Valores, unificando las bases de datos existentes sobre Tandem y SQL Server que ponga las bases para el posterior desarrollo de sistemas de información gerenciales y estratégicos, con el objetivo de mantener a la empresa en un nivel internacional altamente competitivo.

## 1.2 SISTEMAS DE INFORMACIÓN.

Actualmente todas las empresas cuentan con al menos un sistema de información que les ayuda a operar con mayor rapidez y eficiencia. Los sistemas de información cambian la forma en que las organizaciones operan automatizando los procesos operativos de la empresa, proporcionan la información necesaria para la toma de decisiones y ayudan a las empresas a mantenerse en el mercado competitivo.

La habilidad para utilizar la información, más que la habilidad para manejar los recursos financieros, para obtener ventajas competitivas ya sea a través de nuevos productos y servicios o con un trato más eficaz hacia los clientes, proveedores y competidores es el factor decisivo del éxito de una empresa.

### CONCEPTO.

Para definir el concepto sistemas de información debemos partir del concepto de sistema y de información.

Un sistema es un conjunto de componentes que interactúan entre sí para lograr un objetivo común.

Información son datos que dentro de un contexto dado tienen un significado para alguna persona.

Daniel Cohen y Enrique Asin señalan que un *sistema de información* es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

Senn señala que un sistema de información es una entidad abstracta de la cual depende todo sistema organizacional, proporciona servicios a todos los demás sistemas de una organización y enlazan todos sus componentes en forma tal que éstos trabajen con eficiencia para alcanzar el mismo objetivo.

Tomando como base los conceptos de sistema y de información mencionados anteriormente, podemos definir que un sistema de información es un conjunto de componentes que interactúan entre sí, tomando como componentes a los usuarios, los datos y los procesos, para lograr un objetivo común que es el de generar y controlar la información de una organización, logrando una alta competitividad.

Los principales objetivos de los sistemas de información son:

- Automatizar procesos operativos dentro de las organizaciones.
- Proporcionar información que sirva de apoyo al proceso de la toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso.

#### ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

- *El equipo computacional.* Es el hardware es necesario para que el sistema de información pueda operar.
- *El recurso humano.* Son las personas que interactúan con el Sistema de Información, alimentándolo con datos o utilizando los resultados que genere.

- *Los datos o información fuente* que es introducida en el sistema.
- *Los programas* que son procesados y producen diferentes tipos de resultados.

#### PRINCIPALES FUNCIONES DE UN SISTEMA DE INFORMACIÓN.

Las finalidades de los sistemas de información son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas.

- *Entrada de Información.* La entrada es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información.
- *Almacenamiento de información.* El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sesión o proceso anterior. Esta información suele ser almacenada en archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o disquetes y los discos compactos (CD-ROM). Sin embargo, existen otras formas de almacenamiento.
- *Procesamiento de Información.* Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.
- *Salida de Información.* La salida es la capacidad de un Sistema sacar la información procesada o bien datos de entrada al exterior. Pueden ser impresoras, terminales, disquetes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. La salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo.

También existe un control básico que deben emplear los sistemas de información, el cuál consiste en:

- Un estándar para lograr un desempeño aceptable.
- Un método para medir el desempeño actual.
- Un medio para comparar el desempeño actual contra el estándar.
- Un método de retroalimentación.

#### CICLO DE VIDA DE LOS SISTEMAS DE INFORMACIÓN.

El ciclo de vida de un sistema de información es medular en las investigaciones de sistemas. Durante su desarrollo, cada sistema se mueve a través de varias fases de un ciclo de vida, después del cual sólo funciona por varios años con un mínimo mantenimiento. El sistema se deteriora gradualmente hasta el punto en que cesa de funcionar por completo y se comienza un nuevo ciclo de vida con el desarrollo de un nuevo sistema.

Algunos autores definen el ciclo de vida como el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. Sin embargo este concepto corresponde más al proceso de análisis y diseño de sistemas de información.

La figura muestra cinco fases. Éstas son la fase de estudio preliminar: la fase de análisis de sistemas, la fase de diseño de sistemas, la fase de implantación, la cual incluye una actividad separada llamada "auditoría posterior". Los ciclos de vida de sistemas varían en gran manera en términos de longitud, pero por lo regular el ciclo de vida de un sistema de información está en el rango de 3 a 8 años. Las primeras cuatro fases de este ciclo de vida pueden llamarse las "fases de investigación de sistemas".

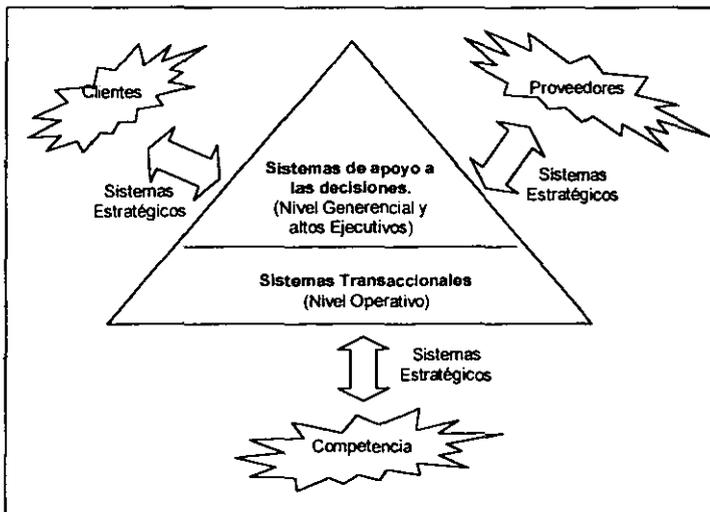
1. *Fase de estudio preliminar.* Durante esta fase, con un sistema de información existente se descubre un problema o una oportunidad de desarrollar útilmente un nuevo sistema, y se lleva a cabo una cantidad limitada de investigación preliminar para ver si un proyecto de sistemas está garantizado.
2. *Fase de análisis de sistemas.* Durante la fase de análisis, se identifica un problema u oportunidad asociada con el sistema, se examinan los puntos débiles y fuertes del sistema antiguo, y se determina para qué serviría un nuevo sistema.
3. *Fase de diseño de sistemas.* Durante esta fase se diseña un nuevo sistema o una aplicación computarizada para satisfacer las necesidades que se han determinado durante la fase de análisis. Asimismo se completara, tanto los estudios de hardware, como el diseño del software.
4. *Fase de implantación.* Esta fase involucra la programación, instalación de equipo, y otras actividades relacionadas con la implantación de un sistema diseñado.
5. *Fase de madurez y mantenimiento de sistemas.* Esta fase incluye la operación continua del sistema después de su instalación. Por lo general, el sistema alcanza su más alto desempeño, y después la efectividad de su costo declina gradualmente al cambiar su ambiente, al cambiar sus costos de operación, o al gastarse o convertirse en obsoleto su equipo. Cerca del final de esta fase, reconoce que el sistema no está funcionando satisfactoriamente y se reemplaza.

Una breve auditoría posterior es parte de la fase de madurez y mantenimiento de sistemas. Para determinar si la investigación se realizó con eficacia y para establecer hasta

qué punto la empresa ha recibido los beneficios esperados, un equipo de auditoría posterior revisa los procesos de investigación de sistemas, así como el funcionamiento del nuevo sistema.

### TIPOS DE SISTEMAS DE INFORMACIÓN.

Existen diferentes tipos de sistemas de información: los transaccionales, los sistemas de apoyo a las decisiones entre los que se encuentran los Sistemas de soporte a la Toma de Decisiones (DSS), Sistemas para la Toma de Decisiones en Grupo (GDSS), Sistemas Expertos de Soporte a la Toma de Decisiones (EDSS) y Sistemas de Información para el Ejecutivo (EIS) y los finalmente, los sistemas estratégicos.



**Fig. 1.1 Tipos de Sistemas de Información**

### *Sistemas Transaccionales*

- También conocidos como sistemas de procesamiento de transacciones (TPS) tienen como objetivo principal mejorar las actividades rutinarias de una empresa y de las que depende toda la organización
- Automatizan tareas operativas de la organización.
- Generalmente son el primer tipo de Sistemas de Información que se implanta en las organizaciones. Se empieza apoyando las tareas en el ámbito operativo de la organización para continuar con los mandos intermedios y posteriormente con la alta administración conforme evolucionan.
- Tienen muchas entradas y salidas de información; sus cálculos y procesos suelen ser simples y poco sofisticados, pero de gran volumen.
- Existe gran similitud entre las transacciones.
- Son recolectores de información, es decir, a través de estos sistemas se cargan las grandes bases de información para su explotación posterior.
- Brindan velocidad y exactitud.
- Manejan la mayor parte de la información que se maneja en la organización, la cual será utilizada posteriormente para apoyar a los mandos intermedios y altos.
- Sus beneficios son visibles por lo que el proceso de justificación puede realizarse enfrentando ingresos y costos. Esto se debe a que en el corto plazo se pueden evaluar los resultados y las ventajas que se derivan del uso de este tipo de sistemas.
- Son fácilmente adaptables a paquetes de aplicación que se encuentran en el mercado, ya que automatizan los procesos básicos que por lo general son similares o iguales en otras organizaciones.

### *Sistemas de Información Administrativa.*

- Los sistemas de información administrativa (MIS) ayudan a los directivos a tomar decisiones y a resolver problemas, obteniendo información generada del procesamiento de las transacciones, pero también empleando otro tipo de información.
- Auxilian en la toma de decisiones sobre asuntos que se presentan con regularidad (a la semana, mes, trimestre, etc.).
- En este tipo de decisiones, los procesos de decisión se encuentran claramente definidos, por lo que los sistemas de información administrativa se desarrollan de forma que periódicamente se generen reportes para el soporte de decisiones. Estos reportes se encuentran bien estructurados y contienen la información necesaria para las decisiones así como el estado de las variables importantes.
- Con frecuencia la información proporcionada por este tipo de sistemas de auxilia de información externa como tendencias económicas, demanda, costos, tasas de interés, etc.

### *Sistemas de Apoyo a las Decisiones*

- Suelen introducirse después de haber implantado los principales Sistemas Transaccionales debido a que son su plataforma de información.
- Este tipo de sistemas apoyan a las decisiones que se presentan una sola vez o escasamente (no recurrentes).
- Ayudan a los directivos a tomar decisiones no muy estructuradas
- La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información.

- No suelen ahorrar mano de obra. Debido a ello, la justificación económica para el desarrollo de estos sistemas es difícil, ya que no se conocen los ingresos del proyecto de inversión.
- Suelen ser Sistemas de Información interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.
- Estos sistemas pueden ser desarrollados directamente por el usuario final sin la participación operativa de los analistas y programadores del área de Informática.

### *Sistemas Estratégicos*

- Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones. Sin embargo, este tipo de sistemas puede llevar a cabo dichas funciones.
- Generalmente son desarrollados dentro de la misma empresa.
- Típicamente su forma de desarrollo es con base a incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van agregando nuevas funciones o procesos.
- Su función es lograr ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este contexto, los Sistemas Estratégicos son creadores de barreras de entrada al negocio.
- Apoyan el proceso de innovación de productos y procesos dentro de la empresa, debido a que buscan ventajas respecto a los competidores y una forma de hacerlo es innovando o creando productos y procesos.

Algunos autores consideran un cuarto tipo de Sistemas de Información denominado *Sistemas Personales de Información*, el cual está enfocado a incrementar la productividad de

sus usuarios. Dentro de esta clasificación se encuentran las hojas de cálculo, los sistemas de procesamiento de palabras, utilización de agendas, calendarios, etcétera. Sin embargo, estos son considerados generalmente como herramientas y no como sistemas.

#### USUARIOS DE LOS SISTEMAS DE INFORMACIÓN.

Los sistemas de información se desarrollan generalmente para automatizar tareas y apoyar a la toma de decisiones, sin embargo la eficiencia de éstos sistemas depende, además de los profesionales informáticos que los desarrollan, del buen uso que se haga de éstos. A las personas que los utilizan se les denomina usuarios y existe una gran variedad de ellos. Existen varias formas de clasificarlos, algunos los clasifican de acuerdo a la experiencia que tengan en el uso de recursos informáticos, sin embargo a continuación daremos una clasificación según el tipo de sistema del que se trate y la utilidad que le den al mismo.

*Usuarios Primarios.* Son los que interactúan con el sistema. Ingresan los datos de entrada o reciben salidas. Generalmente utilizan sistemas de información transaccionales.

*Usuarios Indirectos.* Son aquellos que se benefician de los resultados o reportes generados por estos sistemas pero que no interactúan de manera directa con el hardware o el software.

Sin embargo a pesar de haber diferentes tipos de usuarios, los sistemas deben ser utilizados con facilidad y de manera oportuna cuando sea necesario, aunque no se utilice constantemente. Los sistemas de información deben satisfacer las necesidades de todos los usuarios.

*Usuarios Gerentes.* Son lo que tienen responsabilidades administrativas en los sistemas de aplicación. Estos usuarios son gerentes de la empresa que utilizan en gran medida los sistemas de información, en este caso sistemas de información administrativa, por lo que los utilizan en gran medida como un soporte para la toma de decisiones rutinarias.

*Usuarios Directivos.* Este tipo de usuarios cada vez toma más conciencia de la importancia de los sistemas de información en términos de competencia. Las organizaciones bien dirigidas consideran el impacto y los beneficios de los sistemas de información al momento de elaborar las estrategias competitivas.

#### EVOLUCIÓN DE LOS SISTEMAS DE INFORMACIÓN.

En la década de los setenta, Richard Nolan, desarrolló su llamada Teoría de las Etapas, que influyó considerablemente en el proceso de planeación de los recursos y actividades de informática.

Nolan afirmaba que la función de la informática en las organizaciones evoluciona a través de ciertas etapas de crecimiento, las cuales son:

##### *Etapas de Inicio*

- Inicia con la adquisición de la primera computadora y se justifica por el ahorro de mano de obra y el exceso de papel.
- Las aplicaciones que se implantan son sistemas transaccionales, generalmente nómina y de contabilidad.

- La mayoría de las veces el área de informática o sistemas depende del área de contabilidad.
- La función de los sistemas suele ser manejada por una persona de Administración sin una preparación formal en el área de Informática.
- El personal que labora en sistemas es un operador y/o un programador.
- Existe una resistencia al cambio por parte del personal y los usuarios.
- Esta etapa finaliza con la implantación exitosa del primer sistema de información.

#### *Etapa de Contagio o expansión*

- Inicio con la implantación exitosa del primer sistema de información.
- Se termina de implantar todos los sistemas transaccionales de la empresa.
- Existe una proliferación de las aplicaciones de manera desordenada y sin control.
- El departamento de sistemas sube de categoría y puede depender de la gerencia administrativa o contraloría.
- La administración del área está orientado hacia la venta de aplicaciones a todos los usuarios de la organización.
- Se contrasta personal especializado y se crean los puestos de analista de sistemas, jefe de desarrollo, jefe de soporte técnico, etc.
- No existen interfaces automáticas entre las aplicaciones, por lo que las salidas de un sistema se alimentan de forma manual a otro sistema.
- No existen estándares de trabajo, lo que produce sistemas de baja calidad.
- Los gastos ocasionados por sistemas crecen de forma considerable, lo que provoca que se inicie la racionalización del uso de recursos informáticos. Este aspecto es el punto final de esta etapa.

### *Etapa de control o formalización*

- Esta etapa comienza con la necesidad de controlar el uso de recursos informáticos a través de técnicas de presupuestación.
- Las aplicaciones están orientadas a facilitar el control de las operaciones del negocio para otorgarles mayor eficiencia.
- El área de sistemas es ya una gerencia independiente.
- El tipo de administración dentro del área de sistemas se orienta hacia el control administrativo y la justificación económica de las aplicaciones a desarrollar.
- Se empiezan a definir estándares de trabajo dentro del departamento como estándares de documentación, control de proyectos, desarrollo y diseño de sistemas, auditoría de sistemas y programación.
- Se integra al departamento personal con habilidades administrativas y preparación técnica.
- Se empiezan a desarrollar interfaces automáticas entre los sistemas.
- Nace la función de la planeación de sistemas, incluyendo la planeación de requerimientos de cómputo y la planeación de adquisición de recursos computacionales.

### *Etapa de Integración*

- La integración de los datos y de los sistemas surge como una consecuencia de la centralización del departamento bajo una sola estructura administrativa.
- Esta integración de datos se produce con la inclusión de tecnologías como bases de datos, sistemas administradores de bases de datos, y lenguajes de cuarta generación.
- El departamento de sistemas evolucionó de forma descentralizada, lo cual permitió al usuario utilizar herramientas para el desarrollo de sistemas.

- Los usuarios y el departamento de sistemas inician el desarrollo de nuevos sistemas, reemplazando los sistemas antiguos, en beneficio de la organización.

#### *Etapa de administración de datos*

- El departamento de sistemas se encuentra consciente de la importancia de la información para la empresa y de la accesibilidad de ésta para los usuarios.
- Se hace necesario administrar de manera eficiente los datos para que los usuarios puedan utilizarlos y compartirlos.
- Los usuarios adquieren la responsabilidad de la integración de la información y se deben manejar niveles de acceso diferentes.

#### *Etapa de madurez*

- El departamento de sistemas dentro de la organización se ubica dentro de los primeros niveles del organigrama (dirección).
- Se desarrollan sistemas tales como sistemas de manufactura integrados por computadora, sistemas basados en el conocimiento y sistemas expertos, sistemas de apoyo a las decisiones, sistemas estratégicos y, en general, aplicaciones que proporcionan información para las decisiones de la alta administración y aplicaciones de carácter estratégico.
- Se introducen aplicaciones desarrolladas en la tecnología de bases de datos y se logra la integración de redes de comunicaciones con estaciones de trabajo en lugares remotos, a través del uso de recursos computacionales.
- Se establecen precios para los servicios informáticos y en algunos casos se define el área de informática como centro de utilidades en lugar de centro e costos. Nace la idea

de independizar el área de sistemas desde el punto de vista económico y organizacional (subcontratación).

- Suele existir una planeación rigurosa de los recursos de cómputo y las aplicaciones con horizontes de planeación no menor a cinco años.

### 1.3 LA EVOLUCIÓN INFORMÁTICA EN LA BMV

Sobre la base de la teoría de las etapas de Nolan definiremos la etapa en la que se encuentra la Bolsa Mexicana de Valores.

La empresa puede ubicarse en diferentes niveles, si tomamos en cuenta que la función informática es llevada a cabo mediante una empresa subsidiaria de la BMV (Bursatec), podremos decir que la BMV se encuentra en una etapa de madurez, sin embargo en lo que se refiere a manejo de información, no es así. Por lo tanto, tomando como base la forma en que es manejada la información dentro de la empresa, podremos ubicar a la BMV en la etapa de administración de datos; esto, tomando en cuenta lo siguiente:

- La empresa ya cuenta con toda la infraestructura en equipos de cómputo, como computadoras personales, servidores de bases de datos, servidores de aplicaciones, etc.
- Ya se encuentran implementados diversos sistemas transaccionales que controlan la operación diaria de la BMV.
- Ya fueron establecidos los primeros estándares a seguir tanto en documentación como dentro del desarrollo de sistemas.
- El personal que labora en el área es personal especializado en programación, análisis y diseño de sistemas y administración de proyectos. Este personal se ha

ido formando dentro de la empresa y se ha especializado según las funciones que desempeña.

- Existen interfaces automáticas entre los diversos sistemas y plataformas.
- El desarrollo de sistemas y el mantenimiento a los mismos se realizan mediante la planeación de proyectos.
- Se han implementado las nuevas tecnologías concernientes a bases de datos y a sistemas administradores de bases de datos, así como al desarrollo de sistemas mediante lenguajes de cuarta generación.

Actualmente surge la necesidad de administrar de mejor forma la información generada estableciendo las bases para el desarrollo de aplicaciones basadas en conocimiento, sistemas expertos, estratégicos y de toma de decisiones. Esta necesidad surge debido al crecimiento de la empresa y a la alta competitividad de los mercados internacionales en cuestión de sistemas. Por lo que se requiere de tecnología y sistemas capaces de mantener a la BMV en un alto nivel competitivo.

Por todas estas causas, se planteó el proyecto de Unificación de las Bases de Datos, para diseñar una base de datos confiable en cuanto a integridad y seguridad de la información, que proporcione las condiciones propicias para implementar sistemas estratégicos que impulsen el desarrollo de la empresa.

## CAPÍTULO 2

### BASES DE DATOS

## 2.1 IMPORTANCIA DE LAS BASES DE DATOS.

Las bases de datos son los pilares básicos que dan soporte a cualquier arquitectura de software. Representan el núcleo de cualquier organización, sus datos informan de todos los hechos que la envuelven y de ahí se toman las decisiones más relevantes y trascendentes.

Una base de datos bien diseñada es una garantía del resultado de las aplicaciones, un aval de su futuro mantenimiento y una tranquilidad de las posibles variaciones que puedan darse en los datos con las implicaciones que ello lleva consigo.

Las grandes organizaciones suelen utilizar dos bases de datos: una base de datos operacional y otra de apoyo a la toma de decisiones. Una base de datos operacional contiene la información necesaria para la gestión diaria de la organización. La base de datos de apoyo a la toma de decisiones contiene información resumida, que se extrae periódicamente de la base de datos operacional y que ayudan a los directivos de la empresa a la hora de tomar decisiones estratégicas y/o tácticas.

Dentro de este contexto, el presente proyecto, busca diseñar una base de datos que tenga la estructura y los requerimientos necesarios para que en un futuro sea una base de datos de apoyo a las decisiones. Este proceso es paulatino debido a que primero se tienen que estandarizar los datos almacenados en las dos bases de datos para formar una sola que sea totalmente confiable en cuanto a seguridad e integridad de los datos, y que empiece el proceso de creación de una base de datos de apoyo a las decisiones, guardando la información histórica generada. Los datos de las transacciones diarias se seguirán almacenando en la base de datos de Tandem como se ha venido manejando hasta ahora.

## 2.2 ANTECEDENTES DE LOS SISTEMAS DE BASES DE DATOS.

La información es el elemento más importante dentro de la administración y organización de una empresa. Antes de que las computadoras fueran una herramienta para controlar la información, las empresas almacenaban todos sus datos en papel; con la invención de las computadoras, el procesamiento de la información sufrió cambios radicales.

En los años 60's las empresas comenzaron a computarizar sus sistemas de información; los datos se guardaban en medios electrónicos y se utilizaban lenguajes de alto nivel para recuperar y manejar los datos en aditamentos de almacenamiento.

En los sistemas de información convencionales, las aplicaciones individuales se desarrollaban independientemente, y cada programa de aplicación procesaba sus propios archivos privados. Como resultado, algunas actividades se duplicaban y la información redundante se almacenaba para usarla en distintas operaciones. De esta manera fueron surgiendo los primeros sistemas de archivos que guardaban datos por periodos largos y permitían almacenar grandes cantidades de datos.

En los sistemas de archivos la redundancia de los datos originaba muchos problemas concernientes a la integridad de los datos. La inconsistencia de la información se produce más fácilmente cuando los datos son almacenados en más de un lugar mientras se realizan operaciones de inserción actualización y borrado. Los sistemas de archivos tenían una muy limitada capacidad de compartir datos, éstos se implementaban como unidades separadas haciendo casi imposible poder ser utilizados por múltiples aplicaciones. Éstos modelos estaban basados en punteros físicos por los que tenía que navegar el programador a fin de recuperar y actualizar los datos.

Al final de los años 60 los sistemas de administración de bases de datos imperantes en el mercado, de tipo Red y Jerárquico, no habían logrado superar el inconveniente de la dependencia de las aplicaciones desarrolladas en ellos respecto a las estructuras de los datos. En el año de 1970 Ted Codd propuso en su famoso trabajo "A relational model for large shared data banks", aislar al usuario de las estructuras físicas de los datos, consiguiendo así la independencia en las aplicaciones respecto de los datos, finalidad perseguida desde los inicios de las bases de datos.

### **2.3 SISTEMAS DE ADMINISTRACIÓN DE BASE DE DATOS.**

**Base de Datos:** Una base de datos es una colección de datos interrelacionados. Una base de datos permite que la definición de los datos y las relaciones entre esos datos sean totalmente independientes de las aplicaciones.

**Sistemas de Bases de Datos (DBMS):** Un sistema de administración de bases de datos es una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. Las características principales de los DBMS son las siguientes:

1. Minimiza la redundancia de los datos.
2. Incrementa la consistencia de la información.
3. Facilita el acceso a los datos
4. Aumenta la seguridad de la información.
5. Satisface ciertas restricciones de consistencia proporcionando integridad a los datos.
6. La base de datos puede ser compartida entre usuarios de distintas aplicaciones.
7. Respaldo y recuperación.

El DBMS interpreta y procesa las peticiones del usuario para extraer información de la base de datos. El DBMS permite a los usuarios el acceso a los datos sin necesidad de conocer su organización dentro del dispositivo de almacenamiento.

Entre las principales funciones de los DBMS se encuentran:

1. Crear y organizar la base de datos.
2. Establecer y mantener las trayectorias de acceso a la base de datos, de tal manera que los datos puedan ser accedidos fácilmente.
3. Insertar, modificar, borrar y buscar eficazmente datos específicos
4. Ordenar, reordenar y convertir datos.
5. Mantener la integridad y seguridad de los datos.
6. Registrar el uso de las bases de datos.

Algunas ventajas de las bases de datos son las siguientes:

1. Datos compartidos actualizados.
2. Flexibilidad
3. Rápida creación de nuevas aplicaciones
4. Cumplimiento de reglas o normas de la empresa
5. Proporcionar independencia de datos.

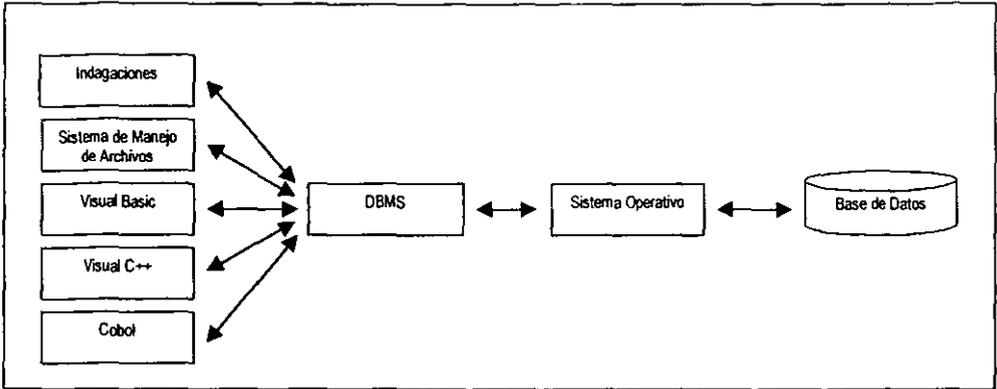


Figura 2.1 El DBMS como interface entre la base de datos física y las peticiones del usuario.

Algunos DBMS distribuidos comercialmente son: Oracle, SQL Server, Informix y Sybase, a los cuales haremos referencia dentro del proyecto más adelante.

COMPONENTES PRINCIPALES DE UN SISTEMA DE BASE DE DATOS.

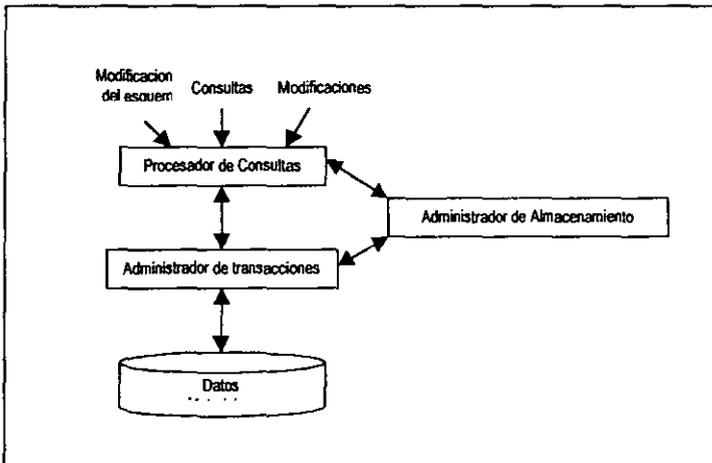


Figura 2.2 Principales componentes de un sistema de administración de bases de datos.

En la figura 2.2 se muestra la manera en la que está estructurado el administrador de bases de datos y la forma en la que sus elementos interactúan entre sí.

En la parte inferior de la figura se encuentra el *lugar para almacenar* los datos y los metadatos. Los metadatos es la información referente a la estructura de los datos; los metadatos contendrán los nombres de las relaciones, los nombres de los atributos de éstas y los tipos de datos de los atributos.

La función del *administrador (manejador) de almacenamiento* es obtener del almacenamiento de datos la información solicitada y modificarla cuando se lo pidan el procesador de consultas y el administrador de transacciones.

El administrador de almacenamiento controla el almacenamiento en disco. Consta de dos componentes:

1. El administrador de archivos que lleva el control de la localización de los archivos en el disco y, a solicitud del administrador del búfer, obtiene el bloque (o bloques) que contiene uno de ellos.
2. El administrador de búfer se encarga de la memoria principal. Obtiene bloques de datos del disco a través del administrador de archivos y selecciona una página de la memoria principal donde guardarlos

El *procesador de consultas*, no solo maneja consultas, sino también las peticiones de modificaciones de datos o los metadatos. Su función consiste en encontrar la mejor manera de llevar a cabo una operación solicitada y emitir comandos al administrador de almacenamiento que los ejecutará. La parte más difícil del procesamiento de consultas es la

optimización de la consulta, esto es, seleccionar un buen plan de consulta o sea una serie de peticiones al sistema de almacenamiento que las atenderá.

El *administrador de transacciones* es el componente encargado de conservar la integridad del sistema. Debe asegurarse de que las consultas que se ejecutan al mismo tiempo no interfieran entre ellas y de que el sistema no pierda información cuando ocurra alguna falla. Interactúa con el administrador de consultas, pues debe saber sobre qué datos están operando las consultas actuales (con el fin de impedir acciones conflictivas) y posiblemente necesite posponer ciertas consultas u operaciones para evitar esos conflictos. Interactúa con el administrador de almacenamiento porque los esquemas de protección de información suelen requerir guardar una bitácora de los cambios efectuados en ella.

El administrador de transacciones permite al usuario agrupar una o más consultas, modificaciones, o ambas en una transacción. El administrador de transacciones se encarga de que cada una de esas operaciones se realicen cumpliendo las propiedades ACID, abreviatura de los cuatro requisitos principales de la ejecución de transacciones (Atomicity: atomicidad; Consistency: consistencia; Isolation: aislamiento; Durability: durabilidad).

- Atomicidad. Deberá realizarse una operación completa o nada de ella.
- Consistencia. La información debe cumplir con normas de consistencia definidas por el usuario en respuesta a las necesidades. La información debe ser coherente a los hechos que describe.
- Aislamiento. Cuando dos o más transacciones son ejecutadas al mismo tiempo, es necesario aislar sus efectos.
- Durabilidad. Si la transacción ha cumplido su función, no debería perderse en caso de una falla de sistema.

En la parte superior de la figura 2.2 hay tres clases de entradas al sistema de administración de datos (DBMS):

1. Consultas: Son las preguntas concernientes a los datos. Se generan de dos maneras:
  - a) Mediante una interfaz de consultas genéricas.
  - b) Mediante interfaces de programas de aplicación.
2. Modificaciones. Estas operaciones modifican los datos, mediante la interfaz genérica de consultas o de la interfaz de un programa de aplicación.
3. Modificaciones al esquema. Estos comandos suelen ser emitidos por personal autorizado para cambiar el esquema de la base o crear otra nueva.

#### CLASIFICACIÓN DE LOS DBMS

Los DBMS pueden tener varias clasificaciones, algunas de las cuales son las siguientes:

- a) Según el modelo de datos en el que está basado.
  1. Relacional
  2. En red
  3. Jerárquico
  4. Orientado a Objetos
  5. Otros: Objeto-Relacional

- b) Según el número de usuarios a lo que da servicio simultáneamente.
1. Monousuario
  2. Multiusuario
- c) Según el número de lugares en el que se almacenan los datos.
1. Centralizado. Las bases de datos centralizadas son aquellas en las que los datos residen en un único ordenador. Este tipo de bases de datos son las más comunes actualmente, de hecho el proyecto que se trata en este trabajo de seminario es sobre una base de datos centralizada.
  2. Distribuido: La base de datos y el software del DBMS pueden estar repartidos en varios sitios conectados en red. La base de datos distribuida (BDD) es una colección de datos que pertenecen lógicamente al mismo sistema, pero físicamente está dispersa en varios sitios de una red de ordenadores.
    - 1.1 Distribuido homogéneo. Utiliza el mismo software DBMS en todos los sitios.
    - 1.2 Distribuido heterogéneo. Cada sitio puede tener un software de DBMS distinto. Este es DBMS híbrido entre centralizado y distribuido. Por un lado, los usuarios de cierto sistema de base de datos pueden acceder a éste de forma local, como si fuera un sistema centralizado. Por otro lado, el sistema visto globalmente es un sistema distribuido, puesto que un usuario puede acceder a los datos almacenados en cualquier otro sitio.

d) Según su propósito.

1. Propósito General. Cualquier aplicación puede comunicarse con él para acceder a la información de la base de datos.
2. Propósito específico. Construido para un tipo determinado de aplicaciones cuyo rendimiento es muy importante.

## **2.4 MODELOS DE DATOS**

La característica más importante del enfoque de bases de datos es proporcionar al usuario una visión abstracta de los datos, es decir, ocultarle detalles de almacenamiento y mantenimiento de los datos, que no necesita conocer.

Para conseguir esa abstracción de los datos se utilizan los modelos de datos.

### **DEFINICIÓN.**

Un modelo de datos se utiliza para describir la estructura de una base de datos, es decir, los tipos de datos, las relaciones entre ellos y las restricciones que deben cumplir.

## ESTRUCTURA DEL MODELO DE DATOS.

Los modelos de datos cuentan con dos partes:

### *Parte Estática:*

Esta se define mediante el esquema, con el lenguaje de definición de datos (DDL). El esquema de la base de datos son los metadatos, el cual se especifica en el diseño y rara vez es modificado.

El esquema se encuentra formado por estructura y restricciones.

- a) Estructura: La estructura queda definida por los objetos del modelo.
- b) Restricciones: Conjunto de reglas de definición de las estructuras.

Los objetos y las restricciones generalmente son: entidades, atributos, dominios, relaciones, representación y restricciones inherentes.

Las *restricciones inherentes* vienen impuestas por la propia naturaleza del modelo. Las *restricciones opcionales o de usuario* son definidas por el usuario, pero el modelo de datos las reconoce y proporciona las herramientas para manejarlas. Las *restricciones libres de usuario* son responsabilidad del usuario y el modelo ni las reconoce ni las maneja.

La estructura de una base de datos debe ser diseñada para recuperar los datos de forma eficiente; debido a esto, la complejidad de la estructura se oculta a los usuarios a través de niveles de abstracción con el fin de simplificar la interacción con la base de datos.

La arquitectura de tres esquemas para sistemas de bases de datos ayuda al cumplimiento de los objetivos del enfoque de bases de datos: la separación entre los

programas y los datos y el manejo de múltiples vistas de usuarios. Esta arquitectura es la arquitectura *ANSI/X3/SPARC*.

El objetivo de esta arquitectura es separar las aplicaciones del usuario de la base de datos física. Los esquemas pueden ser definidos en tres niveles:

- a) *Nivel Interno*. Es el nivel más bajo de abstracción. Tiene un esquema interno (EI) que describe cómo se almacenan realmente los datos, utilizando un modelo físico de datos, y muestra detalles de la organización física de los ficheros (estructuras físicas de almacenamiento, orden de secuencia de los registros físicos, tamaño de página, tamaño de bloque, etc.) y caminos de acceso (tipos de índice, etc.).
- b) *Nivel Conceptual (EC)*. Describe la estructura de toda la base de datos para el conjunto de usuarios. El EC oculta los detalles físicos y describe qué datos se almacenan en la base de datos y qué relaciones existen entre ellos (entidades, relaciones, tipos de datos, interrelaciones, restricciones, etc.). En este nivel puede utilizarse un modelo de datos lógico o uno de alto nivel para describir el esquema conceptual.
- c) *Nivel externo o de vistas*. Es el nivel más alto de abstracción y describe solo parte de la base de datos. Este nivel está compuesto por varios esquemas externos (EE) o vistas de usuario. Cada EE describe la parte de la base de datos que interesa a un grupo de usuarios determinado, ocultándoles el resto de la base de datos. Para este nivel puede utilizarse un modelo de datos de alto nivel o uno lógico para describir cada esquema externo.

Estos tres esquemas son descripciones de datos, los datos reales solo se encuentran en el nivel físico.

Dentro de los niveles de abstracción de los datos, existe un concepto que se debe de tomar en cuenta para darle una ventaja más a este tipo de abstracción de datos: la independencia de los datos.

La independencia de los datos es la capacidad para modificar el esquema en un nivel de sistema de bases de datos sin tener que modificar el nivel inmediato superior. Existen dos tipos de independencia de datos:

- a) *Independencia lógica con respecto a los datos.* Que es la capacidad de modificar el esquema conceptual (su estructura) sin alterar los esquemas externos (lo que ven los usuarios) ni los programas de aplicación.
- b) *Independencia física de los datos.* Es la capacidad de modificar el esquema interno sin alterar el esquema conceptual (o los esquemas externos), ni los programas de aplicación.

### *Parte Dinámica.*

Es un conjunto de operaciones con el lenguaje de manipulación de datos (DML).

Las operaciones sobre un modelo de datos pueden ser:

- a) Selección. Localización de los datos buscados.
- b) Acción: Realización de una acción sobre los datos seleccionados, ésta acción puede ser:
  - 1. Recuperación. Obtención de los datos seleccionados.
  - 2. Actualización. Que puede ser:

## 2.1 Modificación

## 2.2 Inserción.

## 2.3 Borrado

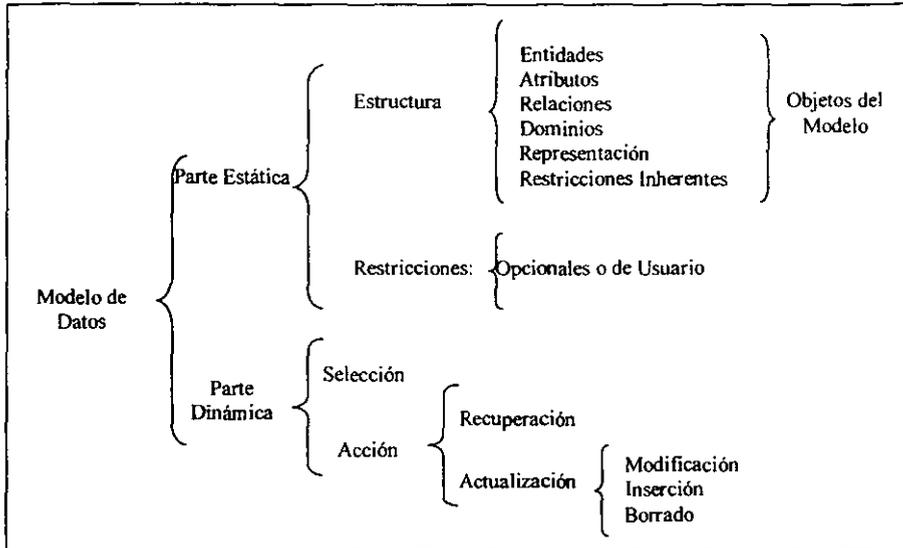


Figura 2.3 Estructura del Modelo de Datos.

### CLASIFICACIÓN DE LOS MODELOS DE DATOS.

Los modelos de datos se aplican a tres niveles: externo, conceptual e interno. Los correspondientes a los dos primeros son los Modelos de Datos Lógicos y los correspondientes al último son los Modelos de Datos Físicos.

Los modelos lógicos a su vez se clasifican en:

- Modelos lógicos basados en objetos
- Modelos lógicos basados en registros.

### *Modelos lógicos basados en objetos.*

Los modelos lógicos basados en objetos se utilizan para describir los datos en los niveles conceptual y externo. Constan de conceptos muy cercanos al modo en que el usuario percibe la realidad.

Se caracterizan por ser muy flexibles y hacer posible la especificación de las limitaciones de los datos.

Algunos de estos modelos son:

- Modelo entidad-relación
- Modelo Binario
- Modelo semántico de datos
- Modelo infológico.

### *Modelos lógicos basados en registros.*

Se utilizan para describir los datos a nivel conceptual y externo al igual que los modelos anteriores.

Estos modelos sirven para especificar, tanto la estructura lógica general de la base de datos como una descripción en un nivel más alto de la implementación, pero no permiten especificar de forma clara las limitaciones de los datos.

Proporciona conceptos que pueden ser entendidos por los usuarios finales, aunque no están muy alejados de cómo los datos se organizan físicamente.

Modelos de este tipo son:

- Modelo relacional.
- Modelo de datos orientados a objetos.
- Modelo jerárquico.
- Modelo en red.

Dentro de este trabajo de seminario se mencionarán las características principales del modelo entidad-relación, en el caso de los modelos lógicos basados en objetos y los modelos relacional, orientado a objetos, jerárquico y en red en el caso de los modelos lógicos basados en registros. Sin embargo se estudiarán más a fondo el modelo entidad-relación y el modelo relacional debido a que son con los que se va a trabajar en la unificación de bases de datos de la BMV.

## **2.5 MODELO ENTIDAD RELACIÓN**

El proyecto Unificación de Bases de Datos Institucional, que concierne a este trabajo de seminario, tiene por objetivo principal el diseño de una base de datos unificada, para llegar a tal fin, es necesario conocer las diversas técnicas de diseño de bases de datos, las consideraciones importantes a seguir al momento de diseñar la estructura de la base de datos, las restricciones y los lineamientos que nos ayudan a obtener como resultado una estructura sólida y consistente.

Empezaremos por definir lo que es el modelo entidad-relación, sus características principales y la forma de realizarlo. Este modelo no se usará como parte del proyecto, debido a que éste se utiliza para realizar la abstracción de los datos dentro de una organización (datos del mundo real), ésta abstracción ya fue hecha, al diseñar las bases de datos

existentes en la Bolsa Mexicana de Valores; sin embargo es importante saber a qué se refiere el modelo entidad-relación, para entender la estructura de las bases de datos, además de servir como antecedente al momento de diseñar una nueva tabla.

El modelo entidad-relación, que también es llamado modelo conceptual, es una técnica especial de representación gráfica que incorpora información relativa a los datos y la relación existente entre ellos, para darnos una visión del mundo real.

Es un paso previo al diseño de bases de datos en el modelo de datos que sea designado; relacional, jerárquico o red.

Las características del modelo entidad-relación son:

- Refleja solo la existencia de los datos, no lo que se hace con ellos.
- Se incluyen todos los datos del sistema en estudio y, por lo tanto no están orientados a aplicaciones particulares.
- Es independiente de las bases de datos y sistemas operativos concretos.
- No tienen en cuenta restricciones de espacio, almacenamiento y tiempo de ejecución.
- Está abierto a la evolución del sistema.

El modelo entidad-relación, se basa en la percepción de un mundo real que consiste en un conjunto de objetos básicos denominados Entidades, así como las relaciones existentes entre ellos.

Existen tres clases de objetos dentro de un modelo entidad-relación: Entidades, relaciones y atributos.

## ENTIDAD.

*Entidad.* Una entidad es una cosa u objeto concreto o abstracto que existe, que puede distinguirse de otros y del cuál se desea almacenar información. Pueden existir *conjuntos de entidades* que representan a un conjunto de entidades del mismo tipo.

Puede haber dos tipos de entidades:

- Entidades fuertes o propias
- Entidades débiles o regulares

Las entidades fuertes son aquellas cuyas ocurrencias son identificables por sí mismas. Este tipo de entidades se representan mediante un rectángulo.



Las entidades débiles son aquellas cuyas ocurrencias son identificables solamente por estar asociadas a otra u otras entidades. Es decir, que alguno de los atributos que las identifican se refiere a otra entidad. Éste tipo de entidades se representan mediante dos rectángulos.



Por ejemplo, los medios de comunicación de los funcionarios es una entidad débil debido a que depende de la entidad de funcionarios y de la entidad de medios de comunicación. La información por sí solo no representa nada.

A la instancia individual de una entidad se le conoce como instancia de entidad u ocurrencia de entidad. Por ejemplo, hay muchas emisoras representadas por la entidad emisoras, una instancia u ocurrencia de la entidad sería una emisora específica de la entidad.

*Papel ("rol")*. Es la función que cada una de las entidades realiza en la relación; se representa poniendo el nombre del papel en el arco que une cada entidad. Cuando no exista ambigüedad se puede omitir el papel.

*Clave de una entidad*. Existen diversas claves que identifican a una entidad, son las Superclaves, las claves candidatas, las claves foráneas y las claves primarias.

Una *Superclave* es un conjunto de uno o más atributos que, considerados conjuntamente, nos permiten identificar de forma única a una entidad en el conjunto de entidades.

Cuando en una Superclave ningún subconjunto propio es una Superclave, se le llama *clave candidata*.

La *clave primaria* es una clave candidata que elige el diseñador de la base de datos como el medio principal de identificar entidades dentro de un conjunto de entidades.

La *clave foránea* es aquel o aquellos atributos de una Entidad que son clave primaria en otra.

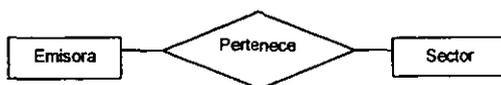
*Existencia.* Existe dependencia en existencia cuando las instancias de una entidad (entidad débil) no pueden existir si desaparecen las instancias de la entidad fuerte de la cuál dependen.

Existe dependencia en identificación cuando además de cumplirse la dependencia en existencia las instancias de la entidad débil no se pueden identificar por sí mismas y exigen añadir el identificador principal de la entidad fuerte de la cual dependen.

Dentro de una relación, la existencia de una entidad es *obligatoria* si la instancia de la entidad siempre debe existir en la relación, la existencia de una entidad es *opcional* si la instancia de la entidad puede no existir en la relación.

#### RELACIÓN.

Una relación en una asociación, sin existencia propia, de varias Entidades. Se representa mediante un rombo. Una restricción del modelo entidad-relación es que no se pueden expresar relaciones entre relaciones.



Un *conjunto de relaciones* es la agrupación de todas las relaciones existentes en un conjunto de entidades.

La *dimensión de una relación* es el número de entidades que participan en ella. La dimensión más generalizada es la binaria y es aquella en la que intervienen o se relacionan tan solo dos entidades.

*Relaciones Recursivas.* Una entidad puede estar relacionada a ella misma. Una relación recursiva es una asociación entre ocurrencias de la misma entidad.

Este tipo de relaciones no corren frecuentemente, pero son muy útiles para definir estructuras organizacionales.

*Clave de una relación.* La clave de una relación es el producto cartesiano o concatenación de las claves primarias de las entidades que asocia, y en ocasiones, alguno de los atributos que forman parte de dicha relación.

En el ejemplo de relación anterior, la clave de la relación sería la clave formada por la concatenación de la clave que identifica a la emisora y la clave que identifica al sector.

Una relación puede tener atributos, cuando esto sucede esa relación será una futura entidad y se le llama *entidad asociativa*. Se representa mediante un rectángulo y un rombo.



Una relación entre entidades se describe en términos de conectividad o correspondencia, cardinalidad y existencia.

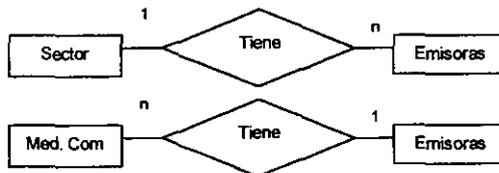
Grado de una relación (conectividad, correspondencia). El grado de una relación (también llamado correspondencia o conectividad, por algunos autores) representa la participación en la relación de cada una de las entidades afectadas y existen tres tipos posibles:

- 1:1 (una a una). A cada ocurrencia de una entidad corresponde no más de una ocurrencia de la otra.
- 1:n (una a muchas). A cada ocurrencia de la primera entidad pueden corresponderle varias ocurrencias de la segunda y a cada ocurrencia de la segunda corresponde no más de una de la primera.
- n:m (muchas a muchas). A cada ocurrencia de la primera entidad pueden corresponderle más de una ocurrencia de la segunda y viceversa.

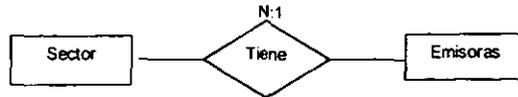
El grado de las relaciones también es representado en el diagrama entidad-relación.

Se puede representar de varias formas

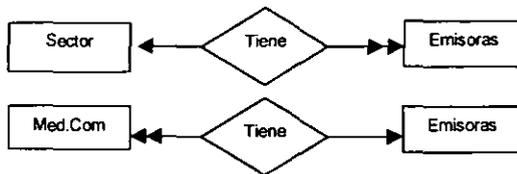
1. Indicando el grado en los segmentos que asocian las entidades de la relación. Se hace que el valor del grado se encuentre en la misma orientación de los segmentos.



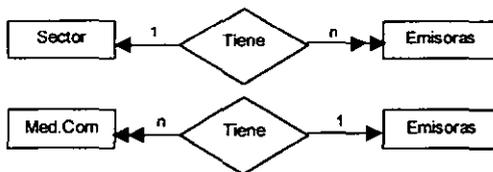
2. Indicando el grado en la relación, por ejemplo, en la parte izquierda el grado de la entidad de la izquierda y en la derecha el grado de la entidad de la derecha.



3. Incluyendo en los segmentos que asocian las entidades una flecha para indicar el grado 1, y dos flechas contiguas para mostrar el grado n ó m.



4. También se pueden utilizar dos sistemas combinados sistemas:



*Cardinalidad de una entidad.* La cardinalidad de una entidad en una relación, mide el máximo y el mínimo de la participación de dicha entidad en la relación.

La cardinalidad define cualquier restricción en el número de instancias u ocurrencias de entidades que están relacionados por medio de una relación.

Generalmente la notación para representar la cardinalidad es la siguiente: (0,1), (1,1), (0,N), (1,N), según corresponda



*Redundancia.* Las relaciones redundantes son dos o más relaciones que son utilizadas para representar el mismo concepto. Se deben analizar las relaciones redundantes cuidadosamente y eliminarlas del modelo.

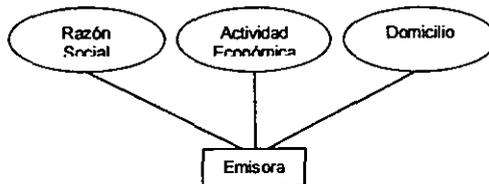
Se pueden permitir dos o más relaciones entre entidades siempre y cuando estas relaciones tengan significados diferentes.

Algunas de las razones por las que hay que eliminar las relaciones redundantes son las siguientes:

- a) Causan complejidad en el modelo.
- b) Pueden direccionar a un atributo a un lugar erróneo.
- c) Pueden ser mal interpretados causando errores en la manipulación de los datos.

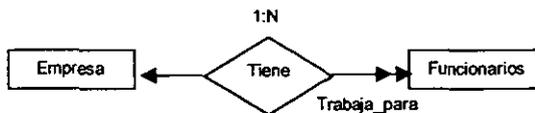
#### ATRIBUTO.

Un atributo es una entidad básica e indivisible de información acerca de una entidad o de una relación, que sirve para identificarla o describirla. Se representa mediante una elipse.



**Ocurrencia.** Se pueden diferenciar tres tipos de ocurrencias:

- Ocurrencia de un atributo: Son el conjunto de valores que puede tomar este atributo.
- Ocurrencia de una entidad: Es el conjunto de ocurrencias de sus atributos con sentido y existencia propia.
- Ocurrencia de una relación. Se encuentra constituida por:
  - a) Una y solo una ocurrencia de cada una de las entidades asociadas por la relación.
  - b) Las ocurrencias de cada uno de los atributos correspondientes a la relación.



**Atributos Identificadores y atributos descriptivos.** Existen dos clasificaciones generales de los atributos: *identificadores* y *descriptivos*. Un identificador es también conocido como llave.

Un atributo identificador es un atributo que identifica una característica única de una entidad en particular.

Un atributo descriptivo es un atributo que especifica una característica de una entidad en particular pero que no necesariamente es única.

**Datos Derivados.** Generalmente son valores calculados de otros atributos, pueden ocasionar redundancia en los datos y complejidad del modelo. Pueden ser bastante útiles

para algunos usuarios específicos; por lo que de acuerdo al tipo de base de datos (operacional o de decisiones) se evaluará la posibilidad de almacenarlos o generarlos a partir de las aplicaciones.

De esta forma, dependiendo del tipo de base de datos que se esté diseñando se tomará la decisión de almacenarse o no. En una base de datos operacional puede resultar contradictorio almacenar dichos datos, sin embargo en una base de datos de apoyo a la toma de decisiones pueden ser muy importantes y de gran utilidad.

*Dominio.* El dominio especifica las restricciones que deben tener los atributos para que puedan almacenar valores válidos. Se deberá estudiar cada atributo y asignar características de dominio. Las características pueden ser un grupo válido de valores o identificar grupos de características similares.

#### REPRESENTACIÓN DEL MODELO ENTIDAD-RELACIÓN.

El objetivo del diagrama entidad-relación es mostrar en diagramas el comportamiento de la organización, identificando los datos principales y la forma en la que éstos se relacionan.

Las principales características del diagrama entidad-relación son:

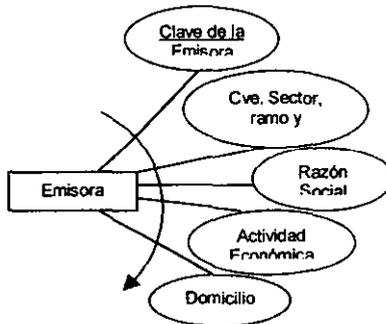
- Proporciona un modelo de necesidades de información de una organización.
- Identifica entidades y sus relaciones.
- Sirve como un punto de partida para la definición de datos.
- Es una excelente fuente de documentación para usuarios, desarrolladores de aplicaciones y administradores de base de datos.

- Es utilizado para crear el diseño físico de la base de datos.

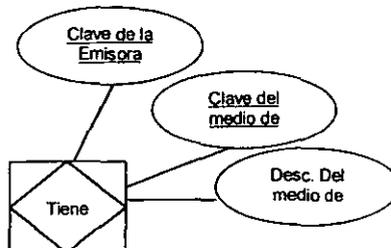
Existen dos tipos de diagramas: Diagrama entidad-atributo (DEA) y diagrama entidad-relación (DER).

### Diagrama Entidad-Atributo.

En él se representa cada entidad con los atributos que mantiene. El nombre del atributo que sea clave será subrayado. En el diagrama, los atributos aparecerán en el orden de la estructura a partir del vértice superior derecho y según las flechas del reloj. Los primeros atributos son los correspondientes a la clave y el resto los más significativos.

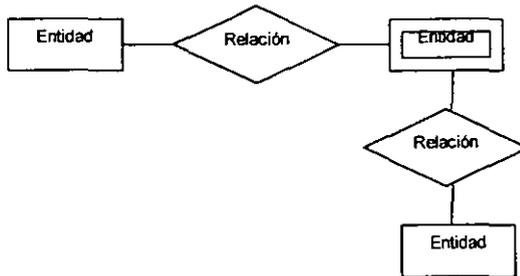


También se puede representar una relación con sus atributos y clave. Es el diagrama relación-atributo (DRA).



### Diagrama entidad-relación.

En este diagrama se representan las distintas entidades y relaciones existentes.



Una relación se representa mediante un verbo. Las entidades son sustantivos que actúan como sujeto y complemento cuando se asocian entre sí mediante la relación, formando todo ello una frase que especifica la gestión que se realiza.

## 2.6 MODELO RELACIONAL.

El modelo relacional forma parte de los modelos lógicos basados en registros. Actualmente es el modelo más utilizado en el diseño de bases de datos. El modelo orientado a objetos también es un modelo de datos que está empezando a tomar fuerza con el desarrollo de sistemas orientados a objetos. Dentro de este trabajo de seminario se va a trabajar tomando como base el modelo relacional para diseñar la unificación de las bases de datos, por lo que se tocará más a fondo este tema.

Una base de datos relacional consiste en una colección de tablas, a cada una de las cuales se les asigna un nombre único. Estas tablas tienen filas de datos. Cada una de estas filas contiene múltiples elementos de datos.

A cada uno de los elementos del modelo relacional se les conoce como: una tabla es una relación, una fila es una tupla y un elemento de datos es un atributo.

Una base de datos relacional es un conjunto de tablas que contienen datos. Es relacional, ya que relaciona datos de una tabla con datos de otra. Permite combinar datos de dos tablas.

### TERMINOLOGÍA RELACIONAL.

*Modelo o modelos de datos.* Comprende el universo total de los datos, es el conjunto completo de todas las tablas relacionales almacenadas en la base de datos.

*Submodelo.* Es el conjunto de relaciones a las que puede acceder un usuario determinado.

*Esquema o estructura de la base de datos.* El esquema de la base de datos es el diseño lógico de la base de datos y una instancia de la base de datos, son los datos que se encuentran almacenados en un instante de tiempo dado en la base de datos.

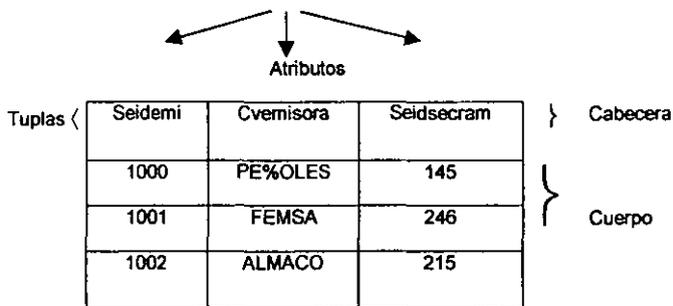
*Valores atómicos de los datos.* Un valor atómico de datos es la unidad de datos más pequeña, que puede ser almacenada o recuperada de una base de datos relacional.

*Dominio.* El dominio es el conjunto de todos los posibles valores atómicos para una o más columnas de una tabla relacional. Pueden identificarse dos tipos de dominios:

- **Dominios Generales o Continuos.** Son los que contienen todos los valores posibles entre un máximo y un mínimo predefinido.

- Dominios restringidos o discretos. Son los que contienen ciertos valores específicos entre un máximo y un mínimo predefinido.

*Tablas o Relaciones.* Es una forma de organizar los datos en filas y columnas lo que equivale a tuplas y atributos respectivamente. Una tabla tiene una cabecera y un cuerpo como se muestra en la figura siguiente.



Las tablas deben tener las siguientes restricciones:

- Debe tener un solo tipo de fila cuyo formato queda definido por el esquema de la tabla. Todas las filas tienen las mismas columnas.
- Cada fila debe ser única y no deben existir filas duplicadas.
- Cada columna debe ser única y no pueden existir columnas duplicadas.
- Cada columna debe estar identificada por un nombre específico.
- El valor de una columna para una fila debe ser único. No pueden existir múltiples valores en una posición de una columna.
- Los valores de una columna deben pertenecer al dominio que representa y es posible que el mismo dominio se utilice para definir valores de otras columnas.

La tabla que reúnan éstas características se llama tabla relacional o relación, sin embargo el término relación se utiliza generalmente para indicar que en la tabla relacional se mantiene la asociación de otras tablas y por tanto representa a una entidad asociativa.

Se le llama *grado* de una tabla relacional al número de atributos que la forman.

Se le llama *cardinalidad* de una tabla al número de tuplas que contiene.

*Atributos.* Un atributo es la característica de una tabla. Cada atributo es identificado por un nombre (que puede identificarse en varias tablas) y todos los posibles valores que puede tener (dominio). La cabecera de una tabla consta de uno o más atributos.

Un atributo utilizado en dos tablas tiene una identificación única de pertenencia a una de las tablas a través de un calificador. El calificador es el nombre de la tabla. Por ejemplo, clientes.nombre y empleados.nombre.

*Tuplas.* Como se mencionó anteriormente, una tabla tiene dos partes, cabecera y cuerpo. La cabecera se conforma por atributos y el cuerpo contiene tuplas o filas.

Las tuplas están formadas por parejas, cada pareja contiene un atributo y un valor correspondiente a ese atributo. Ese valor pertenece a un dominio o puede ser un valor nulo. Cada pareja de las tuplas contiene atributos contenidos en la cabecera

Una pareja atributo-dato puede aparecer en la fila en cualquier orden, de la misma forma las tuplas no tienen por qué estar ordenadas dentro de la tabla.

{Seidemi}	{Cvemisora}	{Seidsecram}	}	Cabecera
{seidemi, 1000}	{cvemisora, PE%OLES}	{seidsecram, 145}		
{cvemisora, FEMSA}	{seidemi, 1001}	{seidsecram, 246}		
{cvemisora, ALMACO}	{seidsecram, null}	{seidemi, 1002}		

Cada una de las tuplas debe contener el mismo número de atributos que la cabecera.

Todas las tuplas tienen el mismo número de atributos.

Para facilitar la presentación de las tablas, los atributos aparecen en la cabecera y no en cada una de las tuplas. De la misma forma los valores de datos se encuentran alineados bajo el atributo de cabecera correspondiente.

*Valores Nulos.* El valor nulo es un dato específico que determina que ningún dato determinado se encuentra presente en el dominio correspondiente. Esto es diferente a un valor vacío. El valor nulo produce resultados predecibles en comparaciones lógicas. Un valor no especificado producirá resultados desconocidos y no predecibles.

*Claves.* Como se ha venido mencionando, cada fila de la tabla debe ser única. Debido a esto, cualquier fila debe tener una única identificación para permitir la selección. Una clave permite esta selección.

Una clave es un atributo o conjunto de atributos cuyos valores distinguen únicamente a una tupla específica de una tabla.

La clave no se debe buscar entre algunos valores de las tuplas, sino a través de todos los valores posibles (dominio) de los atributos.

*Claves Compuestas.* Cuando un atributo no es suficiente para identificar una tupla en una tabla, varios atributos deben combinarse para formar una clave. Una clave formada por varios atributos se llama clave compuesta.

*Claves Candidatas.* Dentro de una tabla puede haber más de una posible clave. Cada posible clave por separado es una clave candidata. Cada tabla debe tener por lo menos una clave candidata, éstas pueden ser claves individuales, compuestas o una combinación de ambas. Las claves candidatas deben ser mínimas, esto quiere decir, que no puedan descomponerse en otras claves candidatas.

Una clave de las claves candidatas debe ser la clave primaria, ésta se elige de acuerdo a la que por excelencia identifique a la tupla. Las claves candidatas no elegidas como clave primaria permanecen como claves alternativas o secundarias.

Los atributos que pertenecen a la clave primaria se denominan atributos primarios y a los demás se les llama atributos no primarios o secundarios.

*Claves ajenas.* Las claves permiten el enlace de la información de diferentes tablas. La clave primaria en una tabla puede localizar datos en otra tabla.

La clave ajena es una clave primaria de una tabla que se utiliza para acceder datos de una tabla diferente.

Las tablas se relacionan únicamente a través de claves primarias y claves ajenas. Estas asociaciones a través de las claves ajenas permiten a los datos ser combinados desde múltiples tablas.

Una clave ajena no es necesariamente parte de la clave primaria de una tabla.

*Interrelaciones.* Una interrelación es una asociación entre tablas mediante atributos que tienen el mismo dominio (o compatible). Estos atributos generalmente hacen referencia a los mismos conceptos t la interrelación se establece entre la clave ajena de una tabla (tabla hija), y la clave principal de la otra tabla (tabla padre).

Las interrelaciones se identifican por un nombre

*Reglas de Integridad.* El modelo relaciona contiene dos reglas que ayudan a asegurar la integridad de los datos:

- Integridad de la entidad.
- Integridad referencial.

La integridad de entidad establece que una clave primaria no puede tener valores nulos. No existe ninguna regla que obligue a que una clave ajena sea única.

La integridad referencial considera la asociación entre las claves primarias y las claves ajenas. La integridad referencial establece que una clave ajena debe existir en una tabla que contenga la clave primaria.

*Vistas.* El usuario puede crear y manejar tablas ficticias cuya definición y tuplas se obtienen a partir de una o más tablas base. Éstas tablas obtienen vistas parciales de datos para que los usuarios solo accedan a determinada información.

Las vistas tienen las siguientes características:

- Sus columnas se obtienen a partir de múltiples tablas base e incluso pueden estar calculadas a partir de valores de las tablas base.
- Pueden estar definidas a partir de otras vistas.
- Sus datos se obtienen a partir de operaciones de recuperación (lectura) de datos.
- Se puede almacenar su definición (estructura, esquema) para una utilización posterior.

La vista es una tabla virtual. Solo se almacena, si se desea, su estructura, donde se muestra cómo se obtiene a partir de tablas base mediante una sucesión de operaciones. Es una forma de ver determinados datos de tablas base, por lo tanto, tal y como se modifiquen las tablas base quedarán afectadas las vistas.

## ÁLGEBRA RELACIONAL.

El álgebra relacional consta de un conjunto de operaciones que toman una o dos relaciones (tablas) como entrada y producen una nueva relación como resultado.

Las operaciones fundamentales en el álgebra relacional son seleccionar, proyectar, producto cartesiano, renombrar, unión y diferencia de conjuntos. Además de las operaciones fundamentales existen otras operaciones como son intersección de conjuntos, cociente, división y asignación.

### *Operaciones fundamentales.*

Las operaciones fundamentales se pueden clasificar según el número de tablas sobre el que opera. De esta forma existen las operaciones:

- Operaciones Unitarias. Aquellas que operan sobre una relación.
  - a) selección
  - b) proyección
  - c) renombre
- Operaciones binarias. Aquellas que operan sobre dos relaciones.
  - a) unión
  - b) diferencia
  - c) producto cartesiano

Selección. Esta operación selecciona tuplas que satisfacen un criterio de selección.

Proyección. La proyección obtiene un subconjunto de columnas de una relación con todas sus filas, creando con este subconjunto una nueva tabla. Para definir el subconjunto de columnas debe existir algún criterio de selección. Las filas o tuplas que se encuentren duplicadas solo aparecerán una sola vez.

Renombre. Cuando se necesita hacer referencia más de una vez a una misma relación es necesario renombrar esa relación con otro nombre de forma que no haya ambigüedades y se puedan comparar datos de la misma relación, para que cumplan con cierto criterio, como si se tratara de dos relaciones diferentes.

Unión. La unión de dos tablas solo se puede realizar si tienen el mismo grado (número de columnas) y los dominios son compatibles (generalmente los mismos atributos). El resultado

es una nueva tabla con las columnas de una de ellas y las filas de ambas tablas. Las tuplas repetidas solo aparecerán una sola vez, debido a que es una condición de las tablas relacionales. Se representa  $R \cup T$ .

Las cardinalidad resultante es la suma de las cardinalidades de las tablas de entrada y restando aquellas que estén duplicadas.

Diferencia. La diferencia entre dos tablas R y T, se representa  $R - T$ , solo se puede realizar si tienen el mismo grado y los dominios son compatibles. El resultado es una nueva tabla con las columnas de una de ellas y las filas de R que no están en T.

La cardinalidad de la tabla resultante es la cardinalidad la suma de las cardinalidades de ambas tablas menos las tuplas comunes.

Producto Cartesiano. También se le puede llamar solamente producto. Se puede realizar entre dos tablas con grados diferentes. El resultado es una nueva tabla cuyo grado es la suma de los grados de las dos tablas operadores y con todas las tuplas que resultan de concatenar las dos tablas. La cardinalidad de la nueva tabla será el producto de las cardinalidades de las tablas de entrada. Esta operación se representa por  $R \times S$ .

### *Operaciones Adicionales.*

Las operaciones fundamentales del álgebra relacional son suficientes para realizar cualquier tipo de consulta, sin embargo algunas de éstas consultas son largas de expresar. Por lo tanto existen operaciones adicionales que simplifican consultas comunes.

Intersección. La intersección de dos tabla R y T es una operación que se deriva de la operación básica diferencia. Esto indica que las tablas deben tener el mismo grado y un dominio compatible. Se representa como  $R \cap T$  y se obtiene al realizar la siguiente secuencia de operaciones.

$$R \cap T = (R - (R - T))$$

Por lo tanto, la tabla resultante tiene las columnas de una de ellas y las filas comunes a ambas tablas.

Cociente. El cociente se realiza entre dos tabla R y T con diferente grado, que cumplen las siguientes condiciones:

- R debe tener los atributos de T, y por lo tanto tener un grado mayor que T, estando el grado de T incluido en el de R. Se tiene que cumplir que  $G(R) = m + n$  y  $G(T) = n$ .
- La cardinalidad de T debe ser distinta de cero.

El resultado es una nueva tabla con grado m (los atributos de R que no están en T) y las tuplas de R de grado m, tales que al concatenar (hacer el producto cartesiano) con todas las tuplas de T producen tuplas contenidas en R. Por tanto, la tabla resultante será igual o estará incluida en R. Se representa como  $R / T$ .

La operación se obtiene de aplicar las siguientes operaciones básicas:

$$R / T = W - Z \quad \text{siendo:}$$

$$W = P(Q, (\text{atributos no comunes a T}))$$

$$Z = p((W \times T) - R, (\text{atributos no comunes a T})).$$

Join, unión natural, yunción o reunión. La operación de join, también llamada unión natural o yunción, es de las más importantes del álgebra relacional. Se obtiene de realizar el producto cartesiano y aplicar sobre la tabla resultante una selección preestablecida que se denomina selección o condición del join.

Como operandos utiliza dos tablas que se confrontan (mediante el operador de igualdad, desigualdad, mayor, etc.) entre sí, fila a fila por las columnas que se especifiquen y generalmente corresponderán a claves ajenas. El resultado es una tabla con las columnas de las dos tablas y los valores de las tuplas que satisfacen la condición de comparación sobre los atributos definidos.

El grado resultante dependerá de si van a existir columnas iguales o no. La cardinalidad resultante dependerá de la condición impuesta por el join.

Las columnas o atributos que van a utilizarse para la comparación del join tienen que pertenecer al mismo dominio. El símbolo que se utiliza para denotar la operación es el asterisco (\*).

Todas estas operaciones relacionales son funciones internas que ejecuta el DBMS y que tras operar con argumentos de entrada devuelve un resultado. Las tablas resultantes sólo se generan en memoria interna para visualizar sus datos u operar con ella pero nunca se almacena dentro de la base de datos.

#### REGLAS DE LAS BASES DE DATOS RELACIONALES.

Debido al auge de las bases de datos relacionales y a su proliferación en el mercado, Ted Codd, el creador de las bases de datos relacionales, publicó en 1985 once reglas que debe cumplir cualquier base de datos que se considere relacional.

- a) 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.
- b) Regla de acceso garantizado. Todos y cada uno de los datos (valor atómico) de una base de datos relacional se garantiza que sean lógicamente accesibles recurriendo a una combinación de nombre de tabla, valor de clave primaria y nombre de columna.
- c) Tratamiento sistemático de valores nulos. Los valores nulos se soportan en los DBMS completamente relacionales para representar la falta de información y la información inaplicable de un modo sistemático e independiente del tipo de datos.
- d) 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 pueden aplicar a su interrogación el mismo lenguaje relacional que aplican a los datos regulares.
- e) Regla de sublenguaje completo de datos. Un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal. Sin embargo debe haber al menos un lenguaje cuyas sentencias sean expresables, mediante alguna sintaxis

bien definida, como cadenas de caracteres, y que sea completa en cuanto al soporte de todos los puntos siguientes:

- Definición de datos
  - Definición de vista
  - Manipulación de datos (interactiva y por programa)
  - Restricciones de integridad
  - Autorización
  - Fronteras de transacciones
- f) Regla de actualización de vista. Todas las vistas que sean teóricas actualizables son también actualizables por sistema.
- g) Inserción, modificación y borrado 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 modificación, inserción y borrado de los datos.
- h) 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.
- i) Independencia lógica de los datos. Los programas de aplicación y las actividades terminales permanecen lógicamente inalterados cuando se efectúen sobre las tablas de base cambios preservadores de la información de cualquier tipo que teóricamente permita alteraciones.
- j) Independencia de integridad. Las restricciones de integridad específicas para una base de datos relacional particular deben de ser definibles en el sublenguaje de datos relacional y almacenables en el catálogo, no en los programas de aplicación.

- k) Regla de no subversión. Si un sistema relacional tiene un lenguaje de bajo nivel (un solo registro cada vez) ese bajo nivel 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),

Existen otros tipos de modelos de datos distintos al modelo relacional. El modelo orientado a objetos es un nuevo modelo que está tomando fuerza junto con los lenguajes de programación orientados a objetos. El modelo jerárquico y el modelo de red fueron los primeros modelos de datos que se inventaron pero que no alcanzaron a cumplir con todas las expectativas del mercado, teniendo deficiencias considerables en cuanto al manejo de los datos y a la dificultad de su utilización, ya que seguían la línea del sistema de archivos. Ninguno de éstos dos últimos modelos se utilizan actualmente, sin embargo son un antecedente de los actuales modelos de datos.

## **2.7 MODELO DE DATOS ORIENTADO A OBJETOS**

El modelo de datos orientado a objetos, surgió con el desarrollo de lenguajes de programación orientados a objetos. Dentro del proyecto de unificación de las bases de datos de la Bolsa Mexicana de Valores no se utilizará este tipo de modelo, debido a la naturaleza de los sistemas de información imperantes dentro de la organización y al diseño de las actuales bases de datos y de la información en general.

Por lo tanto, se tratará este tema como una visión general del diseño orientado a objetos, mas no se tratará a fondo.

El modelo orientado a objetos se basa en una colección de objetos. Encapsula el código y los datos en una única unidad, llamada objeto. La interfaz entre un objeto y el resto del sistema se define mediante un conjunto de mensajes.

Los objetos tienen una identidad única de objeto (IO) que los distingue de cualquier otro.

Un objeto tiene contiene:

- Un conjunto de variables que contienen los datos del objeto. El valor de cada variable es un objeto.
- Un conjunto de mensajes a los que el objeto responde.
- Un método que es un trozo de código para implementar cada mensaje. Un método devuelve un valor como respuesta al mensaje.

Los mensajes son las peticiones que se realizan entre objetos, para acceder a la información propia de cada uno de los objetos. Los envíos de mensajes entre objetos son las llamadas a los métodos de otro objeto con el fin de obtener información. Debido a que la única interfaz externa que presenta un objeto es el conjunto de mensajes al que responde, es posible modificar la definición de métodos y variables sin afectar a otros objetos.

Los objetos se agrupan en clases de objetos que posean propiedades semejantes, Los conceptos de objeto y de clase en las bases de datos son esencialmente los mismos que los que se encuentran en los lenguajes de programación orientados a objetos como C++ o Smalltalk.

ESTA TESIS NO FORMA PARTE DE LA BIBLIOTECA

En el diseño orientado a objetos se describen tres tipos de propiedades:

- a) **Atributos.** Son las propiedades cuyos tipos se construyen a partir de tipos primitivos como enteros o cadenas. Un atributo tiene un tipo que no incluye ninguna clase.
- b) **Relaciones.** Son propiedades cuyo tipo es una referencia a un objeto de alguna clase o una colección de tales referencias.
- c) **Métodos.** Son funciones que pueden aplicarse a los objetos de la clase

De la misma forma en que los objetos se agrupan en clases, el conjunto de clases está estructurado en sub y superclases, puesto que el valor de un dato en un objeto también es un objeto, es posible representar el contenido del objeto dando como resultado un objeto compuesto.

La capacidad de modificar la definición de un objeto sin afectar el resto del sistema está considerada como una de las mayores ventajas del modelo de programación orientado a objetos.

El lenguaje de definición de objetos (ODL) es el propuesto como estándar para especificar la estructura de las bases de datos en términos orientados a objetos.

## 2.8 MODELO JERÁRQUICO.

Una base de datos jerárquica consiste en una colección de registros que se conectan entre sí por medio de enlaces. Este modelo se representa en forma de árbol.

Retomando la estructura de los modelos de datos, se analizará este modelo en base a su parte estática y su parte dinámica.

### PARTE ESTÁTICA:

En cuanto a la estructura que ofrecen este tipo de modelos, se distinguen los siguientes conceptos principales:

*Segmento.* Es como de les llama en este modelo a las Entidades. Existen Segmentos raíz y segmentos dependientes, también llamados segmentos padre y segmentos hijo, dependiendo de la posición que tengan dentro de la estructura jerárquica. También se pueden distinguir los segmentos tipo (definen a una entidad) y los segmentos ocurrencia (definen a un conjunto de datos de dicha entidad).

*Campo.* Son las partes individuales que forman a los segmentos, se les conoce en el modelo relacional como atributos.

*Dependencia o Relación Jerárquica.* Es la relación que existe entre los segmentos tipo. Las relaciones solo se pueden dar entre dos segmentos y no puede existir más de una relación para cada segmento, por lo que no pueden existir relaciones 1:N.

Entre las principales restricciones del modelo jerárquico se encuentran:

- No se permite más de una relación entre segmentos.
- No se admiten relaciones reflexivas.
- No se permite un segmento hijo con más de un segmento padre.
- No se permiten relaciones de tipo M:N
- El árbol es recorrido en un orden preestablecido.
- Es obligatorio entrar por el segmento raíz para acceder a cualquier segmento ocurrencia.

#### PARTE DINÁMICA.

Para realizar cualquier tipo de consulta a los datos, en el modelo jerárquico, es necesario recorrer registro por registro hasta encontrar el que cumpla con la condición requerida. Es un modelo procedimental.

Sus principales características son:

- Solo tiene un segmento raíz.
- El segmento raíz (padre) puede tener n segmentos dependientes (hijos).
- Cada segmento hijo puede tener a su vez n segmentos hijos.
- Ningún segmento hijo dependiente de otro segmento hijo, puede existir sin el segmento hijo superior a él.

## 2.9 MODELO DE RED

El modelo de red difiere del modelo relacional en que los datos se representan por medio de colecciones de registros y las relaciones de los datos mediante enlaces.

Este modelo fue propuesto por el grupo DBGT (Data Base Task Group) del comité CODASYL.

Este modelo se explicará también bajo los términos de parte estática y parte dinámica.

### PARTE ESTÁTICA

Dentro de la estructura que propone este tipo de modelo se destacan los siguientes conceptos:

*Artículo o Registro.* Es el equivalente a las relaciones o tablas del modelo relacional. Se define como un conjunto de datos homogéneos que constituye la unidad de intercambio de información.

*Conjuntos o Sets.* Los conjuntos son las asociaciones entre artículos. En cada conjunto existe un artículo cabecera o propietario y uno o varios artículos miembro.

## PARTE DINÁMICA.

En este modelo, al igual que en el jerárquico, las consultas se realizan recorriendo cada uno de los elementos de los artículos, por lo que también es un modelo procedimental.

Una base de datos de red comprende una colección de registros que están conectados entre sí por medio de enlaces. Un registro es similar a una entidad en el modelo entidad-relación. Cada registro es una colección de datos (atributos), cada uno de los cuales contiene solamente el valor de un dato. Un enlace es una asociación exclusivamente entre dos registros.

Este tipo de modelo a pesar de que mejoró el procesamiento de consultas continuaba teniendo un proceso complicado, debido a que la lógica de los procesos continuaba dependiendo de la organización física de los datos, además de que no proporcionaba una independencia total de las aplicaciones.

### **2.10 NORMALIZACIÓN DE BASES DE DATOS RELACIONALES**

Como se ha venido tratando a lo largo del capítulo, las bases de datos tienen objetivos fundamentales como lo son reducir la redundancia de los datos, cuidar la integridad de los datos, aumentar la seguridad de la información, fácil manejo y control de cambios, crecimiento y requerimientos nuevos. Para poder proveer de todas estas características a las bases de datos se debe seguir un proceso llamado normalización, implementando reglas establecidas llamadas formas normales. La normalización proporciona las reglas para asociar atributos a entidades de manera correcta.

Las tres primeras formas normales fueron definidas por E. F. Codd. Posteriormente algunos problemas detectados obligaron a crear la FNBC (forma normal de Boyce y Codd), después Fagin definió la cuarta y quinta forma normal.

El proceso de normalización es utilizado para asegurar que el modelo conceptual de la base de datos funcionará. La normalización tiene como objetivo crear el mayor número de tablas posible, cada una de ellas compuestas por atributos necesarios que la representan y por relaciones entre entidades a la que hace referencia la tabla por medio de llaves.

Las principales ventajas de la normalización son:

- **Facilidad de uso.** Datos agrupados en tablas que identifican claramente un objeto.
- **Flexibilidad.** Todos los datos pueden ser extraídos de la base de datos utilizando el álgebra relacional.
- **Precisión.** Las relaciones de las tablas se pueden controlar con exactitud.
- **Seguridad.** Los controles de acceso son menos complicados de implementar.
- **Facilidad de Implementación.** Las tablas se almacenan como archivos planos
- **Independencia de datos.** La estructura de la base de datos no está relacionada a los programas de aplicación que los generan o utilizan, por lo que se pueden realizar modificaciones como aumento y eliminación de atributos o de tablas sin ningún problema.
- **Claridad.** La información se representa en forma clara para los usuarios.
- **Facilidad de diseño.** Los programas de aplicación pueden manejar de forma sencilla la base de datos por medio del álgebra y el cálculo relacional.
- **Minima redundancia.** La información no se duplica innecesariamente dentro de las estructuras.

- Máximo rendimiento de las aplicaciones.
- Minimiza el costo de mantenimiento de la base de datos, debido es más sencillo y seguro manejar los cambios.
- Incrementa la estabilidad del modelo de datos.

Cada forma normal obliga a los datos a estar más organizados que la forma normal anterior, por lo que cada forma normal debe ser cumplida antes de aplicar la siguiente.

#### PRIMERA FORMA NORMAL (1FN).

*"Una entidad (relación) se encuentra en 1NF si no contiene grupos repetitivos"*

Cada uno de los registros debe tener el mismo número de atributos, y cada atributo debe contener solo un valor a la vez.

Si en una tabla tenemos grupos repetitivos, lo que se deberá hacer para que cumpla con 1FN es formar un nuevo registro con cada uno de los datos repetitivos. Para llegar a esta forma normal, es necesario explicar qué es y en qué consiste la dependencia funcional de los atributos.

La *dependencia funcional* significa que los valores para dos columnas están lógicamente asociados. El valor de A determina el valor de B. Si B es funcionalmente dependiente de A, entonces A es el determinante de B.

Una columna puede ser funcionalmente dependiente de dos o más columnas que formen una clave compuesta.

Cada una de las otras columnas de una tabla dependen funcionalmente de una clave candidata.

Se dice que existe una dependencia funcional completa cuando una columna es funcionalmente dependiente de una llave compuesta, pero no tiene dependencia funcional con las partes de la llave.

#### SEGUNDA FORMA NORMAL (2FN).

*"Una entidad se encuentra en segunda forma normal si ya ha cumplido con la primera forma normal y además todos sus atributos deben tener dependencia funcional completa de la clave primaria"*

Si una relación se encuentra en segunda forma normal, la llave primaria es necesaria para identificar todos sus atributos asociados.

La segunda forma normal requiere que cada atributo que no forme parte de la llave primaria, sea dependiente de ésta para que el valor que almacena sea significativo.

### TERCERA FORMA NORMAL.

*"Una tabla debe estar en 2FN antes de estar en 3FN. En una tabla en 3FN, todos los elementos no clave deben ser mutuamente independientes."*

Esto quiere decir que un atributo secundario (atributo no clave) solo se debe conocer a través de la llave primaria y no por medio de otro atributo no primario.

Cuando un atributo A tiene una dependencia funcional con un atributo B que tiene dependencia funcional con la llave primaria y que no forma parte de ésta, entonces el atributo A tiene una dependencia transitiva con la llave primaria, ya que depende de ella solo a través del atributo B.

### FORMA NORMAL BOYCE-CODD (FNBC)

*"Una tabla está en FNBC sí y solo sí las únicas dependencias funcionales son sobre las claves candidatas"*

Esta forma normal es un requisito más estricto que la 3FN, donde las columnas no clave tienen que ser independientes funcionalmente. En una tabla en FNBC no hay dependencias funcionales sobre las columnas que no sean claves candidatas.

La FNBC se aplica para la normalización de una tabla que esté en 3FN y que tenga las siguientes propiedades:

- Múltiples claves candidatas
- Claves candidatas compuestas
- Columnas compartidas entre las claves compuestas.

## CUARTA FORMA NORMAL.

Algunas tablas a pesar de estar en FNBC siguen teniendo muchos datos repetitivos y continúan siendo difíciles de actualizar. La cuarta forma normal se apoya sobre dependencias multivaluadas.

Una dependencia multivaluada se da cuando para cada valor de un atributo A corresponden un conjunto de valores de B asociados, y ésta asociación de conjunto de valores es independiente del resto de los atributos.

Como es necesario que en una dependencia multivaluada entre dos atributos el resto de campos sean independientes, deben existir al menos tres atributos para que exista una dependencia multivaluada.

La cuarta forma normal se aplica para eliminar las dependencias multivaluadas de las tablas, ya que causan redundancia en los datos y dificulta la actualización provocando problemas de integridad.

La 4FN la generó Fagin y cuyo teorema dice:

*"Una tabla T con atributos A, B y C se pueden descomponer sin pérdidas en dos partes T1(A, B) y T2(A, C) si y solo si la T cumple con la dependencia multivaluada de A sobre B y éstos son independientes de C"*

Se dice que una tabla se encuentra en la 4FN si:

- Está en FNBC
- Las únicas dependencias multivaluadas existentes son las dependencias funcionales de la clave con los atributos no clave.

Si una dependencia multivaluada se entiende como una relación entre atributos de tipo 1:n, si n es mayor a 1 ya que si n es igual a 1 entonces se trata de una dependencia funcional, por lo que se puede concluir que una tabla está en 4FN si no mantiene dependencias multivaluadas.

#### QUINTA FORMA NORMAL (5FN).

Esta forma normal fue creada también por Fagin. En esta se tratan las dependencias entre entidades y no solo entre atributos.

Para que una tabla se encuentre en 5FN se deben cumplir dos condiciones:

- Se encuentre en 4FN
- Toda unión (con otras tablas, conocida como Join) viene implicadas por las claves (principal o candidatas) de la tabla.

Es decir, la tabla estará en 5FN si es posible generar proyecciones y realizar su unión (join), los atributos comunes que realizan la operación (atributos de join) están formados por claves (principal o candidatas) de la tabla.

La dependencia de join (DJ) es una dependencia entre tablas, problema que controla la quinta forma normal.

*"Una dependencia de join indica que una tabla T formada por los atributos A1, A2, A3, ... An tiene una dependencia de join con sus proyecciones T1, T2, ... Tn si la tabla original se puede obtener por medio de la unión de sus proyecciones".*

## 2.11 INTEGRIDAD DE LAS BASES DE DATOS

La integridad de los datos se refiere a la coherencia y veracidad de la información que se encuentra almacenada en tablas dentro de una base de datos. Las principales acciones que pueden afectar la integridad de los datos son las que actualizan la información, como son:

- Modificación
- Inserción
- Borrado

La integridad de la información se refiere a medidas de seguridad utilizados para mantener correctos los datos en la base de datos.

Para asegurar la integridad de los datos no solo son necesarias la confiabilidad física y lógica (hardware y software), sino también es necesario comprobar que los datos son correctos. La validación de los datos de entrada es una operación común en el procesamiento de datos.

La tecnología de bases de datos proporciona recursos que refuerzan la integridad de los datos.

Para mantener la integridad de una base de datos se deben tener los siguientes tipos de restricciones:

- Restricciones de los valores que puede tomar un atributo en la base de datos, considerando tanto el tipo de dato como su dominio o conjunto de valores. Las operaciones DML afectadas son: modificación e inserción.
- Atributos en los que siempre deben permanecer con información y en la que la base de datos no admite valores NULL. Estos atributos pueden ser claves o no.
- Los valores de los atributos que forman parte de la clave principal o primaria son únicos y la base de datos no puede admitir valores nulos ni duplicados. Esta restricción es llamada *Integridad de Entidad*.
- La base de datos tiene que forzar a que no existan registros hijos si es que no existe el registro padre correspondiente. Además para efectos de la manipulación de los datos, en toda operación de inserción o modificación sobre una tabla hija no tiene porqué afectar a su tabla padre. Si hay una inserción o modificación la clave foránea de la tabla, ésta debe tener su correspondencia en la tabla padre o un valor null. A esta restricción se le llama *Integridad Referencial*. Existen tres tipos de restricciones referenciales:

- a) Cíclicas. Aseguran una relación padre-hija entre tablas. Controlan que cada fila en la tabla hija tenga una fila correspondiente en la tabla padre.
- b) Auto-referenciables. Que restringe una relación padre-hija dentro de la misma tabla. Obliga a que cada fila hija en la tabla tenga una fila padre en la misma tabla.
- c) Múltiples. Las restricciones múltiples hacen referencia a llaves primarias que pueden tener varias llaves foráneas correspondientes.

- Pueden existir reglas que son impuestas por el negocio. La base de datos tiene que ser capaz de validar las operaciones que pueden verse afectadas por las reglas del negocio. Este tipo de restricciones toman el nombre de *Reglas Semánticas*.
- En ocasiones también pueden existir tablas que almacenan valores históricos y para alimentar a este tipo de tablas se deben correr procesos de actualización. Estos procesos deben efectuarse automáticamente en la base de datos cada vez que se realizan determinadas operaciones para mantener la consistencia de la información.

La persona responsable de la base de datos puede especificar en la estructura (esquema) las condiciones para la validación o comprobación de la consistencia, además de las medidas de integridad proporcionadas por el sistema.

Es muy importante tener especial cuidado en la integridad de la información, los DBMS proporcionan herramientas muy útiles para este fin, como los son, los triggers (disparadores). Los disparadores son acciones que se ejecuta automáticamente el DBMS cuando se produce un cambio en los valores de una tabla. Los tipos de cambios, la tabla afectada, los atributos y las acciones que se deben llevar a cabo, los define el usuario según sus necesidades.

En éstos se puede definir las instrucciones a realizar al momento de hacer una operación DML en la base de datos. Por ejemplo, en un disparador se puede guardar las instrucciones a seguir cuando se realiza la actualización a una llave primaria de una tabla padre, esta actualización se reflejará en todas las tablas mediante este proceso.

Si no se definen en procesos como los disparadores las instrucciones a seguir al momento de realizar alguna modificación a una tabla padre (las repercusiones a todas las tablas hijas), el DBMS reaccionará de la siguiente manera:

- Si el usuario borra una llave primaria y ésta llave primaria tiene llaves foráneas correspondientes (tablas hijas) la instrucción falla y no se ejecuta.
- Si el usuario modifica una llave primaria y existen valores de llaves foráneas que no corresponden al nuevo valor de la llave primaria, dicha modificación no se ejecuta.
- No existen restricciones relacionadas al borrado de llaves foráneas.
- Si el usuario modifica una llave foránea, y no hay una valor de la llave primaria que le corresponda al nuevo valor no-nulo de la llave foránea, la operación no se ejecuta.
- Todos los valores de la llave primaria deben ser únicos, cualquier intento por insertar valores duplicados a la llave primaria no se ejecutará.
- Cuando se inserta una fila en una tabla hija, si todas las llaves foráneas son nulas y no tienen una valor correspondiente en la llave primaria de la tabla padre, la operación no se ejecuta.

Al momento de diseñar cada una de las tablas que van a formar parte de la base de datos, es importante tomar en cuenta las relaciones existentes entre ellas y el papel que van a jugar, ya sea tablas padre o hijas, y en base a esto diseñar los procesos necesarios para mantener la base de datos actualizada cuidando la integridad de la información que almacena.

Este control de la base de datos se debe de llevar al momento de su diseño, sin embargo también se debe llevar un control estricto cuando esté siendo utilizada por los

usuarios, este control se llevará a cabo por medio de la *seguridad* con la que cuente la base de datos.

## 2.12 SEGURIDAD DE LA BASE DE DATOS.

La seguridad de la información hace referencia al mal uso que se haga de la información almacenada en la base de datos.

La seguridad de una base de datos implica que no accedan a la información almacenada usuarios no autorizados y que las actualizaciones que se realicen en los datos, sean actualizaciones permitidas.

La pérdida accidental de la consistencia de los datos puede deberse a:

- Caídas durante el procesamiento de la transacciones.
- Anomalías por acceso concurrente a la base de datos.
- Anomalías que resultan de la distribución de los datos.

Algunas formas de mal uso de la información son:

- Lectura de datos sin autorización (robo de información)
- Modificación de datos sin autorización
- Destrucción no autorizada de los datos.

Para proteger a la base de datos es necesario adoptar medidas de seguridad en varios niveles:

- Físico. La localidad o localidades que contienen a los sistemas de equipos de cómputo deben protegerse físicamente contra la penetración armada o clandestina.

- **Humano.** Debe tenerse mucho cuidado al conceder autorización a los usuarios para reducir la probabilidad de que un usuario autorizado permita el acceso a un intruso.
- **Sistema Operativo.** Aunque el sistema de base de datos esté bien protegido, si no se protege de forma adecuada el sistema operativo, éste puede servir para obtener acceso sin autorización a la base de datos. Dado que casi todos los sistemas de base de datos permiten acceso remoto a través de terminales o redes, la seguridad a nivel software dentro del sistema operativo es tan importante como la seguridad física.
- **Sistemas de base de datos.** Los sistemas de bases de datos proporcionan las restricciones de acceso de los usuarios a la base de datos, limitando las transacciones que puedan realizar en ella, dependiendo del tipo de usuario que se trate.

Es vital para una organización, cuidar todos los aspectos de la seguridad de la base de datos. En este trabajo de seminario nos enfocaremos a estudiar más a fondo la seguridad a nivel de base de datos, debido a que los demás aspectos de la seguridad se salen del alcance del proyecto.

#### AUTORIZACIONES Y VISTAS.

Dentro de las bases de datos, se utilizan las vistas para controlar el acceso a los datos. Una vista puede ocultar datos que el usuario no necesita ver. Esto simplifica la utilización del sistema y fomenta la seguridad. La seguridad se logra si se cuenta con un mecanismo que limite a los usuarios a su vista o vistas personales. Normalmente las bases de datos relacionales tienen dos niveles de seguridad:

- **Relación.** Puede permitirse o impedirse que el usuario tenga acceso directo a una relación.

- *Vista*. Puede permitirse o impedirse que el usuario tenga acceso a la información que aparece en una vista.

Un usuario puede tener varias formas de autorización sobre partes de la base de datos. Entre ellas están:

- *Autorización de lectura*, que permite leer, pero no modificarla base de datos.
- *Autorización de inserción*, que permite insertar datos nuevos, pero no modificar los ya existentes.
- *Autorización de actualización*, que permite modificar la información, pero no permite la eliminación de datos.
- *Autorización de borrado*, que permite la eliminación de datos.

Un usuario puede tener asignados todos, ninguno o una combinación de los tipos de autorizaciones anteriores. Existen también otro tipo de autorizaciones para que el usuario modifique el esquema o estructura de la base de datos:

- *Autorización de índice*, que permite la creación y eliminación de índices.
- *Autorización de recursos*, que permite la creación de relaciones (tablas) nuevas.
- *Autorización de alteración*, que permite agregar o eliminar atributos de una relación.
- *Autorización de eliminación*, que permite eliminar relaciones.

Este tipo de autorizaciones difieren de las anteriores en que estas actúan sobre el esquema de la base de datos y las anteriores sobre las tuplas o los datos almacenados.

Con la autorización de recursos, la utilización del espacio de almacenamiento puede ser controlada. Al usuario con autorización de recursos que crea una nueva relación se le dan todos los privilegios sobre esa relación automáticamente.

La creación de una vista no requiere de autorización de recursos. EL usuario que crea una vista no recibe necesariamente todos los privilegios sobre esa vista.

La autorización de índice puede aparecer innecesariamente, puesto que la creación o eliminación de un índice no altera los datos en las relaciones. Mas bien, los índices son una estructura para realizar mejoras. Es necesario considerar la creación de índices como un privilegio, para permitir al administrador de la base de datos regular el uso de los recursos del sistema.

La forma fundamental de autoridad es la que se le da al administrador de la base de datos. El administrador de la base de datos puede autorizar nuevos usuarios, reestructurar la base de datos, etc. Esta forma de autorización es análoga a la que se provee a un "superusuario".

Al usuario que se le ha concedido alguna forma de autoridad se le puede permitir pasar esta autoridad a otros usuarios. Sin embargo hay que tener control sobre cómo se pasa esta autoridad entre usuarios para asegurar que tal autorización pueda ser anulada en el futuro.

Otras técnicas de seguridad de la base de datos es a través del "cifrado" o encriptado de la información. Con ésta técnica no es posible leer los datos encriptados a menos que el usuario sepa cómo descifrar la información.

Existe un gran número de técnicas para encriptar la información. Las buenas técnicas de encriptado tienen las siguientes propiedades:

- Para los usuarios autorizados es relativamente sencillo encriptar y descifrar datos.
- El sistema de cifrado no depende de mantener en secreto el algoritmo, sino de un parámetro del algoritmo llamado clave de encriptado.
- Para un usuario no autorizado es muy difícil determinar cuál es la clave de encriptado.

Es muy importante que se tomen al pie de la letra estas consideraciones para mantener la seguridad en la información de la base de datos, ya que si se llega a perder la seguridad las pérdidas serían muchas, no hay que olvidar que la información es el activo más importante de las empresas y es la parte más importante de toda la estructura informática de la organización.

### 2.13 LENGUAJES DE MANIPULACIÓN DE DATOS.

Los sistemas de gestión de bases de datos deben proporcionar lenguajes e interfaces apropiados para cada tipo de usuarios.

Cuando se ha terminado el diseño de la base de datos y ya se ha elegido un DBMS en el que se va a implementar la base de datos, hay que especificar los esquemas conceptual e interno de la base de datos, así como las correspondencias entre ellos.

En muchos DBMS en los que no existe separación entre niveles, el DBA y los diseñadores, utilizan un mismo lenguaje, llamado *lenguaje de definición de datos (DDL: data definition language)* para especificar los dos esquemas.

El DBMS cuenta con un compilador del DDL que procesa las instrucciones escritas en DDL para identificar las descripciones de los esquemas y almacenar esa descripción del esquema en el catálogo del DBMS.

Si el DBMS hace distinción entre los niveles conceptual e interno, el DDL solo se utilizará para especificar el esquema conceptual (entidades, interrelaciones y restricciones). De la misma forma, el *lenguaje de definición de almacenamiento (SDL: Storage definition language)* es utilizado para especificar el esquema interno (las estructuras de almacenamiento y los sistemas de acceso). Las correspondencias entre los dos esquemas puede escribirse en cualquier de los dos lenguajes.

Para poder llegar a una arquitectura de tres esquemas, es necesario el *lenguaje de definición de vistas (VDL: view definition language)* para especificar las vistas de usuario y sus correspondencias con el esquema conceptual.

Después de declarar la estructura de la base de datos y de introducir datos en la misma, los usuarios necesitan disponer de algún medio para acceder a dichos datos. Las operaciones más comunes son la selección (recuperación o consulta), la inserción (introducción de datos), la eliminación y la modificación de datos, para tal fin, el DBMS proporciona un *lenguaje de manipulación de datos (DML: data manipulation language)*.

Los actuales DBMS no disponen de varios lenguajes de datos, sino que ofrecen un único lenguaje que cuenta con los elementos necesarios para definir esquemas conceptuales y vistas, manipular datos y definir su estructura de almacenamiento.

Existen dos tipos de DML: Alto Nivel (No procedimentales o declarativos) y Bajo Nivel (Procedimentales)

a) DML no procedimentales.

- Requiere que el usuario especifique qué datos necesita obtener o actualizar, sin especificar cómo obtenerlos.
- Puede utilizarse de forma independiente para realizar operaciones complejas de base de datos.
- Es posible utilizarlo de dos formas:
  - Interactivamente (desde una terminal).
  - Incorporado a un programa escrito en un lenguaje de programación como C o Visual Basic (DML embebido).
- Puede recuperar y actualizar muchos registros con una sola sentencia (DML orientado a conjuntos).

b) DML procedimental.

- Requiere que el usuario especifique qué datos necesita obtener o modificar y cómo obtener y actualizar tales datos.
- Siempre debe estar incrustado (DML embebido) en el código de un programa escrito en un lenguaje de programación.
- Normalmente solo permite obtener uno a uno los registros de la base de datos para procesarlos por separado (DML orientado a registros).
- Necesita utilizar los elementos de los lenguajes de programación como los ciclos, para poder obtener cada uno de los registros del conjunto de los que interesan, almacenados en la base de datos, y para procesarlo individualmente.

Siempre que las sentencias en DML se incorporen en un programa escrito en cierto lenguaje, a éste último se le denomina *lenguaje host (o anfitrión)*, y al DML *sublenguaje de datos o lenguaje embebido*.

Cuando un DML no procedimental se usa de forma independiente e interactiva, se le denomina *lenguaje de consulta*.

Los usuarios ocasionales utilizan un lenguaje de consulta para especificar sus solicitudes de información. Los programadores suelen utilizar el DML embebido o incrustado en algún lenguaje de programación. Los usuarios paramétricos utilizan interfaces amigables que les permitan interactuar con la base de datos; también pueden utilizarlas los usuarios ocasionales y aquellos no interesados en aprender un lenguaje de consulta.

El lenguaje estándar utilizado como DDL y DML es el lenguaje llamado SQL (Structured Query Language), del cuál hablaremos con más detalle en la sección siguiente.

### SQL (STRUCTURE QUERY LANGUAGE)

El SQL fue el primer lenguaje de base de datos de alto nivel diseñado en los años 70's. Fue diseñado e implementado en IBM Research (San José, California), para el DBMS Relacional Experimental SYSTEM R.

Existen dos estándares principales de SQL: el SQL ANSI y un estándar actualizado que se adoptó en 1992, llamado SQL 92 u SQL2. También existe un estándar más reciente,

el SQL3, que amplía el anterior con características como la recursión, los disparadores y los objetos.

En esta sección, trataremos las características principales de SQL y las sentencias más comunes, sin explicar muy a fondo cada una de ellas, debido a que quedaría fuera del alcance del proyecto. Esta sección se encuentra dividida en dos partes: las sentencias correspondientes al DDL y las sentencias correspondientes al DML.

### *Definición de Datos (DDL).*

Crear una tabla. La sentencia *create table* crea una nueva tabla y también las columnas o atributos de la tabla. El estándar ANSI-86 para la creación de una tabla es:

```
CREATE TABLE nombre-de-la-tabla
  [(nombre-de-columa tipo-de-dato [NOT NULL [UNIQUE]] | UNIQUE),
  ...])
```

La sentencia *create table* crea una tabla y, opcionalmente, las columnas para esa tabla. Por ejemplo:

```
CREATE TABLE Empleado(
  emp-num char(10),
  emp-nomb char(15),
  emp-appa char (15),
  emp-apma char (15), ...
)
```

El estándar indica que no puede haber dos tablas del mismo nombre en la misma base de datos. Algunas implementaciones permiten que dos tablas tengan el mismo nombre siempre y cuando tengan diferente propietario.

*Eliminar una tabla.* No existe un estándar para la sentencia de eliminación, sin embargo la sentencia más común para la eliminación de una tabla de una base de datos es:

```
DROP TABLE nombre-de-la-tabla
```

Crear una tabla temporal. La siguiente sentencia crea una tabla temporal:

```
CREATE TEMP TABLE nombre-de-la-tabla  
(nombre-de-columna, tipo-de-dato [UNIQUE]), ...)
```

Una tabla temporal puede funcionar para almacenar información durante la sesión actual, ésta es eliminada al momento de terminar la sesión.

La palabra reservada **TEMP** no es parte del estándar ANSI y no se encontrará en todas las implementaciones de SQL.

Añadir columnas a una tabla. La sentencia *alter table* añade columnas a una tabla existente. Esta sentencia no forma parte del estándar ANSI, por lo que se puede presentar en algunas formas diferentes como:

```
ALTER TABLE nombre-de-la-tabla ADD COLUMN  
Nombre-de-la-columna tipo-de-dato [NOT NULL]  
[DEFAULT {literal ! SYSTEM}]
```

```
ALTER TABLE nombre-de-la-tabla ADD  
(Nombre-de-la-columa tipo-de-dato) [NOT NULL  
WITH DEFAULT]
```

```
ALTER TABLE nombre-de-la-tabla ADD  
Nombre-de-la-columna tipo-de-dato [NULL ! NOT NULL]
```

Dos columnas en una misma tabla no pueden tener el mismo nombre.

### Acceso a los Datos (DML).

Se pueden recuperar datos selectivamente en las tablas. Esta sección muestra cómo realizar estas selecciones y combinaciones de datos. Se van a describir las operaciones de: selección, proyección, unión, intersección, diferencia, join y vista.

Cada una de las operaciones produce una nueva tabla (conceptualmente, no físicamente). Las primeras operaciones, selección y proyección, operan sobre una única tabla para producir una nueva tabla. Las operaciones de diferencia y join combinan múltiples tablas para producir una nueva tabla. Una vista crea una nueva tabla de una o más tablas.

Selección. La sentencia *select* muestra en pantalla alguna o todas las filas de una tabla. La sintaxis de esta sentencia es la siguiente:

```
SELECT * FROM {nombre-de-la-tabla | nombre-de-la-vista}, ...
```

El asterisco indica que todas las columnas de la tabla serán seleccionadas.

Condiciones de búsqueda. Una condición de búsqueda restringe la selección de los registros de una tabla.

```
SELECT *  
FROM {nombre-de-tabla | nombre-de-vista}, ...  
WHERE condición de búsqueda
```

La condición de búsqueda es de la forma:

```
[NOT] (predicado | condición de búsqueda)  
[ {[AND | OR] [NOT] (predicado | condición de búsqueda)} ...]
```

Los operadores relacionales en una condición de búsqueda pueden restringir una selección. Estos operadores son:

- = Igual
- != Diferente
- <> Diferente
- < Menor que
- <= Menor o igual que
- > Mayor que
- >= Mayor o igual que

Las sentencias aritméticas también pueden restringir una selección. Los operadores aritméticos son:

- + Adición
- Resta
- \* Multiplicación
- / División
- \*\* Exponenciación
- Mod Módulo

Los operadores booleanos también restringen las operaciones de selección. Los valores booleanos son tres: verdadero, falso, desconocido (desconocido es el mismo que nulo).

Los dos operadores booleanos *and* y *or* combinan dos valores para producir un valor verdadero, falso o desconocido. El operador *not* invierte el valor de verdadero o falso.

Los operadores booleanos se pueden combinar con operaciones aritméticas y relacionales. Un ejemplo es:

```
SELECT *
FROM pedidos
WHERE emb_tasa != 0.85 AND emb_tasa > 0.50
```

Cláusula order by. Por definición, los registros de una tabla no se encuentran ordenados. La cláusula order by devuelve los registros de una selección en orden ascendente o descendente. La sintaxis es la siguiente:

```
SELECT *
FROM {nombre-de-tabla | nombre-de-vista}, ...
[WHERE condición-de-búsqueda]
[ORDER BY {nombre-de-columa [ASC | DESC]} ...]
```

Una consulta se puede ordenar utilizando más de una columna, así como el orden ascendente o descendente. Un ejemplo de la cláusula order by es:

```
SELECT *
FROM pedidos
WHERE pedi_num >= 1 AND pedi_num <5
ORDER by pedi_num, emb_cost DESC
```

Proyección. La selección toma filas de una tabla, la *proyección* toma columnas. La lista de selección en la sentencia select produce una proyección.

```
SELECT lista de selección
FROM {nombre-de-tabla | nombre-de-vista}, ...
[WHERE condición-de-búsqueda]
[GROUP BY {nombre de columna}]
[ORDER BY {nombre-de-columa [ASC | DESC]} ...]
```

La proyección selecciona alguna de las columnas de una tabla, pudiendo también reordenar la apariencia de las columnas en una tabla nueva. Como podemos ver en la sintaxis, las proyecciones siempre van acompañadas de las selecciones. En la sentencia `select` se seleccionan las columnas que el usuario quiere obtener y con esto se hace una proyección de una tabla, la cláusula `where` selecciona las filas que se quieren obtener de esas columnas.

La cláusula `Distinct`. Los contenidos de los registros de datos pueden determinar los resultados de una selección. Se pueden agrupar registros distintos basados en sus contenidos.

La palabra reservada *distinct* elimina los registros duplicados de los resultados de una selección. La sintaxis es la siguiente:

```
SELECT [distinct] lista-de-selección
FROM [nombre-de-latabla | nombre-de-vista], ...
[WHERE condición de búsqueda]
[GROUP BY {nombre-de-columna}]
[ORDER BY {nombre-de-columna90}]
```

Agregación. Existen cinco funciones agregadas en SQL. Estas funciones operan sobre las filas recuperadas de una selección. El resultado es un único valor.

Las cinco funciones de agregación de SQL son:

- **Maximum.** Devuelve el valor máximo encontrado entre las filas seleccionadas.
- **Minimum.** Devuelve el valor mínimo encontrado entre las filas seleccionadas.
- **Sum.** Devuelve la suma de los valores de las filas seleccionadas.
- **Count.** Devuelve el contador de las filas seleccionadas.

- Average. La media aritmética de las filas devueltas.

Group By. La cláusula group by conjunta datos seleccionados en subgrupos. Un dato se recupera de cada uno de los subgrupos. La sintaxis es la siguiente:

```
SELECT *
FROM [nombre-de-latabla | nombre-de-vista], ...
[WHERE condición de búsqueda]
[GROUP BY {nombre-de-columna}]
[ORDER BY {nombre-de-columna90}]
```

Una cláusula group by puede dividir los registros seleccionados en grupos y permitir a las funciones agregadas operar sobre esos grupos.

Unión. Una unión acepta dos tablas como entrada una produce una tercera (diferente) como salida. La operación de unión combina las filas de un tabla con las filas de otra para producir una nueva tercera tabla.

Las tablas deben ser compatibles para poderse combinar, siguiendo las siguientes condiciones:

- Cada una de las tablas debe de ser del mismo grado.
- Los atributos de cada una de las tablas proceden de los mismos dominios.
- Los atributos del mismo dominio deben aparecer en el mismo orden en cada una de las cabeceras de las tablas.

Estas tres condiciones aseguran que la unión de cómo resultado una relación. Sin estas reglas, una operación de unión produce un conjunto pero no una relación. La sintaxis es la siguiente:

```
SELECT *
FROM [nombre-de-latabla | nombre-de-vista], ...
[WHERE condición de búsqueda]
[GROUP BY {nombre-de-columna}]
[ORDER BY {nombre-de-columna [ASC | DESC]} ...]
[UNION sentencia-select]
```

**Intersección:** La intersección de dos tablas contiene cualquier fila ubicada en ambas tablas a la vez. El operador de intersección no es parte del estándar ANSI, no estando en todas las implementaciones de SQL. El operador de intersección combina dos a más sentencias select. La sintaxis es la siguiente:

```
SELECT *
FROM [nombre-de-latabla | nombre-de-vista], ...
[WHERE condición de búsqueda]
[GROUP BY {nombre-de-columna}]
[ORDER BY {nombre-de-columna [ASC | DESC]} ...]
[INTERSECT sentencia-select]
```

Al igual que en la unión, las dos tablas deben ser compatibles para formar una intersección.

**Diferencia:** El operador diferencia combina dos sentencias select. El resultado es una tabla que resulta de restar los elementos de una de los elementos de otra. La diferencia puede realizarse solo entre tablas compatibles. No todas las implementaciones de SQL soportan el operador diferencia.

La sintaxis es la siguiente:

```
SELECT *
FROM [nombre-de-latabla | nombre-de-vista], ...
[WHERE condición de búsqueda]
[GROUP BY {nombre-de-columna}]
[ORDER BY {nombre-de-columna [ASC | DESC]} ...]
[MINUS sentencia-select]
```

Join: Un join combina cada una de la fila de una tabla con las filas coincidentes de una segunda tabla. La cláusula *from* selecciona las tablas para el join. La cláusula *where* muestra como se combinan las tablas. La sintaxis es la siguiente:

```
SELECT *
FROM [nombre-de-latabla | nombre-de-vista], ...
OUTER [nombre-de-latabla | nombre-de-vista], ...
[WHERE condición de búsqueda]
[GROUP BY {nombre-de-columna}]
[ORDER BY {nombre-de-columna [ASC | DESC]} ...]
```

Incremento de filas (insert). La sentencia insert incrementa los registros de una tabla.

```
INSERT INTO (nombre-de-tabla | nombre-de-vista)
[({nombre-de-columna}, ...)]
VALUES({literal | NULL }, ...) |
Sentencia-select
```

Una sentencia insert que tiene una cláusula values inserta una única fila.

Una cláusula select puede obtener registros de una tabla e insertar copias en la tabla destino, puede proyectar columnas para la inserción de tabla destino. También se puede restringir la cláusula select.

Por ejemplo:

```
INSERT INTO destino
SELECT a.c1, b.c2, c.c3
FROM a, b, c
WHERE condición-de-búsqueda
```

**Borrado de filas (delete).** La sentencia delete borra filas de una tabla. Su sintaxis es:

```
DELETE FROM {nombre-de-tabla | nombre-de-vista}
[WHERE condición-de-búsqueda]
```

Si se omite la cláusula where se borrarán todas las filas de la tabla.

**Modificación en filas (update).** La sentencia update modifica la información de las filas de una tabla. La sintaxis es:

```
UPDATE {nombre-de-tabla | nombre-de-vista}
SET {nombre-de-columna = expresión | NULLE}
[WHERE condición-de-búsqueda]
```

Una sentencia update que omita la cláusula where cambiará todos los registros de una tabla, o bien a nulos, o bien a un valor dado.

**Vistas.** Una vista crea una tabla virtual ensamblando una o más tablas existentes. Una vez que la vista se ha creado, se podrá acceder a ella como a una tabla.

Una vista se puede crear para que presente al usuario la información que necesita.

También suministra protección de datos. La sintaxis es:

```
CREATE VIEW nombre-de-vista
{{{nombre-de-columna-de-vista}, ...}}
```

Se puede recorrer una vista para seleccionar u ver diferentes datos seleccionados.  
Se pueden eliminar vistas con DROP VIEW nombre-de-vista

Cualquier sentencia select puede crear una vista; por lo que también una vista puede combinar selecciones, proyecciones y joins.

## **2.14 ACTORES EN UN SISTEMA DE BASES DE DATOS.**

Son muchas las persona que participan en el diseño, mantenimiento y uso de una base de datos. Por un lado, están los que emplean en forma cotidiana la base de datos (la información que ésta contiene) y, por otro lado, las personas encargadas de mantener el entorno del sistema de base de datos.

### **ADMINISTRADOR DE LA BASE DE DATOS.**

Una de las principales razones para usar un DBMS es tener control centralizado de los datos, así como de los programas que acceden a dichos datos. La persona que tiene ese control central sobre el sistema es el administrador de las bases de datos (DBA). Es el responsable de administrar los recursos de la base de datos, el DBMS y el software relacionado con éste.

La funciones principales de DBA son:

- Definir y modificar el esquema de la base de datos y las restricciones de los datos.
- Crear y modificar la estructuras de almacenamiento y definir los métodos de acceso.

- Autorizar el acceso a la base de datos, coordinar y controlar tales accesos.
- Garantizar el funcionamiento del sistema y prestar servicio técnico: Se ocupa de los problemas de violación de seguridad del sistema de bases de datos o de respuesta lenta del sistema.
- Definir y poner en práctica planes adecuados de copias de seguridad (bakckups) del contenido de la base de datos.
- Adquirir los recursos necesarios de software y hardware.
- Especificar las restricciones de integridad de los datos.

#### DISEÑADORES DE LA BASE DE DATOS.

El diseñador de la base de datos identifica los datos que se almacenarán y elige las estructuras adecuadas, para representar y guardar dichos datos. Estas tareas se realizan antes de que se implemente la base de datos.

Interactúan con los futuros usuarios o grupos de usuarios de una base de datos, para comprender sus necesidades, desarrollar una vista de la base de datos que satisfaga los requisitos de cada grupo, y construir un diseño final que cumpla con las necesidades de todos los grupos.

El presente proyecto se enfoca a este tipo de actores, ya que es este rol el que desarrolla las actividades que lo concierne.

## USUARIOS FINALES.

Un objetivo primordial de un sistema de bases de datos es proporcionar un entorno para recuperar información de ella y almacenar nueva información en la base de datos. Existen cuatro tipos distintos de usuarios de sistemas de bases de datos, diferenciados por la forma en que esperan interactuar con el sistema.

### *Analistas de Sistemas y Programadores de Aplicaciones.*

Los profesionales informáticos interactúan con el sistema por medio de llamadas en DML, las cuales están incorporadas en un programa de aplicación. Los analistas determinan los requisitos de los usuarios finales y desarrollan especificaciones de conjuntos de operaciones (transacciones programadas) que satisfagan esos requisitos. Los programadores implementan esas especificaciones en forma de programas de aplicación, las prueban, depuran, documentan y mantienen.

### *Usuarios sofisticados.*

Los usuarios sofisticados interactúan con el sistema si escribir programas. Conocen los recursos del DBMS para satisfacer sus necesidades. Hacen consultas a la base de datos desde una terminal utilizando un lenguaje de consulta para explorar los datos en la base de datos.

### *Usuarios especializados.*

Utilizan bases de datos personalizadas mediante una aplicación comercial o paquete de software.

*Usuarios ingenuos.*

Este tipo de usuarios interactúan con el sistema invocando a uno de los programas de aplicación permanente.

### CAPÍTULO 3

#### PROCESO DE CREACIÓN DE BASES DE DATOS

### 3.1 CICLO DE VIDA DE UN SISTEMA DE INFORMACIÓN ORIENTADO A BASES DE DATOS.

Las bases de datos se han convertido en una parte importante de la gestión de recursos de información (en general, los sistemas de información) en muchas organizaciones, como consecuencia de las ventajas que conlleva su uso.

Cuando se crea una base de datos pequeña, utilizada por pocos usuarios, el diseño no resulta muy complicado. Pero cuando se diseñan bases de datos medianas o grandes para el sistema de información de una organización, el proceso es complejo pues el sistema de información debe satisfacer las necesidades de muchos usuarios diferentes (de 25 a más de 100).

Una base de datos mediana o grande contiene millones de bytes de información y sobre ella se realizan cientos de consultas y programas de aplicación. Los sistemas para este tipo de bases de datos son los sistemas de procesamiento de transacciones. En estos sistemas es decisivo el promedio de transacciones ejecutadas por minuto, así como el tiempo medio y máximo de respuesta del sistema (a cada transacción). Es imprescindible realizar un adecuado diseño de la base de datos tal que:

- Satisfaga las necesidades de procesamiento de la organización (tiempos de respuesta y de procesamiento, espacio de almacenamiento).
- Permita y facilite la evolución del sistema (como consecuencia de cambios en los requisitos).

El sistema de base de datos suele formar parte de un sistema de información mucho mayor. Con el que la organización controla sus recursos de información, es decir, los datos en sí mismos, el DBMS, el hardware y los medios de almacenamiento, el personal que utiliza

los datos (DBA, usuarios finales, etc.), el software de aplicación que accede a los datos y los actualiza, y los programadores que crean estas aplicaciones.

FASES DEL CICLO DE VIDA DE UN SISTEMA DE INFORMACIÓN ORIENTADO A BASE DE DATOS.

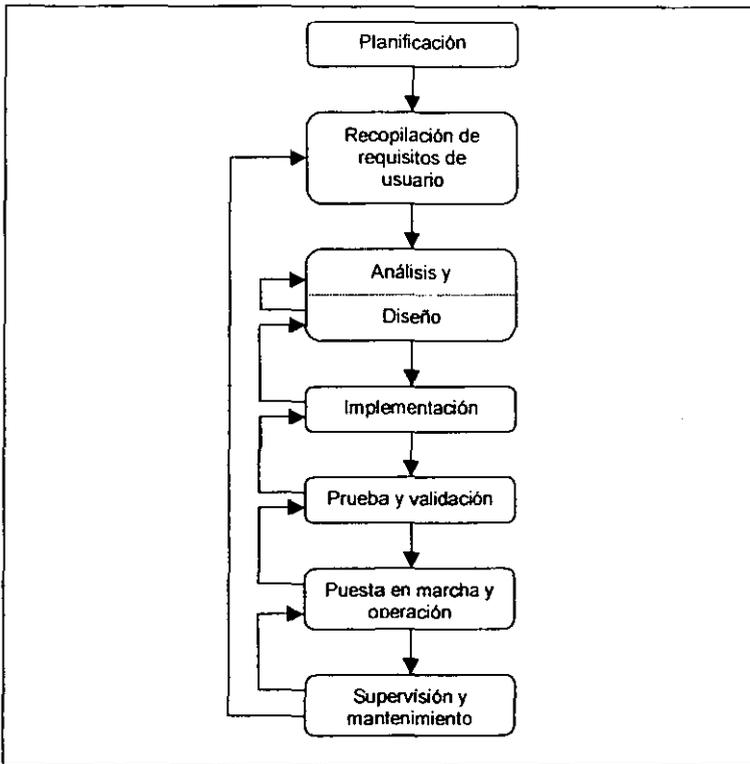


Fig. 3.1 Fases del ciclo de vida de un sistema de información orientado a base de datos

1. *Análisis de factibilidad, definición del sistema y objetivos, plan de trabajo.* Esta etapa consiste en analizar las posibles áreas de aplicación, estudiar la relación costo-beneficio, establecer prioridades entre las aplicaciones, etc. Por definición del sistema, se entiende establecer el alcance del sistema de base de datos, sus usuarios, etc.

2. *Recopilación de requisitos de usuario.* Se recogen los requisitos de los usuarios con el fin de identificar sus problemas y sus necesidades en lo referente a datos, funcionalidades, rendimiento, etc.
3. *Diseño (análisis y diseño).* Diseño del sistema de base de datos y los sistemas de aplicación (programas) que utilizan los datos y los procesan.
4. *Implementación, carga o conversión de datos y aplicaciones.* Implementación del sistema de información, carga de datos reales en la base de datos (nuevos u otros ya existentes convertidos al formato adecuado).
5. *Pruebas, validación y ajuste.* Prueba de las aplicaciones (o transacciones, operaciones sobre los datos a través de los programas). Un sistema queda validado cuando satisface los requisitos de los usuarios (especificaciones del comportamiento del sistema y los criterios de rendimiento).
6. *Operación.* Puesta en marcha del sistema. La fase operativa comienza cuando todas las funciones de sistema están disponibles y validadas.
7. *Supervisión y mantenimiento.* Durante la fase de operación, el sistema es vigilado continuamente. Cuando surgen nuevas necesidades de datos o se requieren nuevas aplicaciones, pasan por todas las fases anteriores hasta, una vez validadas, ser incorporadas al sistema.

### **3.2 EL PROCESO DE DISEÑO DE BASES DE DATOS.**

El proceso de diseño de base de datos es el proceso de diseñar la estructura lógica y física de una o más bases de datos para satisfacer las necesidades de información de los usuarios de una organización, para un conjunto definido de aplicaciones.

Los objetivos del diseño de bases de datos son los siguientes:

- a) Satisfacer requisitos de contenido de información de los usuarios y aplicaciones.
- b) Proporcionar una estructuración de los datos natural y fácil de entender.
- c) Apoyar los requisitos de procesamiento (objetivos de rendimiento como tiempo de respuesta, tiempo de procesamiento, espacio de almacenamiento, etc.).
- d) Conseguir un esquema de base de datos flexible, es decir, tal que sea posible modificarlo (como consecuencia de cambios en los requisitos del sistema) fácilmente una vez implementada la base de datos.

#### FASES DEL DISEÑO DE BASE DE DATOS.

Las fases del diseño de bases de datos son:

1. Recopilación de requisitos de usuarios.
2. Diseño conceptual
3. Elección del DBMS
4. Diseño Lógico
5. Diseño Físico
6. Implementación.

En algunas ocasiones se considera que la fase 1 y 6 no forman parte del diseño de la base de datos sino del ciclo de vida del sistema en general, sin embargo en este trabajo de seminario se tomará como parte del diseño.

El proceso de diseño consta de dos actividades paralelas:

- a) Diseño del contenido de datos y de la estructura de la base de datos, y
- b) Diseño del procesamiento de la base de datos y las aplicaciones de software.

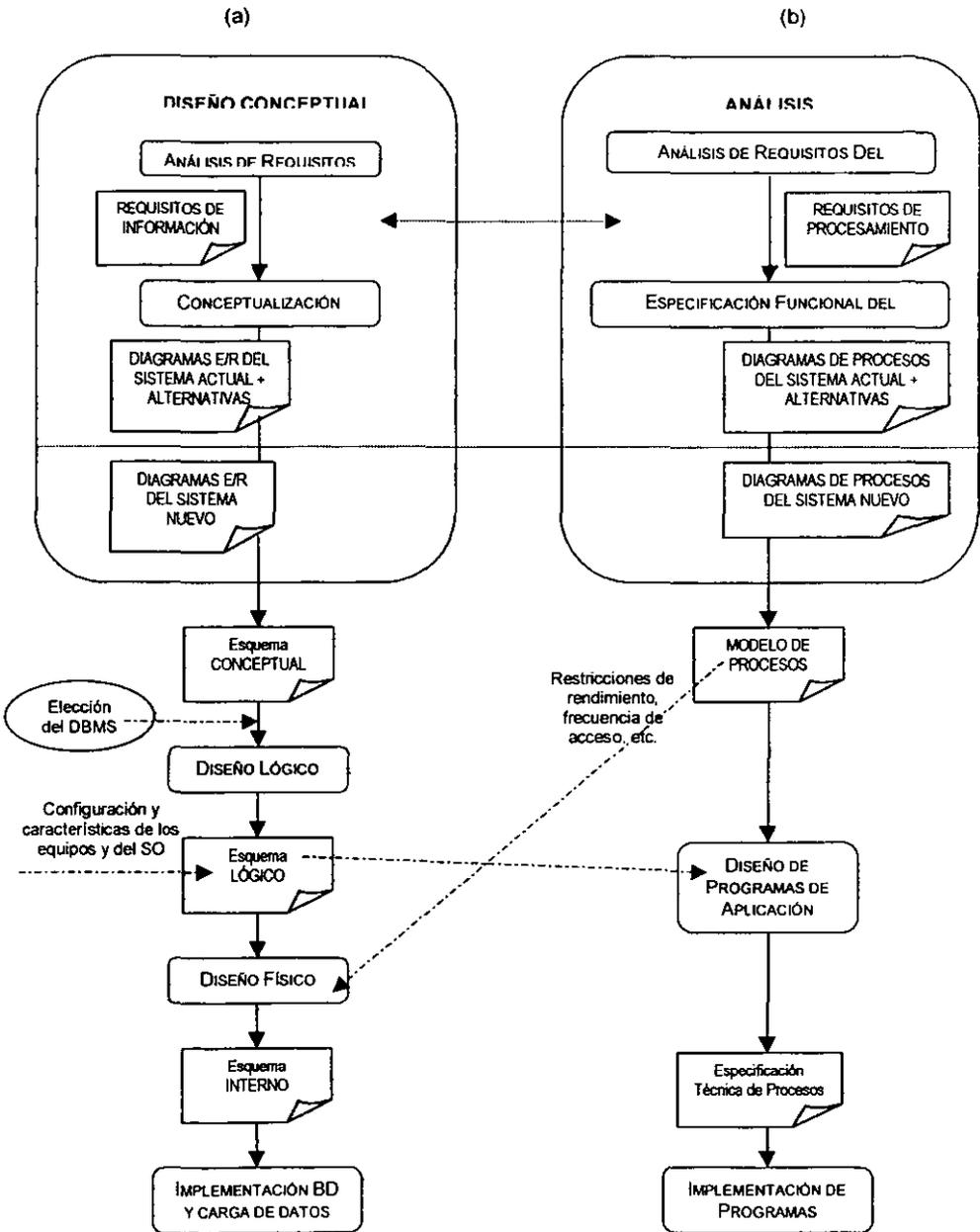


Fig. 3.1 Actividades del Proceso de Diseño

## DISEÑO DEL CONTENIDO DE DATOS Y DE LA ESTRUCTURA DE LA BASE DE DATOS.

*Diseño Conceptual.* Su objetivo es obtener una buena representación (descripción) de los requisitos de información del sistema, es decir, el esquema conceptual de la base de datos, que es independiente de los usuarios de aplicaciones, de DBMS específicos y de equipos informáticos.

Para describir este esquema (el contenido de información de la base de datos) se utiliza un modelo de datos de alto nivel, como el modelo E/R.

*Diseño Lógico.* En esta fase, el esquema conceptual, expresado en el modelo de datos de alto nivel, es transformado en un esquema lógico expresado en:

- a) El modelo de datos del DBMS elegido (diseño lógico dependiente del sistema).
- b) El modelo de datos lógico, por ejemplo si se decide utilizar algún DBMS relacional, pero no se elige ninguno en particular (diseño lógico independiente del sistema, pero dependiente del modelo de datos).

En términos de la arquitectura del DBMS en tres niveles, de esta fase se obtiene un esquema conceptual en el modelo de datos elegido, además de esquemas externos (vistas) para aplicaciones específicas.

*Diseño Físico.* Su objetivo es conseguir la implementación del esquema lógico más eficiente posible. Esta fase está muy relacionada con la del diseño lógico, pues las decisiones tomadas durante el diseño pueden afectar a la estructura del esquema lógico.

El resultado de esta fase es el esquema interno (en términos de la arquitectura del DBMS en tres niveles), que depende del DBMS, configuración y características de los equipos y sistemas operativos elegidos, y que describe la implementación de la base de datos en almacenamiento secundario (es decir, las estructuras de almacenamiento físicas, colocación de registros y caminos de acceso utilizados para acceder eficientemente a la información almacenada).

Una vez completadas estas fases, el esquema lógico y el físico se expresan utilizando el DDL (Data Definition Language) del DBMS destino, con el que ya puede crearse la base de datos, para posteriormente realizar pruebas, etc.

#### DISEÑO DEL PROCESAMIENTO DE LA BASE DE DATOS Y DE LAS APLICACIONES DE SOFTWARE.

*Análisis Funcional del Sistema.* Se basa en los requisitos de procesamiento y produce un modelo de procesos (a veces conocidos como esquema funcional), que es una descripción de alto nivel de las actividades (operaciones, tareas, funciones, etc.) desarrolladas dentro de la organización y de la información utilizada en cada actividad, intercambiada entre actividades, etc.

En esta fase, la base de datos suele ser considerada como un conjunto de simples depósitos de información, perdiéndose la visión de los datos como recurso global de la organización (visión que sí proporciona el esquema conceptual).

*Diseño de Programas de Aplicación.* Produce descripciones de alto nivel (especificaciones) del comportamiento de las aplicaciones, es decir, la estructura de los programas, cómo acceden éstos a qué información de la base de datos, etc.

*Implementación.* Produce las especificaciones detalladas de los programas de aplicación y quizás, el código (expresado en un lenguaje de programación determinado) de los programas.

Tradicionalmente los métodos de diseño de bases de datos se han centrado en una sola de éstas actividades, de forma que podía hablarse de diseño de bases de datos guiados por los datos (data-driven) o guiado por los procesos (function-driven).

Actualmente existe un consenso de que debe realizarse ambas actividades de forma coordinada, de hecho:

1. El modelo de procesos (esquema funcional) y el esquema conceptual deben ser mutuamente consistentes (sin conflictos entre ellos) y completos, puesto que todos los datos requeridos por las funciones deben estar representados en el esquema conceptual, y todas las operaciones requeridas por la base de datos deben estar reflejadas en las funciones del sistema.
2. EL diseño físico de la base de datos depende de las aplicaciones que van a utilizar los archivos de la base de datos (forma y frecuencia de acceso a los datos, restricciones de rendimiento, etc.)
3. En el diseño de los programas de aplicación se hace referencia a los elementos que tiene el esquema lógico de la base de datos.

)

### 3.3 MÉTODOS DE DISEÑO DE BASES DE DATOS.

El diseño de bases de datos es una tarea larga y costosa; durante mucho tiempo fue considerada una tarea de expertos, sin embargo, en la actualidad, numerosos autores reconocen que el diseño ha de llevarse a cabo siguiendo procedimientos ordenados y metódicos.

En distintas áreas de la ingeniería de software se han realizado grandes esfuerzos para encontrar los métodos de diseño más adecuados. Y es que el uso de un método tiene un importante impacto en el desarrollo de un producto de software (costos y plazos de entrega, calidad y mantenimiento, etc.)

Un buen diseño es la clave de una eficiente ingeniería de software. Un sistema de software bien diseñado es comprensible y fiable, fácil de aplicar y mantener. Un sistema de software mal diseñado puede funcionar, pero será costoso de mantener, difícil de probar y poco fiable.

#### CONCEPTO Y FASES DE UN MÉTODO DE DISEÑO.

*Concepto de Método.* Según Adoración de Miguel y Pittini: "Un método es un conjunto de modelos de datos, lenguajes y otras herramientas que facilitan la representación de los datos en cada fase del proceso de diseño de una base de datos junto con las reglas que permiten el pase de una fase a la siguiente".

Una herramienta es "cualquier recurso particular de disposición del método, para realizar las operaciones que en él se prevén". Los modelos de datos, los lenguajes de datos y la documentación sin herramientas.

Un modelo de datos es "un conjunto de conceptos, reglas y convenciones que permiten describir y manipular los datos de la parte del mundo real que constituye el universo del discurso". El esquema obtenido al describir cierto universo de discurso mediante las construcciones de un modelo de datos, será la visión del mundo real que tiene el diseñador, que lo ve en el contexto del sistema de información que está creando.

El lenguaje de datos es el resultado de definir una sintaxis sobre un modelo de datos; permite expresar un esquema (por ejemplo basado en el modelo relacional) en una sintaxis concreta (por ejemplo la de SQL).

La documentación permite describir los resultados de cada etapa, facilita el trabajo de los integrantes del equipo de diseño, la comunicación entre ellos y la revisión y mantenimiento de la base de datos.

Las reglas actúan sobre los elementos de entrada de cada fase del diseño, para conseguir las salidas; permiten el paso de una fase a la siguiente.

Cuando se habla de otras herramientas, se suele hacer referencia a herramientas software:

- De tipo CASE (Computer Aided Software Engineering: Ingeniería de software asistida por computadora, que permiten el diseño conjunto de esquemas y aplicaciones de bases de datos).
- De adquisición de requisitos.
- De diccionario de datos.

El método de diseño propuesto por Adoración de Miguel y Piattini consiste en tres grandes fases.

*Fases de un método de diseño de bases de datos.*

Cada una de las fases del método corresponderá a cada etapa en el proceso de diseño de bases de datos, a las que ya se ha hecho referencia.

Diseño Conceptual. Su objetivo es obtener una representación de los requisitos de información de la organización, independiente de usuarios, aplicaciones y rendimiento o eficiencia del equipo informático, es decir, el esquema de datos conceptual (EC). Las etapas del diseño conceptual son dos:

- a) **Análisis de Requerimientos**. En esta etapa se analizan las especificaciones de requisitos, para identificar aquellos relacionados con la información manipulada por el sistema. En esta etapa es necesario:
  - Eliminar las ambigüedades de los requisitos recabados.
  - Completar los requisitos, debido a que los usuarios transmiten sus necesidades pero de manera vaga.

- Dotar a los requerimientos de una estructura.
- Entender exactamente el significado de todos los términos.

b) **Conceptualización.** En esta etapa se realiza un refinamiento y estructuración sucesivos del esquema percibido, para obtener el esquema conceptual de la base de datos. Para describir el esquema conceptual suele utilizarse un modelo de datos de alto nivel, de forma que el esquema conceptual sea independiente de características de implementación, y permita ver la información con todo su contenido semántico.

Se debe representar el contenido de la base de datos sin considerar no cómo será implementada esta información en el sistema real, ni la eficiencia de los programas que utilizarán la información.

El esquema conceptual debe ser independiente de la implementación del DBMS porque:

- La meta del diseño del esquema conceptual es el entendimiento completo de la estructura, semántica (significado), interrelaciones y restricciones de la base de datos.
- El esquema conceptual es una descripción del contenido de la base de datos. La elección del DBMS y las decisiones de diseño posteriores pueden cambiar, pero eso no alterará el esquema conceptual si es independiente del DBMS.
- Es fundamental emplear un modelo de datos de alto nivel para describir el esquema conceptual, puesto que es más general y más expresivo que un modelo de datos de un DBMS en particular.

- El esquema conceptual puede servir como vehículo de comunicación entre usuarios, diseñadores y analistas. Los modelos de datos de alto nivel incluyen conceptos más fáciles de entender que los modelos de datos de nivel más bajo (específicos de un DBMS).

Existen dos enfoques para el diseño de esquemas conceptuales: el enfoque centralizado, en donde los requisitos de las diferentes aplicaciones y grupos de usuarios se combinan en un único conjunto (catálogo) de requisitos, antes de iniciarse el diseño, y el enfoque de vistas, en donde se diseña un esquema conceptual para cada grupo de usuarios que corresponde solo a sus requisitos propios; en este caso, es necesaria una etapa de integración de vistas en donde se combinen los esquemas para obtener un esquema conceptual global de la base de datos.

*Diseño Lógico.* Su objetivo es adaptar (traducir) el EC al modelo de datos de implementación obteniendo el esquema lógico de datos.

Algunos otros objetivos perseguidos con el diseño lógico son:

- Eliminar Redundancias
- Conseguir máxima simplicidad
- Evitar cargas suplementarias de programación

Estos tres objetivos se deben cumplir para poder conseguir una estructura lógica adecuada y un equilibrio entre los requisitos del usuario y la eficiencia.

En esta fase, el diseño lógico debe ser estructurado con la máxima portabilidad posible para facilitar su implementación sobre diferentes DBMS y migraciones entre versiones de un mismo DBMS. El diseño lógico está conformado por dos etapas:

a) Diseño Lógico estándar, en donde se realiza principalmente:

- Elección del modelo de datos. Se pasa del esquema conceptual al esquema lógico estándar (ELS)
- Uso de un Modelo Lógico de Datos Estándar (MLS): Relacional, red, jerárquico, etc.
- El esquema lógico estándar (ELS) es descrito mediante un lenguaje estándar del modelo de datos (por ejemplo, SQL, diagrama de estructura de datos, etc.)

b) Diseño Lógico Específico (DLE), en donde sus principales actividades son:

- Elección del DBMS específico
- Adaptación del esquema de BD a un DBMS concreto. Del Esquema lógico estándar se pasa al esquema lógico específico (ELE).
- Uso del modelo lógico de datos propios del DBMS elegido. (Sybase, Oracle, SQL Server, etc.)
- El esquema Lógico Específico (ELE) es descrito mediante un lenguaje DDL del DBMS específico.

Para poder llevar a cabo estas actividades se pueden utilizar herramientas de tipo CASE que faciliten la construcción de modelos de datos tanto lógico estándar como el lógico específico.

En este proyecto de seminario se utiliza la herramienta CASE Erwin Platinum, en la que se construyen ambos modelos lógicos.

*Diseño Físico.* Su objetivo primordial es conseguir una implementación del esquema lógico lo más eficiente posible.

Para poder conseguir esto, es necesario cumplir con las siguientes restricciones:

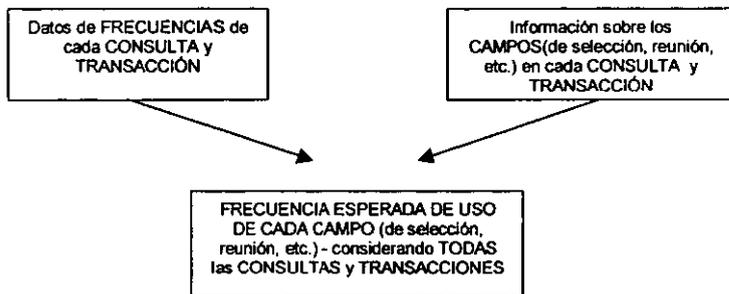
- Minimizar el tiempo de respuesta. Hace referencia al tiempo entre la introducción de una instrucción y la obtención de la respuesta. Esto depende de:
  - a) Tiempo de acceso a la base de datos para obtener los datos que se necesitan. (Actividad controlada por el DBMS)
  - b) Carga del sistema, planificación de tareas del sistema operativo, retrasos de comunicación. (Actividad no controlada por el DBMS).
- Maximizar la productividad de las transacciones, es decir, el número promedio de transacciones que el sistema de base de datos puede procesar por minuto.
- Optimizar el aprovechamiento del espacio. Optimizar la cantidad de espacio ocupado por archivos de la base de datos y sus estructuras de acceso.

Los principales factores a considerar en el diseño físico de las bases de datos son:

1. Análisis de consultas y transacciones. Definir transacciones y consultas que se esperan ejecutar en la base de datos.
  - a) En consultas: Especificar: Archivos a los que accede, Campos sobre los que se especifica alguna condición de selección, campos sobre los que se especifica alguna condición de reunión o de enlace de registros de diferente tipo y campos cuyos valores obtiene la consulta.

b) En transacciones y actualizaciones se debe especificar: archivos que actualiza, operación que se realiza en cada archivo (inserción, modificación, eliminación), campos sobre los que se realiza alguna selección para las modificaciones y borrados, campos actualizados por una operación de modificación.

2. Análisis de frecuencia esperada de invocación de consultas y transacciones.



3. Análisis de restricciones de tiempo sobre consultas y transacciones. Las restricciones de tiempo influyen considerablemente sobre la asignación de prioridades adicionales a los campos candidatos para ESTRUCTURAS DE ACCESO.

4. Análisis de frecuencia esperada de operaciones de actualización. Los archivos que se modifican constantemente deben tener el mínimo número posible de estructuras o caminos de acceso.

### *Características de un Método de Diseño.*

- a) Claridad y comprensibilidad. Es imprescindible que distintos tipos de personas (usuarios, técnicos de sistemas, analistas, etc.) participen en el proceso de diseño; por tanto, el método debe ser lo suficientemente sencillo para ser entendido por diferentes tipos de usuarios.
- b) Capacidad de soportar la evolución de los sistemas. Un buen método de diseño soportará la evolución del sistema de información sin problemas, produciendo en sus distintas etapas esquemas evolutivos, de forma que cuando cambie el universo del discurso, sea posible modificar los esquemas para que recojan dichos cambios sin tener que realizar de nuevo el diseño completo de la base de datos. Para conseguir este objetivo es fundamental que el método proporcione la base para una buena documentación del sistema.
- c) Facilitar la portabilidad. La IEEE considera la portabilidad como "la facilidad con la que un producto de programación puede ser transferido de un sistema informático a otro, o de un entorno a otro"; es esencial para conseguir sistemas abiertos.

Un método que pretenda conseguir esquemas portables, utilizará los siguientes recursos:

1. Unas etapas de diseño independientes que permitan desviarse, en determinados momentos, hacia otro tipo de sistemas.
2. Una subfase diseño lógico estándar, entre el diseño conceptual y el diseño lógico-físico en el DBMS específico que se va a utilizar. Esta subfase permitirá disponer de un esquema relacional general, que podría traducirse posteriormente al modelo relacional específico.

Los propios DBMS comerciales suelen proporcionar sus productos para diversas plataformas diferentes, lo cual hace posible que sea posible portar o transferir las bases de datos de unos entornos a otros.

- d) Versatilidad respecto a tipos de aplicaciones. Un método no debe estar orientado a un tipo de aplicaciones concreto, sino que debe poder utilizarse en aplicaciones diversas.
- e) Flexibilidad (Independencia de la dimensión de los proyectos). Un método debe poder utilizarse tanto en proyectos grandes como pequeños. Para abordar ambos tipos de proyectos se utilizan modelos, herramientas y lenguajes análogos, aunque para proyectos grandes se complementen con otras técnicas, mientras que para diseños menos complejos, se tengan que simplificar algunas etapas del método propuesto.
- f) Rigurosidad. Es conveniente dar un carácter riguroso al método de diseño, apoyándose, siempre que sea posible en fundamentos teóricos formales.
- g) Adopción de estándares. Un método debería aplicar todos los estándares que, para la ingeniería de software en general y para las bases de datos en particular, recomiendan distintas organizaciones internacionales (como ISO, SCM, OSF, etc.).
- h) Automatización. Para que un método resulte útil debe poderse automatizar, aplicando herramientas de tipo CASE. Estas herramientas deben soportar todas las fases propuestas para el diseño de la base de datos.

En el proyecto del presente trabajo de seminario se seguirá la metodología antes descrita del ciclo de vida de un sistema de información orientado a base de datos, para llevar a cabo el diseño de la unificación de las bases de datos.

Dentro de la fase de diseño, se seguirá el enfoque del diseño del procesamiento de la base de datos y de las aplicaciones de software, aunque también se desarrollaran algunas actividades del diseño del contenido de datos y de la estructura de la base de datos que complementen de manera satisfactoria el diseño de la nueva base de datos unificada.

**CAPÍTULO 4**

**CASO PRÁCTICO**

**UNIFICACIÓN INSTITUCIONAL DE LAS BASES DE DATOS DE LA BOLSA MEXICANA DE VALORES**

## 4.1 PLANIFICACIÓN.

### ANTECEDENTES

En 1997, la BMV, en conjunto con la Dirección de Sistemas, ahora Bursatec, inició un proyecto para la unificación de las diferentes bases de datos operativas que se encontraban fuera de la plataforma TANDEM, concretamente sobre la plataforma Windows NT. Dicho proyecto dio como resultado la concentración de información residente en numerosos servidores en uno sólo, así como el establecimiento de estándares y políticas para controlar el crecimiento de la nueva base de datos denominada CONTEN.

Paralelamente al proyecto de base de datos, se realizó la renovación tecnológica, así como el desarrollo de nuevas aplicaciones, permitiendo el funcionamiento en producción tanto de la nueva base de datos CONTEN, como de la nueva generación de aplicaciones en mayo de 1998.

Desde entonces, y a pesar de contar con un ambiente en el que la interacción de los sistemas y aplicaciones es cada vez mayor, la depuración de información y la detección y corrección de excepciones administrativas tanto en CONTEN como en TANDEM, que han mermado la consistencia de la información entre ambos ambientes, ha sido una tarea constante.

Adicionalmente, el acceso a la información por parte de los usuarios se ha dado en dos formas principalmente:

- A través de las aplicaciones desarrolladas internamente.
- A través de archivos solicitados por los usuarios y generados por personal de Bursatec.

Por lo anterior, existe una dependencia de las áreas usuarias hacia Bursatec para la explotación de la información, sin dejar de mencionar que en ocasiones se desconoce la totalidad de la información con que se cuenta.

#### OBJETIVOS DEL PROYECTO

- Unificar las bases de datos CONTEN y TANDEM
  - Crear un modelo de datos único y ajustado a las reglas del negocio
  - Homologar criterios de almacenamiento
  - Eliminar duplicidad de información entre ambas plataformas
- Mejorar la calidad, consistencia y confiabilidad de la información
- Depuración de la información histórica con base en criterios válidos del negocio.
- Proporcionar a los usuarios nuevas alternativas de explotación de la información a través de herramientas que faciliten el análisis y la toma de decisiones

#### ALCANCE Y CONDICIONES GENERALES

1. Esta propuesta está basada en el documento de requerimientos recibido por Bursatec por parte de la Dirección de Vigilancia del Mercado y Estadística de la Bolsa Mexicana de Valores.
2. El proyecto no contempla las adecuaciones que tendrán que realizarse a todos los sistemas y aplicaciones como consecuencia de la generación del nuevo modelo de datos. Sin embargo se tendrá una estimación del impacto en dichos sistemas y aplicaciones.

3. Los periodos definidos en el calendario de actividades, se cumplirán si las prioridades de los proyectos solicitados a Bursatec no sufren modificación, de tal suerte, que sea necesario reasignar recursos de un proyecto a otro.

#### PLAN DEL PROYECTO.

La ejecución del proyecto se llevará a cabo de la siguiente manera:

##### Fase I. Análisis Funcional del Sistema.

1. Análisis y diseño del proyecto
  - a) Inventario
    - ⇒ Inventario de la Base de Datos en TANDEM
    - ⇒ Inventario de la Base de Datos CONTEN
  - b) Información por Área de Negocio
    - ⇒ Matriz Información-Producto
    - ⇒ Matriz Información-Sistema
    - ⇒ Matriz Tabla-Sistema
    - ⇒ Diagrama general de flujo de información a nivel institucional
  - c) Impacto en sistemas de la BMV
    - ⇒ Sistemas Involucrados
    - ⇒ Logística de alimentación y migración de aplicaciones a la base de datos institucional.
2. Determinar plataforma de software (DBMS y herramientas ETL)
3. Determinar la plataforma de hardware.

### Fase II. Diseño de Programas de Aplicación (Migración de Datos)

4. Generación del diccionario de datos CONTEN II.
5. Generación del modelo de datos CONTEN II
6. Definición de metadatos de cada una de las entidades de la nueva base de datos, incluyendo ciclos de actualización y dueños de la información.
7. Extracción y Limpieza de Datos.
  - a) Diseño de procesos de alimentación de la base de datos institucional por medio de la herramienta ETL.
    - ⇒ Documentos de diseño de procesos de alimentación por ETL
8. Transformación y transporte de datos.
  - a) Construcción de procesos de alimentación diseñados.
9. Mantenimiento de la base de datos institucional antes de la migración de aplicaciones.

### Fase III. Implementación (Explotación de Información)

10. Migración de aplicaciones a la base de datos institucional.
11. Administración soporte y mejora.
  - a) Definición de seguridad, periodicidad de respaldos, recuperación de la información y medidas necesarias para el control y administración de la base de datos institucional incluyendo el crecimiento de la misma.

A largo plazo:

12. Minería de Datos.
13. Procesamiento Analítico en Línea.

## EQUIPOS DE TRABAJO.

### *Bursatec:*

Puesto	Función
Gerente de área	Elaboración de modelos de procesos, administración de proyectos, propuesta de soluciones y tendencias tecnológicas. Análisis y Diseño.
Especialista	Seguimiento al análisis y diseño de nuevos modelos de datos, procesos de extracción, transformación y carga de información.
Analista	Análisis y diseño de entidades. Mantenimiento de los inventarios durante la ejecución del proyecto. Procesos de extracción, transformación y carga de información

### *Bolsa Mexicana de Valores:*

Roles	Función
Responsable del Proyecto	Aprobar productos entregados. Llevar el control general del proyecto, toma de decisiones que impacten el proyecto (técnicas, procedimientos y procesos de negocio).

## 4.2 RECOPIACIÓN DE REQUISITOS DE USUARIOS.

Los usuarios, personal de la Bolsa Mexicana de Valores han proporcionado a BURSATEC sus requerimientos generales los cuales son:

- Unificar las bases de información, mismas que se encuentran disgregadas en TANDEM y CONTEN, para una mejor administración.
- Almacenar la información por parte de las áreas operativas de la BMV de acuerdo a los siguientes criterios:
  - ⇒ Funcionalidad
  - ⇒ Consistencia

⇒ Seguridad

⇒ Integridad

- Creación de políticas de acceso, de mantenimiento y calidad de acuerdo a la línea de negocio, requerimientos de autoridades e imagen de la institución.
- Desarrollo de una herramienta para el eficaz manejo y explotación de la información.

Como podemos notar, los requerimientos especificados son no funcionales, debido a que BURSATEC se encuentra totalmente familiarizado con las bases de datos que operan en la Bolsa Mexicana de Valores y por lo tanto conoce más detalladamente el estatus en el que se encuentran ambas bases de datos, sus debilidades y fortalezas.

Para conocer y analizar las dos bases de datos (Tandem, SQL Server (Conten)), se realizó el inventario de datos de ambas. Esta actividad forma parte del análisis y diseño de la base de datos unificada.

#### **4.3 ANÁLISIS Y DISEÑO DE LA BASE DE DATOS UNIFICADA.**

El diseño de la base de datos unificada de la Bolsa Mexicana de Valores se llevará a cabo mediante el Diseño de bases de datos guiado por procesos (function-driven). Sin embargo también se tomará en cuenta el diseño de bases de datos guiados por los datos, al elaborar modelos de datos lógicos y físicos que encuadran el estudio realizado a través del diseño por procesos y que engloban los datos en un todo relacionado.

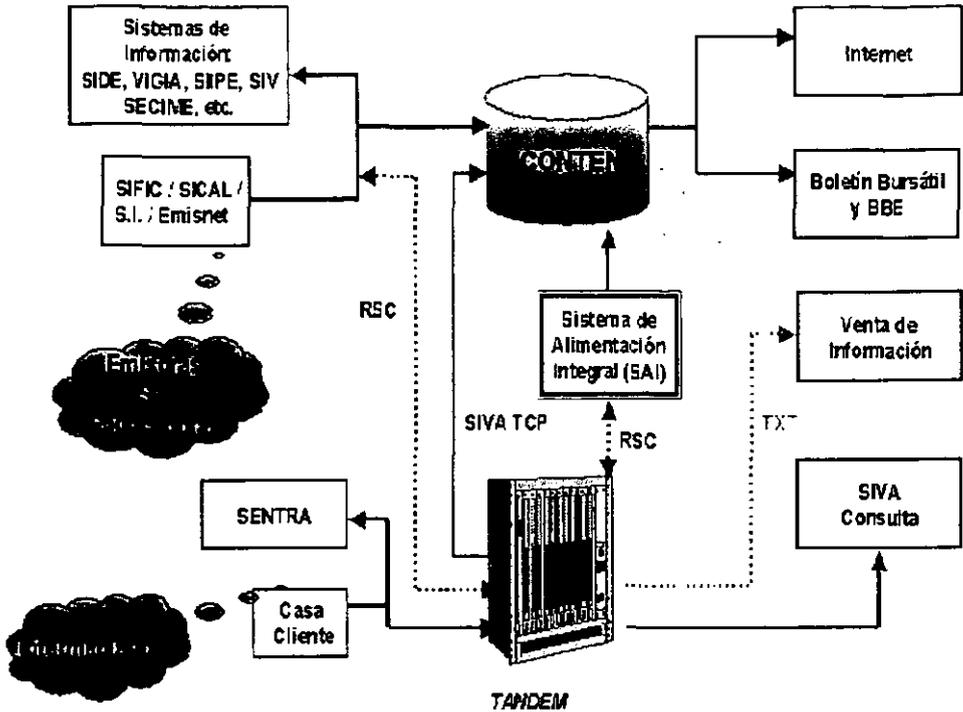
## ANÁLISIS FUNCIONAL DEL SISTEMA.

En esta etapa se analizará por completo la información, los procesos operativos, aplicaciones y bases de datos pertenecientes a la Bolsa Mexicana de Valores y que se encuentran actualmente operando en el ambiente de producción.

Durante el proceso de análisis ejecutado en esta etapa, se desarrollaron actividades generales que involucraron en algunos casos la generación de productos. Ambos elementos del análisis se listan a continuación:

- Planteamiento del problema, situación actual y objetivos del proyecto
- Definición de actividades con las áreas operativas de la BMV
- Levantamiento de información por área de negocio
- Generación de la Matriz Información – Producto
- Realización del inventario de las bases de datos actuales (TANDEM, CONTEN)
- Levantamiento de información de aplicaciones y sistemas
- Generación de la Matriz Información – Sistema y la Matriz Tabla – Sistema
- Generación del Diagrama de Flujo de información entre áreas, aplicaciones y bases de datos de la BMV
- Generación del diccionario de datos de la nueva base de datos institucional  
CONTENII
- Diseño del modelo de datos de la nueva base de datos institucional CONTENII
- Desarrollo de la logística de alimentación y migración de aplicaciones a CONTENII

Diagrama de Situación Actual.



Proceso de Levantamiento de Información por Área de Negocio.

Documentos y formatos utilizados:	Matriz de levantamiento de información Catálogo de sistemas de la BMV Catálogo de áreas e instituciones Catálogo de información Catálogo de publicaciones de la BMV
Descripción:	Proceso para la identificación de áreas generadoras y usuarias de información dentro y fuera de la BMV. En el se identifican los sistemas y personal de la institución que generan, procesan o utilizan la información, dónde se almacena, así como el producto o medio de difusión de la información.
Producto obtenido:	Matriz Información – Producto

Participantes:	Estadística (BMV) Inscripción de Valores (BMV) Administración de Valores y Emisoras (BMV) Control Operativo (BMV) Vigilancia del Mercado (BMV) Sistemas de Información (Bursatec)
----------------	--

*Realización del inventario de las bases de datos actuales.*

Documentos y formatos utilizados:	Documentación técnica de las aplicaciones Información de tablas de las bases de datos TANDEM y CONTEN
Descripción:	Levantamiento del inventario de tablas y atributos existentes en TANDEM y CONTEN. El área de Sistemas de Información de Bursatec concentró y generó en los casos necesarios la relación de las tablas, atributos y su correspondiente descripción, de acuerdo a los conceptos de negocio especificados por las áreas de desarrollo de Bursatec en la documentación técnica de sus aplicaciones.
Producto obtenido:	Inventario Tandem Inventario Conten SQL Server
Participantes:	Sistemas de Información (Bursatec) Desarrollo de Sistemas (Bursatec) Area de Base de datos (Bursatec)

*Proceso de levantamiento de información de aplicaciones y sistemas.*

Documentos y formatos utilizados:	Catálogo de sistemas de la BMV Matriz Información – Producto Documentación técnica de las aplicaciones
Descripción:	Proceso para la identificación de la información generada y utilizada por las aplicaciones y procesos de la BMV, así como de las tablas y atributos de TANDEM y CONTEN, donde se almacena dicha información.
Producto obtenido:	Matriz Tabla – Sistema

	Matriz Información – Sistema Matriz Tabla - Sistema Unificada
Participantes:	Sistemas de Información (Bursatec) Desarrollo de Sistemas (Bursatec)

*Generación del diagrama de flujo de información de la BMV.*

Documentos y formatos utilizados:	Catálogo de sistemas de la BMV Catálogo de áreas e instituciones Catálogo de información Matriz Información – Producto Matriz Información – Sistema
Descripción:	Creación del diagrama de flujo de información en la institución, mediante la relación de las instituciones, empresas, áreas o personal que la origina y desde que sistema o proceso lo realiza. Su principal objetivo es identificar las dependencias entre sistemas y áreas desde el punto de vista de la información. Es necesario también para identificar las aplicaciones críticas y el grado de complejidad en función de la información que obtienen o proveen a otros sistemas. Estos factores se tomarán en cuenta para determinar el orden teórico de migración de los mismos a CONTENII y los procesos de alimentación que en cada caso será necesario desarrollar para asegurar el correcto funcionamiento de cada uno de ellos.
Producto obtenido:	Diagrama de Flujo de Información en la BMV
Participantes:	Sistemas de Información (Bursatec)

*Logística de alimentación y migración de aplicaciones a la base de datos institucional unificada (Conten II).*

<p>Documentos y formatos utilizados:</p>	<p>Catálogo de sistemas de la BMV          Matriz Información – Producto          Matriz Tabla – Sistema          Matriz Información – Sistema          Diagrama de Flujo de Información en la BMV</p>
<p>Descripción:</p>	<p>Para el proceso de alimentación a CONTENII, se tomarán en cuenta los siguientes puntos:</p> <ul style="list-style-type: none"> <li>⇒ Considerar ambientes paralelos de desarrollo y producción para este proceso de migración con sus respectivos procesos de alimentación.</li> <li>⇒ Asegurar la alimentación de los catálogos principales de CONTENII.</li> <li>⇒ Cargar la información histórica por única vez por medio de procesos ETL.</li> <li>⇒ Alimentar la información intra-día necesaria en línea a través de SIVA TCP y procesos ETL, considerando la creación de procesos que generen la información no contemplada por el feed de la BMV.</li> </ul> <p>Para el proceso de migración de aplicaciones a CONTENII, se tomarán en cuenta los siguientes puntos:</p> <ol style="list-style-type: none"> <li>1. Las aplicaciones que actualizarán catálogos principales de CONTENII se migrarán al final.</li> <li>2. Se realizará un análisis particular para la migración de cada sistema, así como de los procesos de alimentación de la información utilizada por cada uno.</li> <li>3. Se clasificaron los sistemas de acuerdo a los siguientes criterios:             <ul style="list-style-type: none"> <li>C1. ¿El sistema alimentará catálogos principales en CONTENII?</li> <li>C2. ¿Cuántos sistemas dependen de la información que genera este sistema?</li> </ul> </li> </ol>

	<p>C3. ¿De cuantos sistemas que alimentarán a CONTENII depende la información que utiliza este sistema?</p> <p>C4. ¿Cuántas tablas actualiza este sistema?</p> <p>C5. ¿Cuántas tablas utiliza este sistema</p> <p>4. Crear grupos de aplicaciones que compartan tablas para ser migradas en conjunto</p> <p>5. ¿Cuál es el área de sistemas responsable de la aplicación?</p>
Producto obtenido:	Documento de orden teórico de migración de aplicaciones y clasificación de sistemas
Participantes:	Sistemas de Información (Bursatec)

*Determinar plataforma de software (DBMS y ETL) y hardware (Propuestas)*

Documentos y formatos utilizados:	NA
Descripción:	<p>Para determinar las plataformas de software se evalúan diferentes opciones bajo ciertos criterios.</p> <p>a) Manejador de la Base de Datos.</p> <ul style="list-style-type: none"> <li>➤ Capacidad para manejar las plataformas existentes (Tandem, SQL).</li> <li>➤ Manejo de consultas abiertas por el manejador y el equipo.</li> <li>➤ Acceso a herramientas de diseño.</li> <li>➤ Acceso desde herramientas OLAP.</li> <li>➤ Aprovechamiento de tecnología actual en la BMV</li> <li>➤ Renovación Tecnológica</li> <li>➤ Costos</li> </ul> <p>b) Herramientas ETL</p> <ul style="list-style-type: none"> <li>➤ Tiempo total de carga</li> <li>➤ Diferencia porcentual en tiempo de proceso</li> <li>➤ Funcionalidad</li> <li>➤ Rendimiento</li> <li>➤ Facilidad de uso</li> <li>➤ Tiempo de desarrollo</li> </ul>

	<ul style="list-style-type: none"> <li>➤ Costos</li> </ul> <p>1. Hardware:</p> <ul style="list-style-type: none"> <li>➤ Espacio requerido</li> <li>➤ Definición del CPU por el número de usuarios</li> <li>➤ Memoria</li> <li>➤ Dispositivos adicionales.</li> </ul> <p>De acuerdo a las evaluaciones realizadas a plataformas de hardware y software se plantearán dos propuestas distintas y se elegirá la más viable de acuerdo a las necesidades de la empresa.</p>
Producto obtenido:	<p>Cuadro comparativo plataforma hardware</p> <p>Documento de evaluación</p> <p>Matriz Evaluación de Manejadores</p> <p>Documento de Propuestas de Solución</p>
Participantes:	<p>Sistemas de Información (Bursatec)</p> <p>Soporte Técnico</p>

#### DISEÑO DE PROGRAMAS DE APLICACIÓN (MIGRACIÓN DE DATOS).

En esta etapa se diseña el modelo de datos y el diccionario de datos, como parte medular del proyecto. Se detallan las entidades ya existentes en ambas plataformas y se unifican las que contengan información común, formando una solo cumpliendo con el mayor número de formas normales (hasta donde lo permita la información que manejan).

Con esto se proporciona una visión general de la estructura de la base de datos unificada y del comportamiento de los datos a través de las relaciones establecidas entre las entidades que forman parte de la muestra.

### *Generación del diccionario de datos Contem II.*

Documentos y formatos utilizados:	Inventario de tablas y atributos de TANDEM y CONTEN
Descripción:	A partir de los inventarios de tablas y atributos de CONTEN y TANDEM, se realizó el análisis de la información duplicada, identificando a la vez aquellos atributos que no cumplen las reglas de negocio. A partir de la depuración, homologación y complemento de las entidades de información, se generó una nueva y única versión de las tablas y atributos que conformaran todo el universo de conceptos y datos existentes actualmente, dando como resultado el diccionario de datos de CONTENII.
Producto obtenido:	Documento de Tablas Equivalentes Diccionario de datos de CONTENII
Participantes:	Sistemas de Información (Bursatec)

### *Generación del Modelo de Datos Contem II*

Documentos y formatos utilizados:	Diccionario de datos de CONTENII
Descripción:	A partir del diccionario de datos, se identificó la relación existente entre las nuevas tablas, considerando el tipo de información que contendrán, así como los atributos que identifican a cada entidad como única. La relación lógica entre las nuevas tablas se realizó a partir de estos atributos, para quedar representada en el diagrama modelo de CONTENII.
Producto obtenido:	Modelo de datos de CONTENII
Participantes:	Sistemas de Información (Bursatec)

## IMPLEMENTACIÓN.

En esta etapa del diseño se realizará la definición de los metadatos tanto de las entidades como de los atributos, el diseño de los procesos de alimentación por medio de herramientas ETL.

*Definición de Metadatos y Diseño de procesos de alimentación por medio de herramientas ETL.*

Documentos y formatos utilizados:	Diccionario de datos de CONTENII Documento de Tablas Equivalentes Documento de control de tablas unificadas
Descripción:	A partir del modelo de datos generado y de la naturaleza de la información, se diseñarán los procesos de alimentación a la base de datos y de mantenimiento a la misma mientras se migran los istemas involucrados. Se elabora un documento que contiene los metadatos de las entidades, de los atributos y los procesos de alimentación y mantenimiento.
Producto obtenido:	Metadatos y Diseño Procesos de Alimentación
Participantes:	Sistemas de Información (Bursatec)

Con esta actividad se da por terminada la fase de Análisis y diseño del ciclo de vida de los sistemas de información orientado a bases de datos.

## **IMPLEMENTACIÓN CARGA O CONVERSIÓN DE DATOS Y APLICACIONES.**

El presente trabajo de seminario no abarca esta etapa debido a la magnitud del proyecto.

La gerencia de Estadística y Emisoras de la Bolsa Mexicana de Valores ya aprobó el proyecto debido a que la necesidad de renovar y ordenar las bases de datos es evidente, sin embargo los recursos monetarios no han sido aprobados aún, debido a los últimos cambios de presidencia y reestructuración administrativa de la empresa, por lo que no se ha podido avanzar en la implementación del proyecto.

## **PRUEBAS Y VALIDACIÓN.**

N/A

## **OPERACIÓN.**

N/A

## **SUPERVISIÓN Y MANTENIMIENTO.**

N/A

## CONCLUSIONES

La información es la parte más importante de cualquier organización. El éxito o fracaso que pueda tener una empresa depende directamente de la forma en que administre y explote su información. Las empresas que no saben aprovechar y controlar la información que generan, así como la que reciben, tienen un crecimiento pobre que se refleja en su competitividad y su posición en el mercado.

Para tener un mayor control sobre la información y una mayor explotación de la misma, las organizaciones recurren a sistemas de información automatizados. Estos sistemas de información manipulan la información, sin embargo ésta debe almacenarse en un DBMS que proporcione seguridad, integridad, veracidad y facilidad de acceso a la información. El DBMS debe estar contenido en un hardware apropiado que tenga la capacidad de almacenar la cantidad de información utilizada por la empresa y que proporcione herramientas de respaldo, recuperación y seguridad.

El proceso de diseño de la base de datos es un factor determinante en su eficiencia y buen funcionamiento, para lo cuál se debe seguir un proceso que permita diseñarla de forma que cubra con las necesidades de información de la empresa y con los requerimientos de seguridad, veracidad, control e integridad de los datos.

Al crear una base de datos, ya sea unificada o de nueva creación, es muy importante tomar en cuenta el alcance y el propósito final de la misma. En nuestros días es necesario diseñar bases de datos que sean capaces de soportar sistemas de información enfocados a toma de decisiones o estratégicos, para que éstas sean el motor de impulso de las empresas a promover el diseño de sistemas de información de alto nivel.

En este trabajo de seminario se describió un proceso de diseño de unificación de bases de datos basado en el enfoque del diseño del procesamiento de la base de datos y de las aplicaciones de software que permitió llevar un análisis por áreas de negocio y por grupos de información, en donde se estudió detalladamente cada una de las entidades existentes, los sistemas que las utilizan y la forma en que lo hacen (inserción, actualización, eliminación de datos), obteniendo con esto el comportamiento de las entidades en los diversos sistemas de información en los que participaban. Se atacó el problema desde el punto de vista de los procesos y sistemas existentes, desarrollando de forma integral, por áreas de negocio, el modelo de datos de la nueva base de datos unificada.

En el proceso de unificación de base de datos planteado en el capítulo 4 de este trabajo, cada una de las etapas del proceso son importantes para el diseño de la base de datos. Desde el inicio se debe recopilar información que describa la situación informática actual, debido a que ésta es la información de soporte de las siguientes fases. La parte medular del proceso es el análisis de cada una de las tablas para identificar aquellas que almacenen información equivalente, unificarlas cuidando no eliminar datos importantes al mismo tiempo que son normalizadas y definidas bajo estándares establecidos de diseño, generando el modelo de datos correspondiente. Los procesos de migración de datos diseñados basados en las herramientas de transporte de datos a utilizar, deben ser definidos cuidadosamente buscando no perder información importante durante el proceso.

El éxito de la implantación de una base de datos unificada, depende del buen análisis llevado a cabo en la fase de diseño, por lo que es importante establecer un proceso a seguir bien definido y plantear las actividades a realizar y los entregables que genera la realización de dichas actividades.

## BIBLIOGRAFIA

DE MIGUEL, A, PIATTINI, M, "Concepción y Diseño de Base de Datos: Del Modelo GIR",

Edit. RA-MA, México 1993 Primera Edición.

ULLMAN, D. Jeffrey, "Introducción a los Sistemas de Bases de Datos", Edit. Prentice Hall,

México 1999 Primera Edición.

TSAI, Alice, "Administración de Bases Datos. Administración y Uso", Edit. Prentice Hall,

México 1998 Primera Edición.

KENDALL, Kennet/ KENDALL, Julie, "Análisis y Diseño de Sistemas", Edit. Prentice Hall

Hispanoamericana, México D.F. 1997 Tercera Edición.

SENN, A. James, "Análisis y Diseño de Sistemas de Información", Edit. Mc.Graw Hill, México

D.F. 1992 Segunda Edición.

KORTH, F. Henry / SILBERSCHATZ, Abraham, "Fundamentos de Bases de Datos", Edit.

Mc.Graw Hill, Madrid España 1993 Segunda Edición.

LUCAS, Angel, "Diseño y Gestión de Sistemas de Bases de Datos", Edit. Paraninful, Madrid

España 1993 Primera Edición.

COHEN, Daniel / ASIN, Enrique, "Sistemas de Información para los negocios. Un enfoque de

toma de decisiones", Edit. Mc.Graw Hill, México D.F. 2000 Tercera Edición.

## REFERENCIAS DE INTERNET

<http://coqui.lce.org/mdejesus/CLAS3>

<http://gsync.escet.urjc.es/docencia/asignaturas/fisioterapia/transpas/node9.html>

<http://elies.rediris.es/elies9>

<http://www.itlp.edu.mx/publica/tutoriales/basedat2>