



*UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO*

*FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN*

*“PROPUESTA DE IMPLANTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE
BASES DE DATOS PARA EL MANEJO DE EMPLEADOS DE TELCEL”*

T E S I S

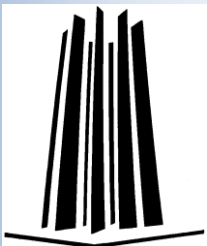
*QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN*

P R E S E N T A:

ALBERTO IVÁN TAPIA DURÁN

ASESORA: INGENIERA NORMA RAQUEL SOTO ARREDONDO

MÉXICO 2010





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.

Dedicatorias y Agradecimientos

A DIOS

Por darme salud y fuerzas para seguir adelante, por permitirme llegar a la culminación de mi carrera, por tener a mi lado a mi madre, mi esposa y mis hijos; y así lograr uno de mis sueños más importantes, al verme en esta etapa de mi vida.

Por tener a mi familia unida, y juntos celebrar este triunfo que es suyo también.

A MI MADRE, REGINA

A quien le debo mi vida, mi educación y superación. No tengo palabras para agradecerte todo lo que has hecho por mí, como reconocerte todo lo que me has dado desde que nací, mi educación tanto moral como escolar, tus consejos que siempre me han servido, pues gracias a aquellas palabras que en ocasiones fueron regaños y castigos han logrado que yo llegará hasta aquí.

A pesar de mis derrotas nunca me diste la espalda, siempre me has extendido la mano, aunque en ocasiones te decepcione, tú siempre encontraste las palabras para orientarme y guiarme de nuevo al camino, volviste a confiar en mí, y no perdiste la esperanza de verme convertido en un profesionista.

Te estaré eternamente agradecido por nunca escatimar esfuerzos ni sacrificios, por tu apoyo tanto moral como económico para llegar al término de un ciclo más en mi vida, por tu esfuerzo para hacer de mí un hombre de provecho y por la dedicación que desde siempre recibí.

Con toda mi gratitud y especial cariño.

A MI ESPOSA, ANA

Te doy mil gracias por todo lo que me has dado y sobre todo, por la confianza que me has transmitido día con día con tan sólo haber creído en mí.

Gracias por la infinita paciencia y apoyo que me brindaste en todo momento, para culminar una de mis más grandes metas y por permitirme robarte mucho del tiempo en el que merecía estar contigo.

Con todo mi amor, este TÍTULO es también tuyo.

A MIS HIJOS, JESÚS ALBERTO E IRVING AXEL

Les dedico este Título a ustedes mis pequeños traviesos, a quienes con el corazón lleno de tristeza, tuve que robarles horas de convivencia, cuidados y juegos para poder terminar y culminar mi carrera, agradeciéndoles que a cambio al verme, bríllen sus ojos y corran con alegría hacia mí, brindándome sonrisas y mil besos, demostrándome así su gran cariño y amor. Para ustedes hijos, que desde que nacieron son mi mayor ilusión, mi valentía, mi fuerza, mi alegría, la razón de mi vida !!!

Con mi más grande amor.

A MI ASESORA, ING. NORMA SOTO

Quien me dedico su valioso tiempo. Mil gracias por el apoyo y los consejos que me dio cuando más lo necesitaba. Con cariño, respeto y admiración por haberme orientado; así mismo le agradezco su amistad que bien ha sabido brindarme.

Muchas Gracias !!!



CONTENIDO

INTRODUCCIÓN

CAPÍTULO 1: REPORTE DEL DIPLOMADO “ADMINISTRACIÓN DE BASES DE DATOS”

1.1. SISTEMAS DE INFORMACIÓN Y EL MODELO DE DATOS RELACIONAL

1.1.1 DATO

1.1.2 INFORMACIÓN

1.1.3 ANÁLISIS DE LA INFORMACIÓN

1.1.4 DEFINICIÓN DE UNA BASE DE DATOS

1.1.5 ARQUITECTURA PARA LAS BASES DE DATOS

1.1.6 CARACTERÍSTICAS DE LAS BASES DE DATOS

1.1.7 MODELO DE DATOS

1.1.7.1 MODELO JERÁRQUICO

1.1.7.2 MODELO DE RED

1.1.7.3 MODELO RELACIONAL

1.1.7.4 MODELO ORIENTADO A OBJETOS

1.1.8 ARQUITECTURA CLIENTE/SERVIDOR

1.1.9 MODELADO RELACIONAL

1.1.9.1 REGLAS DE CODD

1.1.9.2 BASES DE DATOS RELACIONALES

1.1.9.3 LENGUAJE DE CONSULTA RELACIONAL

1.1.10 INDEPENDENCIA DE DATOS Y LA INTEGRIDAD REFERENCIAL

1.1.10.1 INDEPENDENCIA DE DATOS

1.1.10.2 INTEGRIDAD REFERENCIAL

1.1.11 MODELO ENTIDAD – RELACIÓN

1.1.12 NORMALIZACIÓN

1.2. SISTEMAS MANEJADORES DE BASES DE DATOS RELACIONALES (RDBMS)

1.2.1 RDBMS Y SU FUNCIÓN

1.2.2 COMPONENTES DE UN RDBMS

1.2.3 SQL ANSI 89, 92, 99 Y 2003

1.2.4 PRINCIPALES SISTEMAS MANEJADORES DE BASES DE DATOS RELACIONALES

1.2.4.1 MICROSOFT SQL SERVER

1.2.4.2 SYBASE



- 1.2.4.3 ORACLE
- 1.2.4.4 INFORMIX Y DB2
- 1.2.4.5 POSTGRE SQL
- 1.2.4.6 MYSQL
- 1.2.4.7 COMPARACIÓN ENTRE RDBMS'S
- 1.2.5 CONSIDERACIONES DE HARDWARE
- 1.2.6 TIPOS DE USUARIOS EN UNA BASE DE DATOS

1.3. SQL (STRUCTURES QUERY LANGUAGE)

- 1.3.1 DEFINICIÓN Y COMPONENTES DE SQL
- 1.3.2 DEFINICIÓN DE DATOS
 - 1.3.2.1 DEFINICIÓN DE DATOS DEL SISTEMA
 - 1.3.2.2 EL VALOR NULO
 - 1.3.2.3 TABLAS
 - 1.3.2.4 MODIFICACIÓN DE TABLAS
 - 1.3.2.5 TIPOS DE LLAVES E ÍNDICES
- 1.3.3 MANIPULACIÓN DE DATOS
 - 1.3.3.1 SELECCIÓN DE DATOS
 - 1.3.3.2 INSERCIÓN DE DATOS
- 1.3.4 ESTRUCTURAS DE CONTROL DE FLUJO

1.4. FUNDAMENTOS DE SISTEMAS OPERATIVOS

- 1.4.1 SISTEMA OPERATIVO Y SUS COMPONENTES
- 1.4.2 SOFTWARE LIBRE (OPEN SOURCE)
- 1.4.3 UNIX
 - 1.4.3.1 FAMILIAS DE UNIX
- 1.4.4 GNU/LINUX
- 1.4.5 WINDOWS SERVER
- 1.4.6 DIFERENCIAS DE SERVIDORES WINDOWS Y SERVIDORES LINUX

1.5. HABILIDADES DIRECTIVAS PARA ADMINISTRADORES

- 1.5.1 ESTILOS DE COMUNICACIÓN
- 1.5.2 LIDERAZGO
 - 1.5.2.1 TIPOS DE LIDERAZGO
- 1.5.3 PRESENTACIONES EFECTIVAS
- 1.5.4 REUNIONES EFECTIVAS

1.6. ADMINISTRACIÓN DE BASES DE DATOS



- 1.6.1 TAREAS DEL ADMINISTRADOR
- 1.6.2 CONCURRENCIA
- 1.6.3 RESPALDO DE INFORMACIÓN
- 1.6.4 RECUPERACIÓN
- 1.6.5 CREACIÓN E INSTALACIÓN DE LA BASE DE DATOS
- 1.6.6 PROPIETARIOS Y USUARIOS DE UNA BASE DE DATOS

1.7. BUENAS PRÁCTICAS EN LA FUNCIÓN DE LA ADMINISTRACIÓN

- 1.7.1 PROCESOS
- 1.7.2 ESTÁNDARES
- 1.7.3 AUDITORIA INFORMÁTICA

1.8. SEGURIDAD EN BASE DE DATOS

- 1.8.1 SEGURIDAD DE LA INFORMACIÓN
- 1.8.2 ATAQUES Y SUS DIFERENTES TIPOS
- 1.8.3 SEGURIDAD EN BASES DE DATOS

1.9. PERFORMANCE AND TUNNING

1.10. MODELO ORIENTADO A OBJETOS

- 1.10.1 PARADIGMA ORIENTADO A OBJETOS
- 1.10.3 LENGUAJE DE MODELADO UNIFICADO (UML)

1.11. TÓPICOS AVANZADOS DE BASES DE DATOS

- 1.11.1 MINERÍA DE DATOS
- 1.11.2 DATAWAREHOUSE
- 1.11.3 BASES DE DATOS MULTIDIMENSIONALES
- 1.11.4 BASES DE DATOS INTELIGENTES

CAPÍTULO 2: PLANTEAMIENTO DEL PROBLEMA Y POSIBLE SOLUCIÓN

- 2.1 HISTORIA DE LA TELECOMUNICACIÓN EN MÉXICO Y DE LA EMPRESA TELCEL
- 2.2 SISTEMA ADMINISTRADOR DE NÓMINA PARA TELCEL

CAPÍTULO 3: IMPLANTACIÓN DEL SISTEMA



3.1 FUNCIONES DEL SISTEMA

3.2 TABLAS DE LA BASE DE DATOS DE TELCEL

CONCLUSIONES

GLOSARIO



INTRODUCCIÓN

La gran diversidad, volumen e importancia, tanto económica como estratégica, que tiene hoy en día la información en cualquier organización (ya sea ésta una institución pública o privada), implica el uso de sistemas manejadores de bases de datos para garantizar su seguridad, consistencia, integridad y accesibilidad, entre otros factores.

Esto ha creado la inherente necesidad de contar con profesionales de las tecnologías de la información, que tengan los conocimientos y habilidades requeridas para administrar en forma eficaz y eficiente, tan valioso recurso.

La Universidad Nacional Autónoma de México, a través de la Dirección General de Servicios de Cómputo Académico, imparte el Diplomado de Administración de Base de Datos, con el objetivo de formar profesionales que cuenten con los elementos necesarios para realizar una adecuada y productiva administración de Sistemas Manejadores de Bases de Datos Relacionales (RDBMS).

OBJETIVO GENERAL

El participante identificará y aplicará los ejemplos de los RDBMS, desde su modelado, hasta su implementación, optimización, administración y mantenimiento en un servidor; asimismo, comprenderá la trascendencia de las normas ANSI SQL en cada una de las versiones disponibles, y reconocerá las principales tareas del administrador, para llevarlas a la práctica.

DIRIGIDO A:

Profesionales de las áreas de cómputo e informática o afines que requieran diseñar y crear bases de datos, así como administrar Sistemas Manejadores de Bases de Datos y manejar grandes volúmenes de información, para la oportuna toma de decisiones que requiere cualquier organización.

ALCANCES:

El participante diseñará, implementará y administrará sus propias bases de datos, así mismo podrá seleccionar la plataforma de trabajo adecuada a sus necesidades, con base en las características del RDBMS; y establecer las políticas de seguridad y planes de contingencia, sustentándolos con lo aprendido en el diplomado.



El presente informe, describe los objetivos, actividades, ejercicios, prácticas y proyectos, así como los conceptos básicos que componen cada módulo de este Diplomado. Éstos son los módulos y sus actividades principales:

1. SISTEMAS DE INFORMACIÓN Y EL MODELO DE DATOS RELACIONAL

El participante identificará todos los elementos que conforman una base de datos y la diferenciará del manejo de archivos de dato; será capaz de conceptualizar hechos del mundo real como un modelo de datos, utilizando la teoría relacional para su representación.

2. SISTEMAS MANEJADORES DE BASES DE DATOS RELACIONALES (RDBMS)

El participante identificará la función, los componentes de un RDBMS y las normas que debe de cumplir internacionalmente para su elección en el entorno de trabajo, así como su importancia en la actualidad.

3. SQL (Structured Query Language)

El participante utilizará el lenguaje SQL para extraer la información necesaria en el desarrollo de las aplicaciones en su entorno de trabajo.

4. FUNDAMENTOS DE SISTEMAS OPERATIVOS

El participante identificará los comandos y herramientas básicas de administración que intervienen en la configuración de un servidor en diferentes plataformas en las cuales puede residir e RDBMS.

5. HABILIDADES DIRECTIVAS PARA ADMINISTRADORES

El participante conocerá los principios básicos para la administración de la actividad laboral de equipos de trabajo, así como la planeación del tiempo y actividades.

6. ADMINISTRACIÓN DE BASES DE DATOS

El participante analizará las principales tareas que debe cubrir el perfil de un administrador de bases de datos, así como la manera de llevarlas a cabo dentro del RDBMS.



7. BUENAS PRÁCTICAS EN LA FUNCIÓN DE LA ADMINISTRACIÓN

El participante conocerá los fundamentos de la auditoría informática y controles, en el ámbito de la administración de bases de datos, así como las mejores prácticas a seguir en la administración de bases de datos.

8. SEGURIDAD EN BASE DE DATOS

El participante distinguirá las principales vulnerabilidades en seguridad en su entorno de trabajo lo cual reforzará, tomando las medidas necesarias para su corrección y el total aprovechamiento de los elementos y bondades ofrecidos por los RDBMS.

9. PERFORMANCE AND TUNNING

El participante interpretará el funcionamiento del servidor y definirá el RDBMS a través de la optimización de las tareas realizadas por éste.

10. MODELO ORIENTADO A OBJETOS

El participante definirá las principales características de la metodología orientada a objetos, como lo son: su modelado y su enfoque dentro de las empresas y organizaciones.

11. TÓPICOS AVANZADOS DE BASES DE DATOS

El participante conocerá algunas de las tendencias actuales en el manejo de la información.

En el capítulo 2 “Planteamiento del problema y posible solución”, se describe el problema al cual se enfrentan las empresas para tener una buena base de datos para el departamento de nómina. Se da una solución a este problema y se indican las tablas que contendría la base de datos del departamento de nómina de Telcel.

En el capítulo 3 “Implantación del sistema”, se describe el funcionamiento del sistema que administra y maneja la base de datos del departamento de nomina de Telcel y se indica el contenido de las tablas.



CAPÍTULO 1: REPORTE DEL DIPLOMADO “ADMINISTRACIÓN DE BASES DE DATOS”

1.1. SISTEMAS DE INFORMACIÓN Y EL MODELO DE DATOS RELACIONAL

1.1.1 Dato

Un dato es la unidad mínima de información, hecho sin evaluar ó un valor sin significado.

1.1.2 Información

La información se obtiene asociando los hechos en un contexto determinado, es decir, la adición o el procesamiento de los datos, proporcionan el conocimiento o entendimiento de ciertos factores.

Conjunto de datos interrelacionados entre sí de forma que ante una entrada proporcionada por un mundo exterior produce una respuesta (salida).

En un sentido más amplio, la información es un conjunto de componentes que interactúan entre sí para lograr un objetivo común para la toma de decisiones.

Características del valor de la información:

- **Accesible:** es la facilidad y rapidez con que se obtiene la información resultante.
- **Clara:** se refiere a la integridad y entendimiento de la información sin ambigüedades.
- **Precisa:** que sea lo más exacta posible.
- **Propia:** debe de haber relación entre el resultado y lo solicitado por el usuario.
- **Oportuna:** menor duración del ciclo (entrada, procesamiento y entrega al usuario).
- **Flexible:** adaptabilidad de la información a la toma de decisiones.
- **Verificable:** que se pueda examinar la información.
- **Imparcial:** no se puede alterar o modificar la información (sólo por el dueño).
- **Cuantificable:** todo dato procesado produce información.

1.1.3 Análisis de la información

El análisis de la información es un modelo de datos que consiste en la representación conceptual de la problemática que se desea resolver y cuya característica primordial es la claridad de su contenido.



Algo importante de mencionar es que el costo de producción de la información se puede medir gracias a los dispositivos y medios utilizados, pero la información es conceptual por naturaleza y no tiene características tangibles salvo representaciones simbólicas.

1.1.4 Definición de una base de datos

Una base de datos es un conjunto de datos relacionados entre sí con un objetivo común almacenados en algún lugar; es una colección de tablas interrelacionadas. El contenido de una base de datos engloba la información concerniente de una organización, de tal manera que los datos estén disponibles para los usuarios en tiempo real y son compatibles con usuarios concurrentes, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de una base de datos son el hardware, el software RDBMS y los datos a manejar, así como el personal encargado del manejo del sistema.

1.1.5 Arquitectura para las bases de datos

Hace unos años atrás, las bases de datos eran el resultado de una compleja programación y de complicados mecanismos de almacenamiento. La popularización de la informática y la aparición de aplicaciones específicas trajeron la disponibilidad de herramientas de gestión de datos que dieron lugar a los denominados Sistemas de Gestión de Bases de Datos, identificados por sus siglas SGBD por su acrónimo en inglés DBMS (Data Base Management Systems).

Los componentes de un sistema de base de datos, son los siguientes:

- **Hardware.** En donde se encuentra toda la infraestructura física de la base de datos.
- **Software.** Todos los programas aplicativos que se involucran para generar un ambiente idóneo.
- **Datos.** La información en sí.
- **Usuarios.** Cualquier persona que interactúe con la base de datos (DBA, DBO, Usuarios Finales).

Las bases de datos respetan la arquitectura de tres niveles definida, para cualquier tipo de base de datos, por el grupo ANSI/SPARC. En esta arquitectura la base de datos se divide en los niveles externo, conceptual e interno.

- **Nivel interno:** Es el nivel más bajo de abstracción y define cómo se almacenan los datos en el soporte físico, así como los métodos de acceso.
- **Nivel conceptual:** Es el nivel medio de abstracción, se trata de la representación de los datos realizada por la organización, que recoge las vistas parciales de los



requerimientos de los diferentes usuarios y las aplicaciones posibles, se configura como visión organizativa total e incluye la definición de datos y las relaciones entre ellos.

- **Nivel externo:** Es el nivel de mayor abstracción, a este nivel corresponden las diferentes vistas parciales que tienen de la base de datos los diferentes usuarios, en cierto modo, es la parte del modelo conceptual a la que tienen acceso.

La figura 1.1 nos muestra un esquema más genérico que nos permite ilustrar los niveles dentro de la arquitectura de las bases de datos.

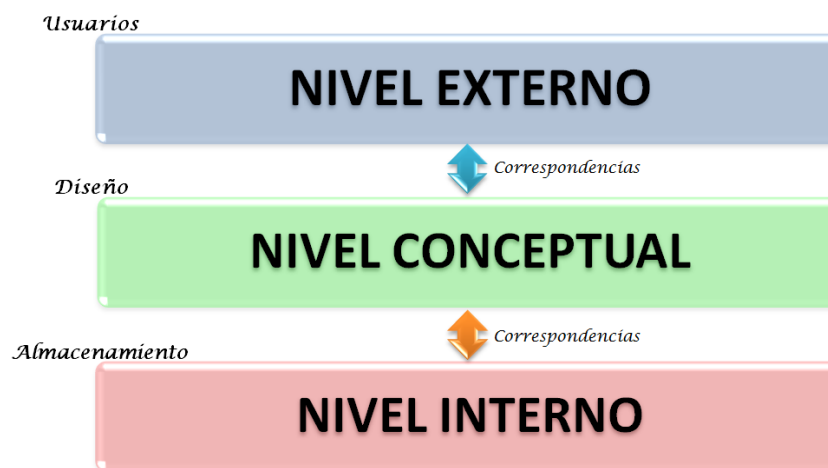


Figura 1.1 Niveles de arquitectura de una base de datos

1.1.6 Características de las bases de datos

- **Redundancia:** La redundancia de datos se refiere, a la existencia de información repetida o duplicada innecesaria para las bases de datos.
- **Consistencia:** La consistencia es la evaluación de las reglas del negocio, esto es, verificar que los datos estén siguiendo dichas reglas.
- **Integridad:** La integridad es la facultad de poder implementar los mecanismos necesarios para que los datos guarden consistencia. Debemos tomar en cuenta que hay diferentes tipos de integridad como la de campo, la de entidad, la de clave y la referencial.

La integridad de las bases de datos se puede lograr mediante:



- El mantenimiento de una redundancia mínima y controlada.
 - El establecimiento de llaves primarias o índices primarios.
 - La validación de las dependencias entre tablas relacionadas.
 - La creación de reglas de validación durante la inserción y edición de datos.
-
- **Seguridad:** Hoy en día se considera a la información de una empresa como uno de los activos más valiosos e importantes, por lo que la seguridad de la misma es muy importante. La seguridad implica asegurar que los usuarios están autorizados para llevar a cabo lo que tratan de hacer.

1.1.7 Modelo de datos

Un modelo es la representación de la realidad que contiene las características generales de algo que se va a realizar.

Un modelo de datos es una colección de herramientas conceptuales para descubrir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Los modelos de datos se dividen en tres grupos, los modelos lógicos basados en objetos, los modelos lógicos basados en registros y los modelos físicos de datos.

Los modelos lógicos basados en objetos se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo representamos los datos de tal forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente.

Los modelos lógicos basados en registros se utilizan para describir datos en los niveles conceptual y físico. Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de la implementación.

Los modelos físicos de datos se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos. Existen dos clasificaciones de este tipo que son: Modelo unificador y Memoria de elementos.

Los cuatro modelos de datos más ampliamente aceptados son:



- Modelo jerárquico.
- Modelo de red.
- Modelo relacional.
- Modelo orientado a objetos.

1.1.7.1 Modelo Jerárquico

La forma de representar los modelos y datos es por medio de registros y sus ligas. Como se muestra en la figura 1.2, la diferencia radica en que están organizados por conjuntos de árboles en lugar de graficas arbitrarias.

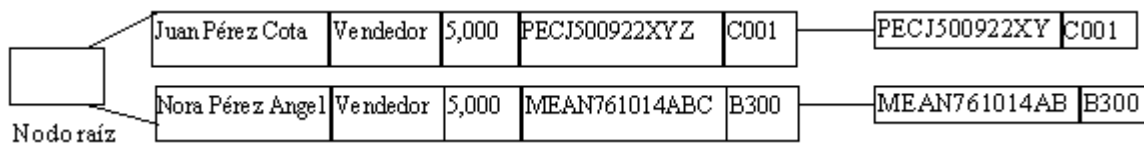


Figura 1.2: Modelo Jerárquico

En este tipo de modelos la organización se establece en forma de árbol, donde la raíz es un nodo ficticio. Así tenemos que, una base de datos jerárquica es una colección de árboles de este tipo.

1.1.7.2 Modelo de Red

Este modelo representa los datos mediante colecciones de registros, sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Lo anterior se aprecia en la figura 1.3.

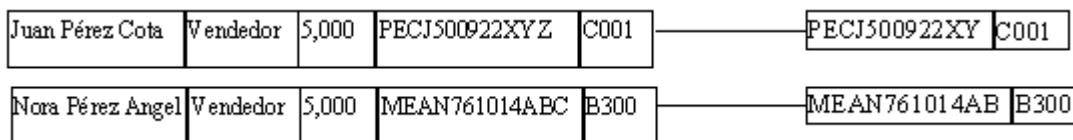


Figura 1.3: Modelo de Red

Una base de datos de red como su nombre lo indica, está formado por una colección de registros, los cuales están conectados entre sí por medio de enlaces. El registro es similar a una entidad como las empleadas en el modelo entidad – relación.

1.1.7.3 Modelo Relacional



En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla. En el modelo relacional el único elemento de representación es la relación. Como se ilustra en la figura 1.4.

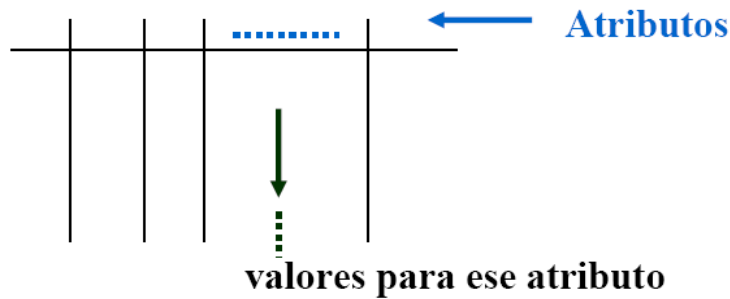


Figura 1.4 Representación de una relación.

1.1.7.4 Modelo orientado a Objetos

El modelo de bases de datos orientado a objetos es una adaptación a los sistemas de bases de datos. Se basa en el concepto de encapsulamiento de datos y código que opera sobre estos en un objeto. Los objetos estructurados se agrupan en clases. El propósito de los sistemas de bases de datos es la gestión de grandes cantidades de información. Las primeras bases de datos surgieron del desarrollo de los sistemas de gestión de archivos. Estos sistemas primero evolucionaron en bases de datos de red o en bases de datos jerárquicas y, más tarde, en bases de datos relacionales.

En la figura 1.5 podemos observar los tipos y subtipos del modelo de datos.

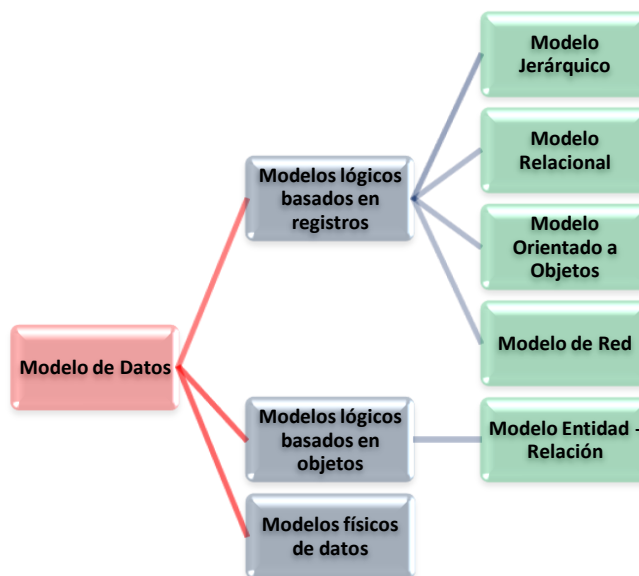




Figura 1.5 Tipos y subtipos del modelo de datos

1.1.8 Arquitectura Cliente/Servidor

Podemos entender el término cliente-servidor como un sistema en el que una máquina cliente solicita a una segunda máquina llamada servidor que ejecute una tarea específica, el cliente suele ser una PC conectada a una red LAN y el servidor, como un servidor de archivos PC o un servidor de archivos UNIX.

El modelo Cliente / servidor se define como la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o a cualquier otro recurso dentro del grupo de trabajo y/o a través de la empresa en diferentes plataformas. En la figura 1.6 podemos observar la como es la relación cliente – servidor.

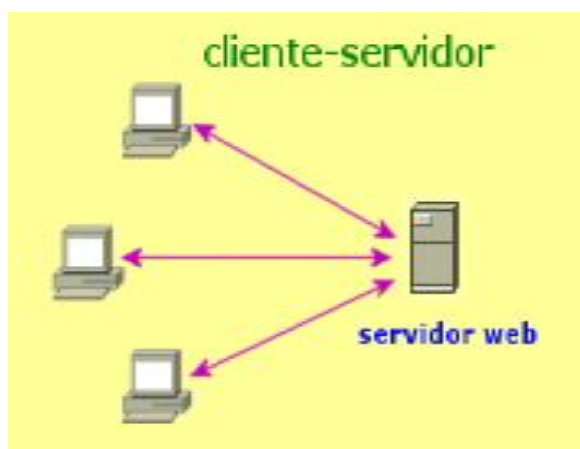


Figura 1.6 Relación Cliente – Servidor

Los clientes interactúan con el usuario, usualmente de forma gráfica y a su vez se comunican con los procesos auxiliares que se encargan de establecer la conexión con el servidor, de enviar el pedido, de recibir la respuesta, manejar las fallas y realizar actividades tanto de sincronización como de seguridad.

Los servidores proporcionan un servicio al cliente y devuelven los resultados, en algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente.



Una de las cosas por la que se distingue el modelo Cliente/servidor, es que su procesamiento es distribuido entre aplicaciones independientes.

1.1.9 Modelado Relacional

En el modelo relacional los datos se almacenan, al menos conceptualmente, de un modo en que los usuarios entienden con mayor facilidad.

Los datos se almacenan y las relaciones entre las filas y las tablas son visibles en los datos. Este enfoque permite a los usuarios obtener información de la base de datos sin asistencia de sistemas profesionales de administración de información.

La tabla 1.1 hace una síntesis de la evolución del Modelo Relacional, desde su surgimiento a fines de la década de los sesenta hasta la actualidad.

Año	Sucesos
1968	Surge el Modelo Relacional (Codd)
1970	Desarrollos teóricos, algebra relacional
1973	Prototipos (Sistema R)
1979	Oracle
1981	SQL
1982	Sybase, Informix
1984	SQL / ANS
1986	SQL ISO
1990	Modelo Relacional Versión 2 (Codd)
1992	SQL 2 estándar
1994 – actualidad	SQL 3 aún no estandarizado BD00

Tabla 1.1 Evolución del Modelo Relacional

El trabajo publicado por Codd en 1970 presentaba un nuevo modelo de datos que perseguía una serie de objetivos, que se resumen en las siguientes líneas:

- **Independencia física.** El modo en el que se almacenan los datos no influye en su manipulación lógica y por tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico.
- **Independencia lógica.** El añadir, eliminar o modificar objetos de la base de datos no repercute en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).
- **Flexibilidad.** En el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera.



- **Uniformidad.** Las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
- **Sencillez.** Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil de comprender y de utilizar por parte del usuario final.

En el enfoque relacional es sustancialmente distinto de otros enfoques en términos de sus estructuras lógicas y del modo de las operaciones de entrada/salida. En el enfoque relacional, los datos se organizan en tablas llamadas relaciones, cada una de las cuales se implanta como un archivo. En terminología relacional una fila en una relación representa un registro o una entidad. Cada columna en una relación representa un campo o un atributo.

Así, una relación se compone de una colección de entidades cuyos propietarios están descritos por cierto número de atributos predeterminados implantados como campos.

Esencialmente el Modelado Relacional tiene cuatro partes importantes:

- El Modelado Conceptual
- El Modelado Lógico
- La Normalización
- El Modelo Físico

1.1.9.1 Reglas de Codd

Cualquier RDBMS que proclame ser relacional, deberá manejar, completamente, las bases de datos por medio de sus capacidades relacionales.

- **Regla de información.** Toda la información dentro de una base de datos relacional se representa de manera explícita a nivel lógico y exactamente de una sola manera, como valores en una tabla.
- **Regla del acceso garantizado.** Se garantiza que todos y cada uno de los datos (valor atómico) en una base de datos relacional pueden ser leídos recurriendo a una combinación del nombre de la tabla, valor de la llave primaria y nombre de la columna.
- **El manejo sistemático de los valores nulos.** En un DBMS totalmente relacional se soportan los valores nulos (que son distintos de una cadena de caracteres vacía o de una cadena con caracteres



en blanco o de cero o cualquier otro número), para representar información faltante o no aplicable de una forma consistente, independientemente del tipo de dato.

- Catálogo dinámico en línea basado en un modelo relacional. La descripción de la base de datos se representa en el nivel lógico de la misma forma que los datos ordinarios, de tal suerte que los usuarios autorizados puedan aplicar el mismo lenguaje relacional para consultarla, que aquél que emplean para con sus datos habituales.
- Regla del sub-lenguaje de datos completos. Se debe contar con un sub-lenguaje que contemple la definición de datos, la definición de vistas, la manipulación de datos, las restricciones de integridad, la autorización, el inicio y fin de una transacción.
- Regla de actualización de vistas. Todas las vistas que teóricamente sean actualizables deberán ser actualizadas por medio del sistema.
- Inserción, actualización y eliminación de alto nivel. La posibilidad de manejar una relación base o una relación derivada como un sólo operador se aplica a la lectura, inserción, modificación y eliminación de datos.
- Independencia física de los datos. Los programas de aplicación y la actividad en terminales no deberán ser afectados por cambios en el almacenamiento físico de los datos o en el método de acceso.
- Independencia lógica de los datos. Los programas de aplicación y la actividad en terminales no deberán ser afectados por cambios de cualquier tipo que preserven la información y que teóricamente permitan la no afectación en las tablas base.
- Independencia de la integridad. Las restricciones de integridad de una base de datos deberán poder definirse en el mismo sublenguaje de datos relacional y deberán almacenarse en el catálogo, no en los programas de aplicación.
- Independencia de la distribución. Un DBMS relacional tiene independencia de distribución.
- Regla de la no subversión. Si un sistema relacional tiene un lenguaje de bajo nivel (un sólo registro cada vez), ese bajo nivel no puede ser utilizado para suprimir las reglas de integridad y las restricciones expresadas en el lenguaje relacional de nivel superior (múltiples registros a la vez).

1.1.9.2 Bases de Datos Relacionales



Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas.

Estas bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo.

Entre las características más importantes de las bases de datos relacionales están las siguientes:

- Cada “Tabla” contiene un solo tipo de registros.
- Los campos no tienen un orden específico.
- Los registros no tienen un orden específico.
- Cada campo tiene un solo valor.

La estructura básica de toda Base de Datos Relacional son las “tablas”, en las que cada tabla está integrada por campos. Para que la Base de Datos sea relacional, debe de existir por lo menos una relación de una tabla con otra tabla.

Cuando se empieza a modelar una Base de Datos, las tablas cambian de nombre, por así decirlo, ahora son llamadas ENTIDADES, y los campos que llenas esas Entidades (Tablas) ahora son llamados ATRIBUTOS. Cada Entidad debe tener Atributos los cuales pueden o no tener registros. Cada Entidad debe tener un identificador para cada registro.

Comúnmente en el primer Atributo de una Entidad se coloca la llave primaria (Primary Key), la cual es un identificador único de un registro. Esta llave primaria debe cumplir con tres características, las cuales son:

- NOT NULL (No Nula, es decir, que no puede tener valor nulo).
- NOT CHANGE (No Cambiar, es decir, no se puede cambiar el valor que fue asignado en su origen).
- NOT DUPLICATE (No Duplicada, es decir, no puede ser igual a ningún otro atributo de la entidad).

Para comprender claramente lo anterior se muestra la figura 1.7 en la cual se observa cómo está conformada una entidad.

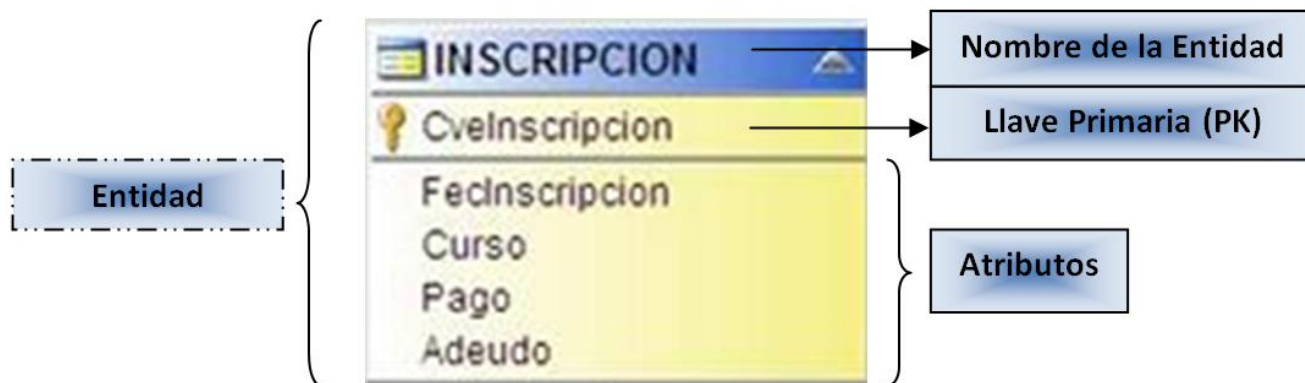


Figura 1.7 Descripción de una entidad

En una base de datos existen diferentes tablas, las cuales tienen distintos tipos de relación unas con otras, en general existen tres tipos de relación entre tablas, las cuales son:

- Uno a Uno (1:1)
- Uno a Muchos (1:M) ó Muchos a Uno (M:1)
- Muchos a Muchos (M:M)

Para comprender estos tres tipos, veamos un pequeño ejemplo, el cual se ilustra en la figura 1.8:

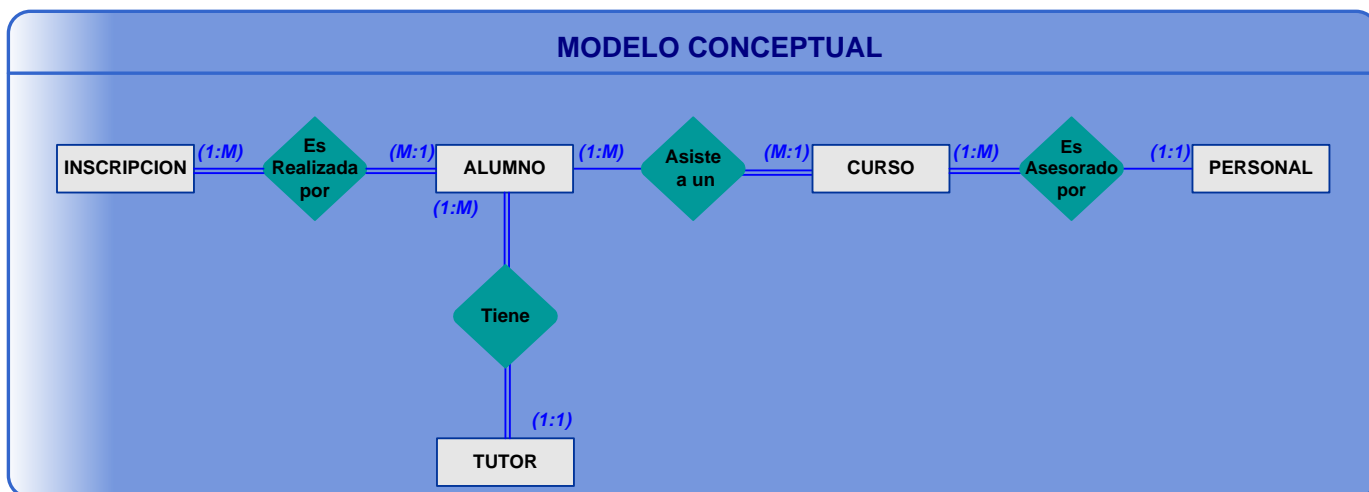


Figura 1.8 : Ejemplo de relación entre tablas.

Explicando el ejemplo anterior tenemos que:

Un Alumno asiste a muchos cursos; a lo cual la relación es 1:M, y a un Curso asisten muchos alumnos.



Un Alumno tiene un Tutor, en este caso la relación es 1:1, y un Tutor tiene muchos alumnos.

Una Inscripción es realizada por muchos alumnos, ahora la relación es 1:M y un Alumno realiza más de una inscripción.

Un curso es asesorado por un personal, la relación es 1:1, y un Tutor puede impartir más de un curso, ahora la relación es 1:M.

1.1.9.3 Lenguaje de Consulta Relacional

El SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. El nombre "SQL" es una abreviatura de Structured Query Language (Lenguaje de consultas estructurado). Como su propio nombre indica, SQL es un lenguaje informático que se puede utilizar para interactuar con una base de datos y más concretamente con un tipo específico llamado base de datos relacional.

Las bases de datos relacionales están basadas en el concepto matemático de relaciones entre conjuntos. Así las operaciones que se pueden efectuar entre relaciones son tanto las comunes a los conjuntos, unión, intersección, diferencia, producto cartesiano; como las específicas de las relaciones, selección, proyección, etcétera.

Si q , r y s son relaciones con todos los dominios iguales, esto es, con el mismo esquema, se les puede aplicar las operaciones típicas de conjuntos; Unión, Intersección, Diferencia, Producto Cartesiano, Proyección, Selección.

1.1.10 Independencia de Datos y la Integridad Referencial

1.1.10.1 Independencia de datos

Una de las principales ventajas que provee una base de datos es la independencia entre los datos y los tratamientos que se hacen de ellos ya que en los sistemas orientados a procesos los datos eran sumamente dependientes de los programas.

Como tal, la independencia de los datos se refiere a la protección contra los programas de aplicación que puedan originar modificaciones cuando se altera la organización física o lógica de la base de datos

Existen dos niveles de independencia de datos:



- Independencia física de datos
- Independencia lógica de datos

La independencia física de los datos es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación. Dicho de otra forma, el esquema conceptual no es afectado por cambios al esquema físico de datos. Es decir se refiere al ocultamiento de los detalles sobre las estructuras de almacenamiento a las aplicaciones de usuario, o sea la descripción física de datos puede cambiar sin afectar a las aplicaciones de usuario. Si el RDBMS modifica su organización interna de ficheros, no pasa nada con el esquema conceptual.

La independencia lógica de los datos es la capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación. Algunos elementos del esquema externo no son afectados por cambios al esquema conceptual.

1.1.10.2 Integridad referencial

El término de integridad referencial se enmarca en la segunda regla de integridad y se aplica a las claves foráneas:

“Si en una relación hay alguna clave foránea, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos”.

Lo que en realidad trata de decir el texto anterior es que las claves foráneas no pueden dejar de tener correspondencia con la clave primaria de la tabla externa.

- **Clave primaria:** Es aquel atributo que identifica de manera única a un registro. Esto es, no debe haber dos tuplas que tengan el mismo valor, por lo tanto, con sólo conocer el valor de la clave primaria para una determinada tupla será suficiente para identificarlo de manera única.
- **Clave candidata:** Es el atributo o conjunto de atributos que podrían servir como llaves primarias. Una llave candidata debe cumplir dos condiciones:
 - **Unicidad:** no pueden existir dos tuplas con el mismo valor en todos los atributos que forman la llave candidata.
 - **Minimidad:** no existe ningún subconjunto de la llave que cumpla la regla de unicidad.
- **Clave secundaria:** Son aquellas claves candidatas que no se eligieron como llave primaria, es decir, tienen todas las características para ser claves primarias, pero que por alguna razón no fueron tomadas como tal debido quizás a que hubo otra que cumplía mejor con ese objetivo.



- **Clave foránea:** Es una clave primaria en otra relación, estas representan las asociaciones entre las diferentes entidades, es decir, son claves que están siendo compartidas por dos tablas para formar una relación entre ellas.

1.1.11 Modelo Entidad - Relación

El modelo Entidad - Relación, es una técnica de diseño de bases de datos gráfica, que incorpora información relativa a los datos y la relación existente entre ellos, para poder así plasmar una visión del mundo real sobre un soporte informático.

Entre sus principales características tenemos:

- Sólo refleja la existencia de los datos sin expresar lo que se hace con ellos.
- La independencia de la base de datos y de los sistemas operativos.
- La inclusión de todos los datos sin considerar las aplicaciones que se tendrán.

1.1.12 Normalización

El proceso de cristalización de las entidades y sus relaciones en formatos de tabla usando los conceptos relacionales se llama proceso de normalización y consiste en agrupar a los campos de datos en un conjunto de relaciones o tablas que representan a las entidades, sus características y sus relaciones de forma adecuada.

La razón de la normalización es asegurar que el modelo conceptual de la base de datos funcionará. Esto no significa que una estructura no normalizada no funcionará, sino que puede causar algunos problemas cuando los programadores de aplicación traten de modificar la base de datos para insertar, actualizar o eliminar datos.

Las *formas normales* son una serie de restricciones que se definen sobre las estructuras relacionales para evitar anomalías al efectuar adiciones, eliminaciones o actualizaciones de tuplas. En la figura 1.9 se puede apreciar los niveles de la normalización.

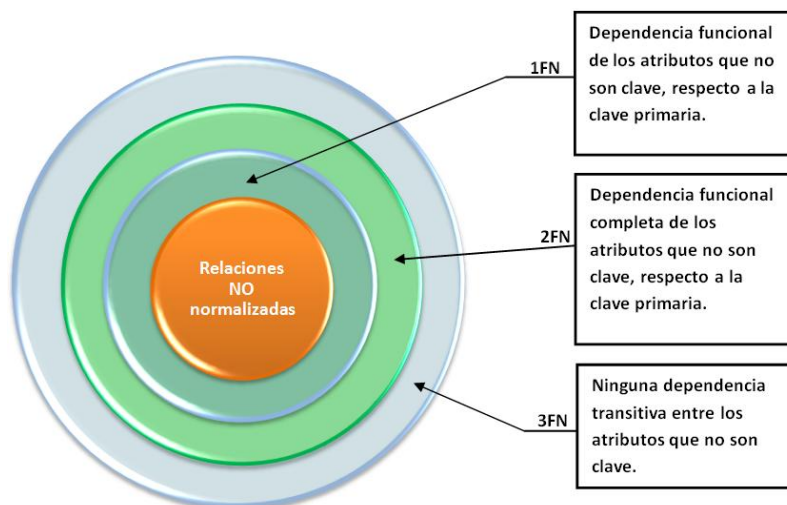


Figura 1.9 Niveles de normalización.

Las formas normales se pueden listar como sigue:

- **Primera Forma Normal (1FN):** Una relación está en primera forma normal si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Si se ve la relación gráficamente como una tabla, estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna.
- **Segunda Forma Normal (2FN):** Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria.
- **Tercera Forma Normal (3FN):** Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. La dependencia es transitiva si existen las dependencias siendo atributos o conjuntos de atributos de una misma relación.

Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.
- Están encaminadas a eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas.



1.2. SISTEMAS MANEJADORES DE BASES DE DATOS RELACIONALES (RDBMS)

1.2.1 RDBMS y su función

Entre la base de datos física (es decir, los datos tal y como están almacenados en la realidad) y los usuarios del sistema, existe un nivel de programas denominado manejador de bases de datos (MBD) o, en la mayoría de los casos, el sistema administrador de bases de datos DBMS (Data Base Management System).

Un RDBMS es el conjunto de programas que permiten la definición, manipulación y control de acceso para una o varias bases de datos.

Algunas características de los RDBMS son:

- Facilitan la integridad, seguridad y acceso de los datos.
- Los datos se almacenan como mínima redundancia.
- Las aplicaciones son independientes del almacenamiento físico de los datos.

Un DBMS debe permitir las siguientes condiciones en una base de datos:

- Los datos han de estar almacenados juntos.
- Tanto los usuarios finales como los programas de aplicación no necesitan conocer los detalles de las estructuras de almacenamiento.
- Los datos son compartidos por diferentes usuarios y programas de aplicación; existe un mecanismo común para la inserción, actualización, borrado y consulta de los datos.
- Los procedimientos de actualización y recuperación, comunes, y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de datos.
- Tanto datos como procedimientos pueden ser transportables conceptualmente a través de diferentes RDBMS.

Lo que sucede en un RDBMS cuando un usuario realiza una petición, es:

- El usuario realiza una petición a la base de datos empleando algún sublenguaje de datos, SQL en la mayoría de los casos (Acceso).



- El RDBMS interpreta esa solicitud y la analiza (Sintaxis).
- El RDBMS inspecciona en orden el esquema externo de ese usuario, la correspondencia externa/conceptual asociada, el esquema conceptual, la correspondencia conceptual/interna y la definición de la estructura de almacenamiento. (privilegios, objetos y la definición de los objetos).
- El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenada y devuelve una respuesta al usuario.

Gráficamente, se aprecia en la figura 1.10.

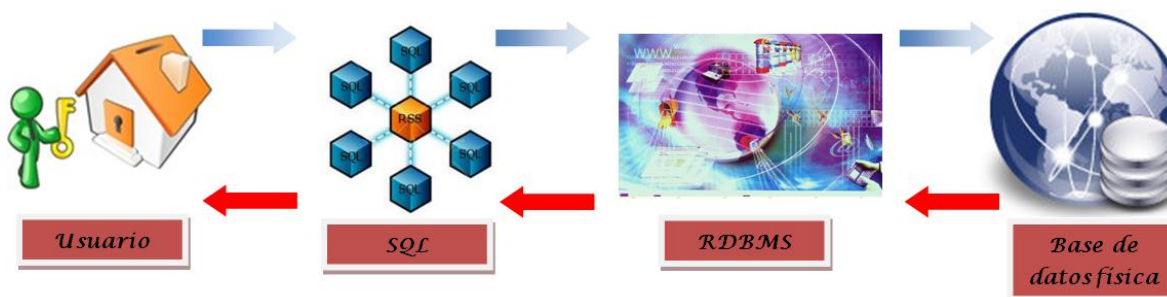


Figura 1.10: Secuencia de una petición dentro de un RDBMS

1.2.2 Componentes de un RDBMS

Un RDBMS se compone principalmente de cuatro partes, las cuales son:

- **DDL o Lenguaje de Definición de Datos:** Se utiliza para crear, eliminar o modificar tablas, índices, vistas, triggers, procedimientos; es decir, nos permite definir la estructura de la base de datos mediante comandos como crear (*Create*), eliminar (*Drop*), o alterar (*Alter*).
- **DML o Lenguaje de Manipulación de Datos:** Se utiliza para realizar la consulta y edición de la información contenida en la base de datos, esto implica: seleccionar, insertar, borrar, modificar. Los DML se distinguen por sus sublenguajes de recuperación subyacentes; se pueden distinguir dos tipos de DML, el procedural y el no procedural. La principal diferencia entre ambos, es que los lenguajes procedurales tratan a los registros individualmente, mientras que los no procedurales operan a un conjunto de registros.
- **DCL o Lenguaje de Control de Datos:** Se utiliza para la definición de los privilegios de control de acceso y edición a los elementos que componen la base de datos (seguridad), es decir, permitir o revocar el acceso. Los permisos a nivel base de datos pueden otorgarse a usuarios para ejecutar



ciertos comandos dentro de la base o para que puedan manipular objetos y los datos que puedan contener estos.

- **DD o Diccionario de Datos:** El contenido del diccionario puede considerarse como “datos acerca de los datos” (los cuales comúnmente reciben el nombre de metadatos), es decir, definiciones de otros objetos de la base de datos.

1.2.3 SQL ANSI 89, 92, 99 Y 2003

En 1989, tanto ANSI como ISO, publicaron estándares que definían al modelo relacional en el manejo de Bases de Datos. Estos estándares son: ANSI X3.135-1989 e ISO/IEC9075:1989.

Las características principales del ANSI SQL 89 son:

- Agregan la capacidad conocida como integridad referencial y la descripción de todo el modelo relacional.
- Se definió que el lenguaje SQL está compuesto por comando, cláusulas, operadores y funciones de agregado. Estos elementos se combinan para definir y manipular la base de datos.
- Se establecen los elementos de un RDBMS (DDL, DML y DCL), así como las instrucciones y sintaxis relacionadas con cada uno de ellos.
- Establecimiento de las cláusulas del comando *select*, (from, where, group by, having, order By).
- Definición de los operadores lógicos: AND, OR y NOT.
- Definición de los operadores de comparación (=, !=, >, <).
- Se determinan las funciones de agregado, tales como: AVG, COUNT, SUM, MAX, MIN.

Las características principales del ANSI SQL 92 son:

- Toma todas características definidas en el estándar ANSI SQL 89.
- Permite la definición de esquemas.
- Permite la definición de dominios por parte de los usuarios, es decir, tipos de datos definidos por el usuario.
- Menciona las consideraciones para realizar consultas sencillas, multi-tablas y subconsultas.



- Incluye los operadores EXISTS y NOT EXISTS.
- SELECT columns FROM table WHERE EXISTS (subquery);
- SELECT *
- FROM suppliers
- WHERE not exists (select * from orders Where suppliers.supplier_id = orders.supplier_id);
- Contempla el uso de la palabra DISTINCT en una consulta.
- Menciona algunas consideraciones para el uso de las cláusulas GROUP BY y HAVING.
- Especifica la definición de vistas en una base de datos.

Las características principales del ANSI SQL 99 son:

- Toma todas características definidas en los estándares ANSI SQL 89 y 92.
- Incluye nuevos tipos de datos escalares: BOOLEAN, CLOB (objeto de caracteres largo) y BLOB (objeto binario grande).
- Presenta dos nuevos operadores de totales: EVER y ANY.
- Incorpora generadores de tipo de dato: REF, ARRAY y ROW.
- Soporta una opción LIKE en CREATE TABLE, lo cual permite que todas o algunas definiciones de columna de una nueva tabla sean copiadas a partir de otra ya existente.
- Incluye la cláusula WITH para introducir nombres abreviados para determinadas expresiones.
- Incorpora una nueva expresión de condición IS DISTINCT para la cláusula FROM.

En cuanto a ANSI SQL 2003, se puede decir que este estándar hace pequeñas modificaciones a todas las partes de SQL 1999, y oficialmente introduce algunas nuevas características tales como:

- Características relacionadas con XML.
- Ventana de funciones.
- Generador de la secuencia, lo que permite secuencias estandarizadas.
- Dos nuevos tipos de columna: valores auto-generados (autogenerated values) y la identidad de columnas (identity-columns)
- La nueva sentencia MERGE.
- Eliminación de los mal implementados tipos de datos "BIT" y "BIT VARYING".



1.2.4 Principales Sistemas Manejadores de Bases de Datos Relacionales

Entre los principales sistemas manejadores de bases de datos relacionales tenemos a Microsoft SQL Server, Sybase, Oracle, Informix y DB2, Postgre SQL y a MySQL.

1.2.4.1 Microsoft SQL Server

Las características principales de este RDBMS son:

- Compatibilidad con estándares de W3C, incluyendo XML, Xpath, XSL, HTTP.
- Obtiene código XML de las consultas realizadas con SQL.
- Manipulación de documentos XML.
- Manejo de bases de datos distribuidas.
- Manejo de varias particiones físicas para almacenamientos de datos flexibles.
- Permite realizar algunas tareas de mantenimiento y administración de la base de datos sin tener que darla de baja.
- Permite realizar acciones OLAP (Online Analyzing Processing), herramienta que permiten analizar datos almacenados en una Base de Datos, por medio de cubos de información.
- Consulta y modificación de cubos virtuales de manera gráfica.
- Conectividad con clientes ODBC y JDBC.

En la tabla 1.2 se muestran los sistemas operativos, hardware, memoria y disco duro, donde se puede instalar Microsoft SQL Server

Sistema Operativo	Windows NT Server Windows 2000 Server Windows 2000 Advanced Server Windows 2000 Data Center Windows Server 2003 Windows XP Windows Vista
Hardware	P III o superior, según los requerimientos del sistema operativo.
Memoria	Enterprise Edition: 64 MB como mínimo. Estándar Edition: 32 MB como mínimo.
Disco Duro	SQL Server 2000, instalación completa 180 MB SQL Server 2000, instalación típica 170 MB SQL Server 2000, instalación mínima 65 MB SQL Server 2000, solo herramientas clave 90 MB
Observaciones	Es necesario disponer de Microsoft Windows 2000 Server para algunas características de SQL Server 2000. Nota: los requerimientos de hardware y software cambian dependiendo de la versión que se desee instalar.

Tabla 1.2: Software y Hardware donde se puede instalar Microsoft SQL Server.



1.2.4.2 SYBASE

Las características principales de SYBASE son:

- Diseñado para soportar aplicaciones OLTP (On Line transaction Procesor, ambiente diseñado para insertar, actualizar y borrar datos en una base datos).
- Permite integrar aplicaciones basadas en XML con la base de datos y crear reglas de negocio que ejecuten Java Beans.
- Conectividad con clientes ODBC y JDBC.
- Soporte para BLOB's (Large Objects).
- Permite realizar query's XQL, lo cual significa que utiliza un motor abierto para búsqueda dentro de contenidos XML almacenados en la base de datos, o en un URL.
- Tamaño expandido de filas y datos, es decir soporta filas más grandes, columnas más grandes. Se soportan ahora tamaños de páginas de 2k, 4k, 8k o 16k (entre mas tamaño de página mayor rendimiento de operaciones SQL).
- Compresión de copias de respaldo.
- Bloqueo a nivel de fila, nivel de página de datos, nivel de página (datos e índices), nivel de tabla.
- Cifrado lógico de la información.

En la tabla 1.3 se muestran los sistemas operativos, memoria, disco duro y las especificaciones del servidor, para instalar Sybase.



	Sistemas Operativo		
	Empresa	32 Bits	64 Bits
Plataformas Soportadas	Sun Microsystems Hewlett Packard IBM SGI Compaq Linux Microsoft Windows	HP - UX AIX IRIX Red Hat NT, 98 (solo clientes)	Solaris HP - UX AIX TRU 64
Memoria	Sistema Operativo		Memoria
	AXP HP 32 HP 64 Linux NT Sun 32 Sun 64		94 MB 64 MB 90 MB 32 MB 46 MB 66 MB 92 MB
Disco Duro	240 MB		
Especificaciones del Servidor	Bases de datos totales 32,767 Tamaño de la base de datos 4 TB Bases de datos en update 16 Tablas en una consulta 50 Usuarios por base de datos 2,146,484,223 Columnas por tabla 1024 Tamaños de página 2k, 4k, 8k, 16k Argumentos por procedimiento 1024		

Tabla 1.3: Software y Hardware donde se puede instalar Sybase.

1.2.4.3 ORACLE

Sus características Principales son:

- Ofrece varias plataformas de desarrollo para Internet y aplicaciones tradicionales, tales como: XML, Enterprise Java Engine, SQL y PL/SQL, C, C++, entre otras.
- Soporte Unicode.
- Extiende las habilidades de una base de datos para Internet.
- Amplía distintos mecanismos para protección de datos.
- Soporta OLTP y OLAP.
- Contiene mecanismos de gran funcionalidad y flexibilidad para compartir la información almacenada en la base de datos con otras bases de datos o aplicaciones.
- Conectividad con clientes ODBC y JDBC.
- Soporte para BLOB's.



- Ofrece escalabilidad y performance sin modificar las aplicaciones instaladas.
- Soporta columnas con cifrado de datos.
- Permite replicación de bases de datos (bases de datos distribuidas).
- Ofrece distintas herramientas para la administración de la base de datos.
- Redefinición de tablas en línea.
- Respaldo y recuperación en línea.

Plataformas:

- Solaris
- HP-UX
- Compaq Tru64
- AIX
- HP Alpha
- Linux
- Windows NT/2000/XP Professional

1.2.4.4 INFORMIX y DB2

Las características principales de estos manejadores de bases de datos relacionales son:

- Soporta Bases de datos de más de cuatro Terabytes (TB).
- Soporte para acceso a la base de datos vía web.
- Provee acceso a cualquier tipo de cliente.
- Permite manejo de base de datos distribuidas.
- Capacidad de replicación de bases de datos.
- Permite realizar query's en paralelo.
- Contiene plataformas de desarrollo con SPL: (Informix Stored Procedure Language), C, Java, XML.
- Conectividad vía ODBC, JDBC, OLE/DB.
- Soporta aplicaciones para Commerce, e inteligencia de negocios.
- Soporta OLAP y OLTP.

Plataformas que soporta:

- IBM AIX.
- SGI IRIX.



- Sun Solaris.
- HP-UX.
- Compaq Tru64.
- Linux.
- Windows NT/2000/XP/2003.

1.2.4.5 Postgre SQL

Postgre SQL tiene las siguientes características:

- Base de datos de distribución libre.
- Velocidad.
- Confiabilidad.
- Flexibilidad.
- Bajo costo de operación.
- Conformación a estándares ANSI.
- Estrategia de almacenamiento MVCC para grandes volúmenes.
- Soporta replicación de bases de datos.
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, XML.
- Soporta SSL nativo.

Postgre SQL corre bajo cualquier plataforma UNIX.

1.2.4.6 MySQL

Sus características principales son:

- Soporta los estándares ANSI.
- Contiene esquemas de almacenamiento independiente que se pueden seleccionar de acuerdo a las necesidades.
- InnoDB para transacciones y bloqueo de registros.
- MyISAM sin transacciones
- Soporte para SSL.



- Query's con manejo de cache que puede incrementar el performance de la base de datos en un 200%.
- Permite manejo de replicación de bases de datos.
- Soporta indexado de texto.

Plataformas que soporta:

- Linux
- Windows.
- FreeBSD
- Sun Solaris.
- IBM-AIX.
- Mac OS X.
- HP-UX.

1.2.4.7 Comparación entre RDBMS's

A continuación se muestra la tabla 1.4 en la cual se hace una comparación, de los datos que manejan los distintos manejadores, ya que cuando se desea migrar una base de datos de un manejador a otro nos es muy útil conocer los tipos de datos que son compatibles en cada manejador.

PostgreSQL	MySQL	Oracle	SQL Server	Informix	DB2
bigint	bigint	binary_integer	bigint		bigint
numeric	double, decimal, float	number, float binary_float, binary_double	numeric, decimal, float, real	decimal, smallfloat, float,	decimal, float, real, numeric
integer	int, mediumint		int	integer	
smallint	smallint, tinyint		smallint, tinyint	smallint	Tinyint, smallint
character	char	char, nchar	char, nchar	char	char, nchar
character varying	varchar, enum, set	varchar, varchar2,	varchar, nvarchar	Varchar,	
text	longtext, mediumtext, text, tinytext		text, ntext	text	ntext
bytea	tinyblob, blob, mediunblob, longblob	lob (blog, clob, nclob, bfile), blob, clob	binary, varbinary	byte	Binary, varbinary
timestamp	datetime, timestamp	timestamp	timestam, rowversion	datetime c1 to c2	timestamp
date	time,date, year	date	date, datetimeoffset, datetime,datetime2	date	date, datetime, smalldatetime, time

Tabla 1.4 Comparación entre distintos RDBMS's



1.2.5 Consideraciones de Hardware

La elección del Hardware no es algo que debe considerarse a la ligera y no sólo involucra lo poderoso que pueda ser, ni que tanto deseamos abarcar en nuestro sitio, sino también el presupuesto destinado para él.

Las partes importantes a considerar son:

- **Procesador.** La velocidad del procesador afecta de manera directa la rapidez de procesamiento de la computadora, pudiéndose tomar esta como referencia para evaluar su desempeño, por ejemplo, influye en la compilación de programas. De igual forma, utilizar el servidor de consola implica desviar memoria RAM de otros procesos.
- **Tarjeta Madre.** Otra parte esencial de una PC es la tarjeta madre, es la parte más complicada de actualizar por que se encuentra muy ligada al procesador. Hay que tener en cuenta que existen fabricantes que utilizan tecnología propietaria en sus equipos y actualizar el equipo significa cambiar: gabinete, tarjeta madre, procesador y memoria.
- **Tipos de Canal (BUS).** PCI, Interconexión de Componentes Periféricos. Está trabaja a 64 bits, un ancho de banda de 133 MBps y se usa en equipos 80486, Pentium, Pentium MMX, Pentium Pro hasta Pentium III. AGP Puerto Acelerado de Gráficos (AGP, Accelerated Graphics Port). Una especificación de interfaz nueva desarrollada por Intel Corporación. AGP está basado en PCI, pero es diseñado sobre todo para las exigencias de rendimiento de gráficos 3-D, ya que el controlador de gráficos puede tener acceso directamente a la memoria principal. El canal AGP es de 32 bits de ancho y corre a 66 MHZ. Esto se traduce en un ancho de banda total de 266 MBps, a diferencia del ancho de banda PCI de 133 MBps. AGP también aporta dos modos opcionales más rápidos, con rendimientos de 533 MBps y 1.07 GBps. Además, permite que las texturas 3-D sean almacenadas en la memoria principal en lugar de hacerlo en la memoria de vídeo.
- **Memoria.** La memoria RAM se encuentra también ligada al diseño de la tarjeta Madre, es importante la velocidad de acceso en nanosegundos (ns), que es el tiempo que le lleva al CPU tener acceso a la RAM. Entre más nanosegundos tarde más lento es el acceso.
- **Disco Duro.** Para un sitio con uno o más usuarios, el disco duro no representa una consideración primaria en el momento del desempeño. Sin embargo, cuando se incrementa tanto el número de usuarios como la demanda de servicios, el tamaño y tipo del disco duro debe de ser tomado en cuenta.



1.2.6 Tipos de Usuarios en una Base de Datos

En una base de datos interactúan distintos tipos de usuarios, como los son:

- El programador de aplicaciones, quien se encarga de escribir los programas de aplicación que utilizan la base de datos.
- El usuario final, el cual tiene acceso a los datos de la base a través de alguna aplicación desarrollada o utilizando una interfaz incluida como parte integral de los programas del DBMS.
- El DBA (database administrator, Administrador de la base de datos).

Los roles o funciones son también conocidos como perfiles, los cuales sirven como medio para conceder privilegios sobre todo el sistema a un usuario que los requiera. Estos permisos pueden verse reflejados sobre objetos o sobre el mismo sistema. En los RDBMS ya vienen predeterminados algunos, sin embargo no en todos se pueden crear nuevos roles; estos son:

- DBA (database administrator, administrador de la base de datos).
- DBO (database owner, Dueño de la base de datos).
- SSO (Security System Officer, Oficial del sistema de seguridad).
- Oper (Operador)

1.3. SQL (Structures Query Language)

1.3.1 Definición y Componentes de SQL

SQL (Structured Query Language; Lenguaje Estructurado de Consulta) es un lenguaje de consulta para bases de datos, siendo adoptado como estándar de la industria en 1986. Desde entonces se han realizado revisiones al estándar para incorporar nueva funcionalidad conforme la industria de las bases de datos lo va requiriendo. Una de las revisiones más importantes fue la de 1992, conocida como ANSI SQL92.

La ventaja de la adopción del ANSI SQL, es que los diversos RDBMS (Relational DataBase Management System; Sistema Manejador de Bases de Datos Relacional) tienen que acoplarse al estándar, permitiendo así una mayor compatibilidad entre ellos. Esto implica que conociendo una variante del SQL, se tienen los conocimientos necesarios para poder utilizar otros RDBMS: MS SQL Server, Oracle, Sybase, Interbase, MySQL, PostgreSQL, DB2, etcétera.



Aunque los distintos fabricantes tratan de acoplarse al estándar ANSI SQL, es cierto que cada uno implementa funcionalidades extra que le dan un valor agregado a su producto pero sacrificando un poco la compatibilidad, por lo cual se podrán notar ciertas diferencias entre distintos RDBMS.

La estructura de SQL utiliza palabras en inglés, lo que lo hace fácil de aprender y utilizar; las instrucciones se enfocan a qué buscar, dejando al RDBMS la tarea de cómo recuperar la información.

Los componentes del SQL son:

- DDL (Data Definition Language; Lenguaje de Definición de Datos). Permite crear, modificar y eliminar estructuras de datos como: tablas, bases de datos, índices, etcétera.
- DML (Data Manipulation Language; Lenguaje de Manipulación de Datos). Permite consultar, insertar, modificar y eliminar datos de las tablas.
- DCL (Data Control Language; Lenguaje de Control de Datos). Permite establecer los privilegios de acceso a los datos, en otras palabras, establece la seguridad de la base de datos.

1.3.2 Definición de Datos

Antes de comenzar a trabajar con SQL, es necesario conocer los elementos que intervienen en la definición de la información en una base de datos, para poder manipularla de manera adecuada.

La tabla es el elemento fundamental de una base de datos relacional, la cual consiste de una serie de renglones (registros) que representan la información. Cada renglón está dividido en columnas (campos) los cuales deben de tener un tipo de dato establecido.

1.3.2.1 Definición de Datos del Sistema

Cada columna dentro de una tabla debe tener asociado un tipo de dato, siendo la labor del diseñador de la base de datos, el de encontrar el mejor tipo de dato que satisfaga las necesidades de almacenamiento y recuperación de cierta información.

Los tipos de datos que se manejan en una base, pueden variar ligeramente entre diferentes RDBMS; sin embargo el estándar ANSI, asegura que cierto tipo de datos estará presente en cualquier RDBMS asegurando así la compatibilidad. Algunos RDBMS implementan sinónimos para los tipos de datos, de manera que puedan cumplir con el ANSI SQL92; no obstante, internamente son convertidos a un tipo de



dato que sí esté soportado. Por ejemplo MS SQL Server acepta el tipo de dato DOUBLE PRECISION pero lo convierte y maneja como un FLOAT.

En la Tabla 1.5 se muestra una comparación de los diferentes tipos de datos que manejan los RDBMS.

Tipo en SQL92/SQL3	MySQL	PostgreSQL	Oracle	Sybase	Ms SQL Server	Descripción
Tinyint	tinyint			tinyint	tinyint	
int, integer	int, integer	int4	int (lo convierte a number)	Int	int	Entero con signo de 4 bytes
smallint	smallint	int2	smallint (lo convierte a number)	smallint	smallint	Entero con signo de 2 bytes
numeric(p,d)	numeric(p,d)	numeric(p,d)	numeric(p,d)	numeric(p,s)	decimal(p,s)	Numérico con precisión p y d decimales
float()	float()	float4, float8	float()	float()	float()	Número de punto flotante
real			real (Lo convierte a float)	real	real	Número Real
double			Doublé precisión (lo convierte a float)	Doublé precisión	Double precisión (se convierte en float)	Número Doble
Carácter varying(n)	varchar(n)	varchar(n)	varchar2(n)	varchar(n)	varchar(n)	Carácter de longitud variable
boolean	bit	bool		bit	bit	Valor booleano
date	date	date				Fecha sin hora del día
time	time	time				Hora del día
timestamp	timestamp	timestamp	date	Datetime	datetime	Fecha y hora del día
char, character(n)	char(n)	char(n)	char(n)	char(n)	char(n)	Cadena de caracteres de longitud fija
interval		interval				Tiempo de intervalo
blob	blob		blob	image	image	Binary large object
clob	text	text	clob	text	text	Character large object

Tabla 1.5: Comparación de tipos de datos entre RDBMS's



1.3.2.2 El Valor Nulo

Es importante conocer el concepto de valor nulo, en el contexto de una base de datos, debido a que frecuentemente un valor nulo es confundido con un valor numérico de cero o una cadena vacía. Un valor nulo se representa en SQL con la cláusula NULL y representa la ausencia de información.

Por ejemplo si en un listado de empleados algunos aparecen en el dato de comisión como NULL, esto indica que no se tiene dicha información aunque erróneamente se podría interpretar que estos empleados tienen comisión del 0%. Ahora, suponiendo que los datos que aparecen como NULL corresponden a la edad, indican que el dato no fue proporcionado y por lo tanto se carece de dicha información y claramente se observa que no es lo mismo que una edad de cero.

Hasta este momento se ha hablado únicamente de tipos de datos numéricos, aunque los valores nulos también se utilizan en cualquier otro tipo de dato, por ejemplo en texto: No es lo mismo una cadena vacía que un valor nulo.

En Oracle cuando uno trata de insertar una cadena de texto vacía, automáticamente lo convierte a valor NULL. Esta característica particular, no es compatible con el estándar y Oracle podría en un futuro cambiar este comportamiento.

1.3.2.3 Tablas

Las tablas son el elemento fundamental que compone a una base de datos relacional porque todo gira en torno a ellas. Las tablas son estructuras de almacenamiento que albergan la información en forma de registros (renglones) que deben de ser identificados de manera única y esto se logra a través de una llave primaria.

La tabla es la representación física en la base de datos de una Entidad mientras que las relaciones son representadas mediante restricciones; lo anterior se muestra en la figura 1.11.

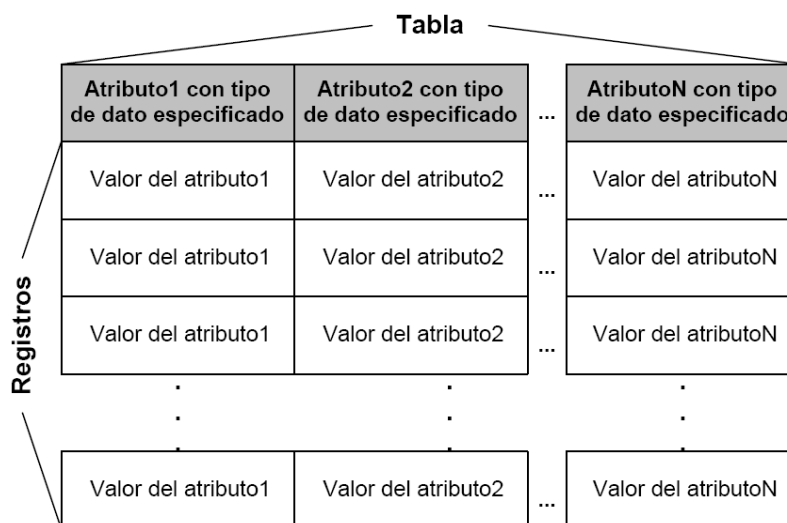


Figura 1.11 : Representación de valores en tablas

1.3.2.4 Modificación de Tablas

Dependiendo del RDBMS esta modificación de estructura de la tabla ofrece mayor o menor flexibilidad. Por ejemplo, MySQL es de las más flexibles, porque permite cambiar incluso el tipo de dato de una columna mientras que en Sybase o MS SQL Server es necesario eliminar y volver a crear la tabla con la estructura deseada. En la Tabla 1.6 se observa una comparación de las modificaciones que permiten realizar los diferentes RDBMS's.

Características soportadas:	MySQL	PostgreSQL	Oracle	Sybase	MS SQL Server
Agregar campo	Si	Si	Si	Si	Si
Eliminar campo	Si	No	Si	No	No
Renombrar campo	Si	Si	No	No	No
Cambiar el tipo de dato del campo	Si	No	Si	No	No
Definir llave primaria	Si	Si	Si	Si	Si
Definir llave foránea	Si	Si	Si	Si	Si
Renombrar tabla	Si	Si	Si	No (pero brinda un procedimiento almacenado)	No (pero brinda un procedimiento almacenado)

Tabla 1.6 : Comparación de modificaciones en los distintos RDBMS.

La sintaxis varía de un RDBMS, debido a las diferentes características soportadas, sin embargo todos ellos utilizan la cláusula ALTER TABLE para realizar las modificaciones posibles, a una tabla.



1.3.2.5 Tipos de Llaves e Índices

Un índice es una estructura de almacenamiento físico que permiten recuperar datos de una manera muy eficiente. En un esquema relacional, cada registro dentro de una tabla debe ser identificado de manera única, y esto se logra a través de una llave primaria, a la cual se le genera de manera automática un índice, que ayuda a ser más eficiente el proceso de consulta de la información. También existen las llaves foráneas, que son las columnas que hacen referencia a la llave primaria de otra tabla. A través de ellas se establecen relaciones entre tablas.

Es un error frecuente confundir entre índices y llaves, quizá sea el hecho de que al crear una llave primaria se genera un índice, lo cual no sucede para una llave foránea. Hay que tener presente que los índices tienen como función acelerar el proceso de recuperación de la información.

- **Llaves Primarias (Primary Key).** Una llave primaria permite identificar de manera única un registro dentro de una tabla. Al crear una llave de este tipo se genera automáticamente un índice de valores únicos, por lo que los valores de los campos que involucra la llave primaria no se pueden repetir ni ser nulos.
- **Llaves Foráneas (Foreign Key).** Las llaves foráneas son atributos de una tabla que hacen referencia a la llave primaria de otra tabla. Estas llaves foráneas permiten establecer relaciones entre las distintas tablas que existen dentro de la base de datos. La mayoría de los RDBMS implementan restricciones (constraints) cuando se genera una llave foránea, de este modo asegura la integridad de la información almacenada en la base de datos. MySQL no implementa estas restricciones, por lo que se puede llegar a dar una falta de integridad en la información, sin embargo esto no es del todo malo porque las restricciones se pueden implementar con lógica de aplicación en caso de que exista y porque la falta de implementación de estas restricciones, permite que MySQL trabaje más rápido que cualquier otro RDBMS. Dependiendo del tipo de información y de aplicación a desarrollar se puede sacrificar la característica a cambio de rapidez en respuesta. Las llaves foráneas se pueden crear dentro de la definición de la tabla, o una vez que esta ya existe, se puede utilizar la cláusula ALTER TABLE para agregar esta restricción. A diferencia de las llaves primarias, las llaves foráneas no generan un índice, por lo que de ser necesario se deberá crear con la cláusula CREATE INDEX.
- **Índices.** Un índice es una estructura de almacenamiento físico que ocupa un espacio. Los índices ayudan al Servidor SQL a localizar datos y son transparentes para el usuario. El propósito principal de un índice es proporcionar un acceso más rápido a los datos, aunque en algunos casos su propósito es asegurar que el contenido de un campo sea único. Abusar del empleo de índices puede llevar a que se degrade el tiempo de respuesta del servidor en lugar de mejorarlo, esto se debe a que en operaciones que involucran inserción, modificación o eliminación de datos, los índices deben de ser actualizados lo cual puede consumir un tiempo considerable. Por lo general



se suele crear un índice únicamente cuando una columna es empleada frecuentemente en una cláusula WHERE y la tabla tiene un tamaño considerable.

1.3.3 Manipulación de Datos

La mayor parte del trabajo con SQL girará en torno a cuatro comandos:

- Select. Permite seleccionar (recuperar) información de una tabla.
- Insert. Permite agregar información a una tabla.
- Delete. Permite eliminar información de una tabla.
- Update. Permite actualizar información que existe en una tabla.

SQL se basa en el álgebra relacional, por ello es importante conocer las operaciones del álgebra relacional y su relación con SQL. En la figura 1.12 se pueden observar las operaciones del álgebra relacional.

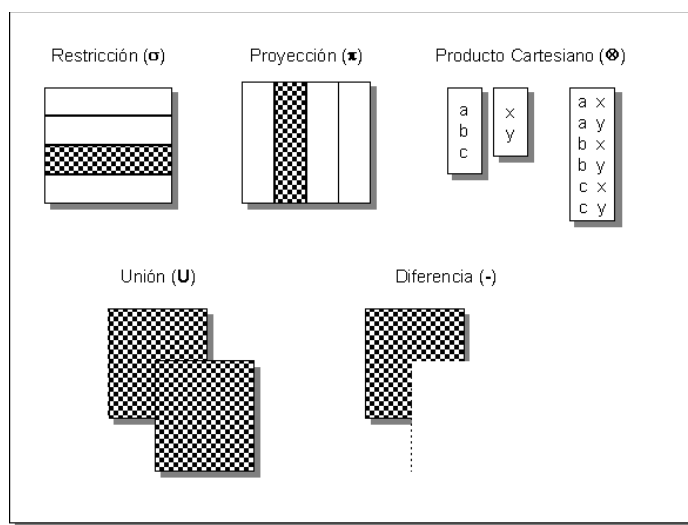


Figura 1.12 : Operaciones del álgebra relacional.

1.3.3.1 Selección de Datos

Las tablas dentro de una base de datos son las estructuras que tienen almacenada la información en forma de registros. Para poder recuperar esa información almacenada, se requiere del comando SELECT de SQL.



El comando SELECT es sumamente útil, ya que a través de él es posible realizar desde una consulta simple que sólo involucra una tabla, hasta una consulta compleja donde intervienen dos o más tablas, varias condiciones, agrupaciones de datos y ordenamientos.

```
SELECT { * | [DISTINCT] <campo>, <campo> ... }  
      FROM <nombre_tabla>, [ <nombre_tabla> ]  
      [WHERE <condición> ]  
      [GROUP BY <campo>, <campo>,... ]  
      [HAVING <condición> ]  
      [ORDER BY <campo>, <campo>,.... ]
```

Estructura para realizar Consultas dentro del RDBMS

Lo que se puede apreciar en la estructura de la instrucción SELECT, es que nunca debe faltar ni la palabra SELECT, ni FROM. Todos los demás elementos son opcionales.

1.3.3.2 Inserción de Datos

A través de la instrucción INSERT de SQL, se introduce la información a una tabla. La estructura general de este comando es la siguiente:

```
INSERT INTO <tabla> [( <nombreCampo1>, <nombreCampo2>, <nombreCampo3> ... )]  
{ VALUES ( <valorCampo1>, <valorCampo2>, <valorCampo3> ... ) | <Expresión select> }
```

1.3.4 Estructuras de Control de Flujo

Este tipo de estructuras se emplea en el desarrollo de programas, que en un RDBMS puede ser un procedimiento almacenado. Los RDBMS comerciales son los que se han desarrollado más en el aspecto de programación en la base de datos, sin embargo las instrucciones aunque similares, difieren entre uno y otro.

IF-THEN-END IF

La sintaxis en Sybase, MS SQL Server, Oracle es:

```
IF <expresión>  
<Bloque A de instrucciones>  
  ELSE  
  <Bloque B de instrucciones>  
END IF
```



Si la expresión evaluada es verdadera se ejecuta el bloque A de instrucciones y de lo contrario se ejecuta el bloque B.

IF-THEN-ELSE-END IF

La sintaxis en Sybase y MS SQL Server es:

```
IF <expresión 1>  
<Bloque A de instrucciones>  
    ELSE IF <expresión 2>  
        <Bloque B de instrucciones>  
    ELSE IF <expresión 3>  
        <Bloque C de instrucciones>  
    ELSE  
        <Bloque D de instrucciones>  
END IF
```

La sintaxis en Oracle únicamente difiere en la cláusula elseif, la cual se escribe como elsif.

Si la expresión 1 evaluada es verdadera se ejecuta el bloque A de instrucciones, de lo contrario se evalúa la expresión 2 y si es verdadera se ejecuta el bloque B de instrucciones, de lo contrario se evalúa la siguiente expresión y así sucesivamente.

1.4. Fundamentos de Sistemas Operativos

1.4.1 Sistema Operativo y sus componentes

Una computadora sin software es un montón de fierros que no nos es de gran ayuda, necesita de programas para solucionar nuestros problemas. Estos programas pueden clasificarse en dos grandes grupos:

- Programas de sistema (software de sistema): controlan la operación de la computadora. Los más representativos son los compiladores y el sistema operativo.
- Programas de aplicación (software de aplicación): resuelven problemas para los usuarios. Algunos ejemplos pueden ser: procesadores de palabras, hojas de cálculo, manejadores de bases de datos, juegos, etcétera.



Como ya se mencionó, entre los programas del sistema, el sistema operativo es el fundamental; controla todos los recursos de la computadora y proporciona la base sobre la que pueden escribirse los programas de aplicación. Es un nivel de software por encima del hardware que controla todas las partes del sistema y presenta al usuario una interfaz que es más fácil de usar que el hardware subyacente.

En un sistema de cómputo se reconocen los siguientes niveles, analizando estos niveles desde el usuario hacia la máquina física, lo anterior se muestra en la figura 1.13.

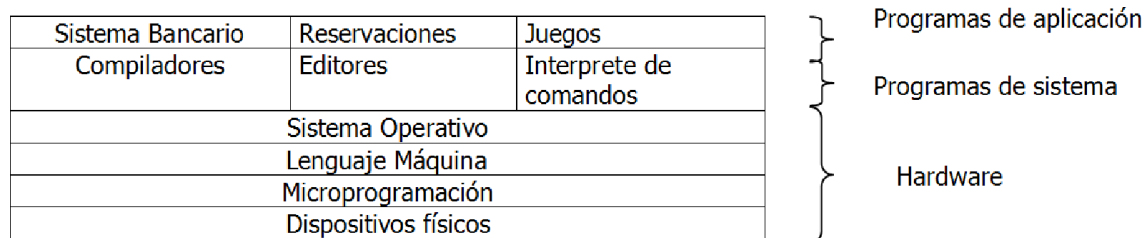


Figura 1.13: Componentes de un sistema de cómputo.

Un Sistema Operativo es el encargado de presentar al usuario el equivalente de una máquina virtual o extendida que sea más fácil de programar que el hardware subyacente. Es también el encargado de proporcionar una asignación ordenada y controlada de los recursos para los varios programas que compiten por ellos.

Un sistema operativo está formado por varios programas que en conjunto presentan al usuario una vista integrada del sistema, estos son:

- Administrador de procesos (scheduler).
- Administrador de E/S.
- Administrador de la Memoria.
- Manejo del Sistema de Archivos.

Todo el software ejecutable de la computadora se organiza en varios procesos. Un proceso es tan solo un programa en ejecución, consta del programa ejecutable, sus datos y pila. Los estados en los que un proceso se puede encontrar son:

- En ejecución
- Listo (no existe CPU disponible para él)
- Bloqueo

Las llamadas al sistema son: Mecanismos de comunicación entre el interprete de comandos y el sistema operativo.

Un **archivo** es una secuencia de bits.

En la figura 1.14, se puede observar las diversas partes en las que se compone un sistema operativo, y como éste interactúa con el hardware.

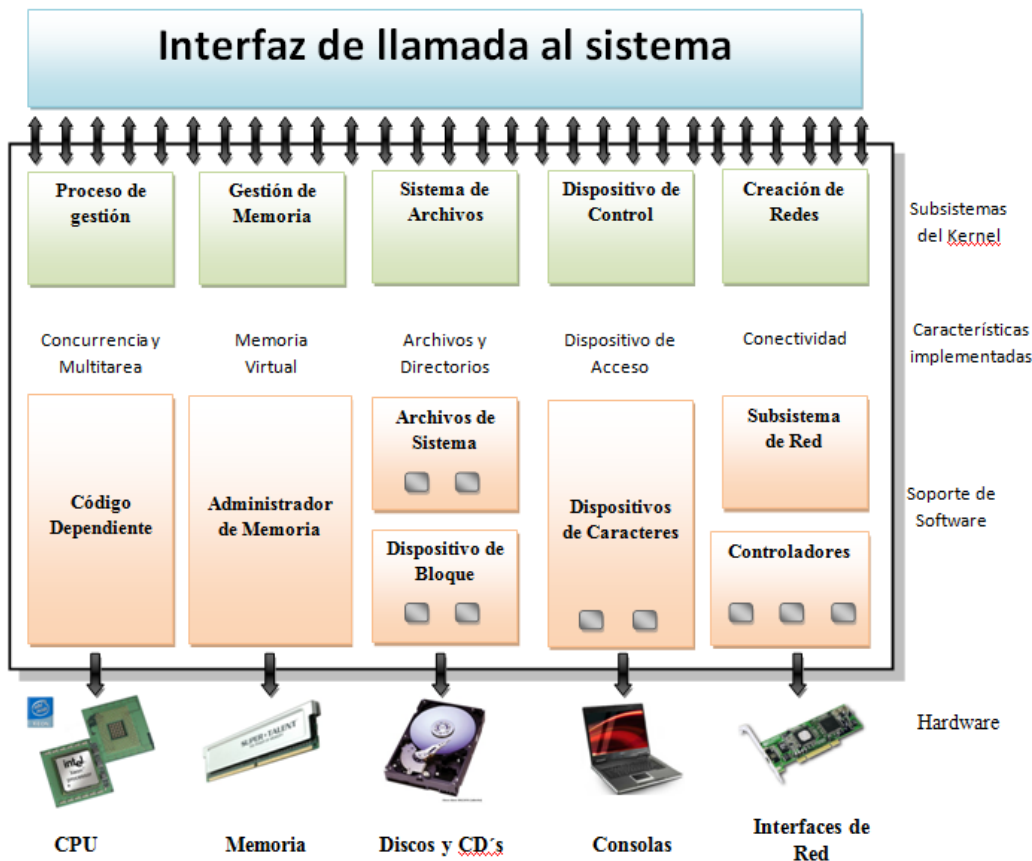


Figura 1.14: Partes de un sistema operativo y su interacción con el hardware.

1.4.2 Software Libre (open source)

El "software libre" es una cuestión de libertad, no de precio. Para entender el concepto, deberá pensar en "libre" como en "libre expresión", no como en "barra libre". Un programa es software libre si los usuarios tienen la libertad de:

- Ejecutar el programa, para cualquier propósito (libertad 0).
- Estudiar cómo trabaja el programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una condición necesaria.



- Redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- Mejorar el programa y publicar sus mejoras, y versiones modificadas en general, para que se beneficie toda la comunidad (libertad 3). El acceso al código fuente es una condición necesaria.

1.4.3 UNIX

Unix es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy. UNIX es una marca registrada de The Open Group en Estados Unidos y otros países. Esta marca sólo se puede aplicar a los sistemas operativos que cumplen la "Single Unix Specification" de esta organización y han pagado las regalías establecidas.

1.4.3.1 Familias de UNIX

En la práctica, el término UNIX se utiliza en su acepción de familia. Se aplica también a sistemas multiusuario basados en POSIX (tales como: GNU/Linux, Mac OS X, FreeBSD, NetBSD, OpenBSD.)

Los cuales no buscan la certificación UNIX por resultar cara para productos destinados al consumidor final o que se distribuyen libremente en Internet. En estos casos, el término se suele escribir como UN*X, *NIX, ó *N?X.

En la Tabla 1.7 se observan los UNIX que existen en la actualidad.

Producto	Fabricante	Características
Solaris	Sun Microsystems	Basado en AT&T, con muchas extensiones. Arquitectura Sparc y X86
HP-UX	Hewlett- Packard	Híbrido AT&T y BSD, con particularidades propias. Arquitectura propietaria
Linux	Público	BSD (SunOS) en lo interno, AT&T en la administración. Arquitectura Intel x86, sparc , alpha, y otras. Múltiples distribuciones; Red-Hat, S.u.s.e., Slackware, Debian, Mandriva. FreeBSD Público Basado en BSD.

Tabla 1.7 : Diferentes UNIX que existen

1.4.4 GNU/LINUX

GNU/Linux es un clon del sistema operativo Unix, escrito desde cero por el finlandés Linus Torvalds con la asistencia de un pequeño grupo de hackers esparcidos por la red.



Estrictamente, Linux se refiere al núcleo o kernel. En un sentido más amplio, comprende el núcleo del sistema operativo más un conjunto de programas que permiten: compilar lenguajes de programación, editar texto, interpretar comandos, manejar archivos y discos, acceder a otras máquinas, establecer comunicaciones, enviar y recibir correo electrónico, manejar las colas de impresión y un sinnúmero de tareas más.

Las características de GNU/Linux son:

- **Portable:** el mismo sistema operativo corre en un espectro de máquinas que van desde notebooks a supercomputadoras. Es el único sistema operativo con estas características.
- **Exible:** se adapta a muchas aplicaciones diferentes.
- **Potente:** dispone de muchos comandos y servicios ya incorporados.
- **Multiusuario:** atiende a muchas personas simultáneamente.
- **Multitarea:** hace muchas cosas a la vez.
- **Elegante:** sus comandos son breves, coherentes, específicos para cada tarea y muy eficientes.
- **Orientado a redes desde el comienzo.**
- **Dispone de un estándar (POSIX)** que debe cumplir todo sistema operativo que pretenda ser Unix, lo que asegura una evolución predecible y compatibilidad con otros Unix.

Existen diferentes versiones de Linux, como se puede ver en la Tabla 1.8.

Versión	Año	Usuarios Estimados	Tamaño del Kernel (KBytes)
0.01	1991	100	63
0.99	1992	1000	431
0.99	1993	20,000	938
1.0	1994	100,000	1,017
1.2	1995	500,000	1,850
2.0	1996	1,500,000	4,718
2.2	1999	7,500,000	10,593
2.4	2001	10,000,000	19,789
2.6	2003	20 – 50,000,000	32,476

Tabla 1.8: Versiones de Linux

GNU/Linux fue diseñado teniendo en mente los siguientes objetivos:

- Crear un sistema interactivo de tiempo compartido diseñado por programadores y para programadores, destinado a usuarios calificados.
- Que fuera sencillo, elegante, escueto y consistente.



- Que permitiera resolver problemas complejos combinando un número reducido de comandos básicos.

1.4.5 Windows SERVER

Windows Server 2008 es el nombre del sistema operativo para servidores de Microsoft. Es el sucesor de Windows Server 2003. Al igual que Windows Vista, Windows Server 2008 se basa en el núcleo Windows NT 6.0.

Hay algunas diferencias (algunas sutiles y otras no tanto) con respecto a la arquitectura del nuevo Windows Server 2008, que pueden cambiar dramáticamente la manera en que se usa este sistema operativo. Estos cambios afectan a la manera en que se gestiona el sistema hasta el punto en que se puede llegar a controlar el hardware de forma más efectiva, se puede controlar mucho mejor de forma remota y cambiar de forma radical la política de seguridad. Entre las mejoras que se incluyen, están:

- Nuevo proceso de reparación de sistemas NTFS: proceso en segundo plano que repara los archivos dañados.
- Creación de sesiones de usuario en paralelo: reduce tiempos de espera en los Terminal Services y en la creación de sesiones de usuario a gran escala.
- Cierre limpio de Servicios.
- Sistema de archivos SMB2: de 30 a 40 veces más rápido el acceso a los servidores multimedia.
- Dirección de carga aleatoria: protección contra malware en la carga de controladores en memoria.
- Error de Arquitectura de Hardware en Windows: protocolo mejorado y estandarizado de reporte de errores.
- Virtualización de Windows Server: mejoras en el rendimiento de la virtualización.
- PowerShell: inclusión de una consola mejorada con soporte GUI para administración.
- Server Core: el núcleo del sistema se ha renovado con muchas y nuevas mejoras.



1.4.6 Diferencias de Servidores Windows y Servidores Linux

La principal diferencia entre Windows y Linux son los lenguajes de programación que soportan. Mientras Linux incluye PHP y bases de datos MySQL (entre otras opciones), Windows incluye ASP, ASP.NET y bases de datos Access y SQL Server (entre otras opciones). Por lo tanto dependerá de las necesidades de programación de su página web la elección de un pack de alojamiento web Linux o Windows. En la Tabla 1.9 se observan las diferencias entre Windows y Linux.

CARACTERISTICAS	Unix / Linux	Windows
MS FrontPage	✓	✓
Flash	✓	✓
Shockwave	✓	✓
RealAudio/Video	✓	✓
Cgi-script	✓	✓
Perl	✓	✓
PHP	✓	✓
SSH (Telnet Seguro)	✓	
MySQL	✓	✓
ASP (active server pages)		✓
MS Access		✓
ODBC		✓
SQL Server		✓
Visual Basic Scripts		✓
Windows Media		✓

Tabla 1.9 : Diferencias entre Windows y Linux

1.5. Habilidades Directivas para Administradores

Necesitamos desarrollar la habilidad para comunicarnos con las personas con las que colaboramos, descubrir sus capacidades, comprender las necesidades del grupo y obtener lo mejor de cada persona en el desempeño grupal.

Para establecer lazos con las personas que integran nuestro grupo de trabajo es necesario:

- Hacer coincidir idiomas verbales: Visual, auditivo, táctil, olfativo y gustativo.
- Hacer coincidir lenguaje corporal: Movimiento de los ojos, postura corporal, saludo, tipo de respiración, gestos, ademanes.
- Coincidencia auditiva: El tono de voz, el volumen, el ritmo, énfasis en las palabras, la velocidad en el hablar.
- Espejear sentimientos: Amor, entusiasmo, angustia, alegría, tristeza.
- Igualar pensamiento: Puntos de vista, visión del mundo, valores y creencias, foco de atención, objetivos, estrategias.



1.5.1 Estilos de Comunicación

En general todas las personas tenemos distintos tipos de comunicación, algunas personas desarrollan más un estilo que otro. Lo esencial para tener una total comunicación con el grupo de trabajo es conocer muy bien estos estilos de comunicación. Existen tres tipos de comunicación: Visual, Auditivo y Cinestésico.

- **Visual:** En este estilo de comunicación las personas piensan con imágenes. Representan las ideas, los recuerdos y la imaginación mediante imágenes mentales.
“Si lo veo lo recuerdo... Si lo veo y lo escucho lo asimilo... Y si lo veo, lo escucho y lo práctico, lo aprendo para siempre”.
- **Auditivo:** Las personas piensan con sonidos, los cuales pueden ser voces o ruidos.
- **Cinestésico:** En este estilo las personas representan los pensamientos como sentimientos internos o sensaciones físicas. El sabor y el olor se incluyen en esta categoría sensorial.

Para transmitir bien el mensaje que se quiere hacer llegar al equipo de trabajo un buen comunicador utiliza de forma natural el estilo que mejor encaja con las preferencias de su interlocutor. Así tiene la certeza de hablar un mismo idioma y de que su interlocutor le entiende.

1.5.2 Liderazgo

Liderazgo es el proceso de infundir, guiar o dirigir a los miembros del grupo hacia el éxito en la consecuencia de metas y objetivos organizacionales.

El líder debe de trabajar con un enfoque de desarrollo, inspirar confianza a su gente; más que dar instrucciones y corregir, debe ser capaz de inspirar una visión común, modelar con el ejemplo, reconocer y recompensar los esfuerzos. Centrarse en las personas, las personas no son cosas. El líder debe responder a las necesidades que cada individuo tiene como ser humano.

Desarrollar un equipo, derribar barreras entre funciones y personas, trabajar de manera interdisciplinaria, aumentar la participación y ser capaz de utilizar y fomentar los mejores recursos del grupo. Esto mejorará la calidad y la productividad.

Generar un clima motivador, los procesos de calidad y el buen servicio se basan en la participación de las personas y estimular su iniciativa para que actúen con responsabilidad sin necesidad de vigilancia.

En la Tabla 1.10 se aprecian las diferencias entre ser Líder y ser Jefe (Administrador):



JEFE / SUPERVISOR / GERENTE	LÍDER / COACH / FACILITADOR
El jefe cree que su trabajo es empujar o conducir a las personas.	El líder cree que su función es elevar a las personas y darles soporte.
El jefe cree que debe dirigirse a las personas en monólogos y conferencias.	El líder cree en la generación de dialogo con las personas preguntando y escuchando.
El jefe cree que debe controlar a los otros a través de las decisiones que toma.	El líder cree que debe facilitar a los otros para que tomen decisiones y las implementen.
El jefe cree que conoce las respuestas.	El líder cree que debe buscar las respuestas.
El jefe dispara inseguridad administrando una sana dosis de temor como modo afectivo de lograr complicidad.	El líder cree que debe usar los objetivos y los valores para inspirar compromiso y estimular la creatividad.
El jefe cree que su trabajo es evidenciar los errores.	El líder cree que su trabajo es celebrar el aprendizaje.
El jefe cree que debe resolver problemas y tomar decisiones.	El líder cree que debe ayudar a otros para que tomen decisiones y resuelvan problemas.
El jefe cree que debe delegar responsabilidad.	El líder cree que debe modelar la responsabilidad.
El jefe cree que debe crear estructuras y procedimientos para que las personas cumplan.	El líder cree que debe generar una visión y promover la flexibilidad a través de valores y lineamientos para el comportamiento.
El jefe cree que debe hacer correctamente las cosas.	El líder cree que debe hacer las cosas correctas.
El jefe cree que el poder radica en su conocimiento.	El líder cree que su poder viene ligado a su vulnerabilidad.

Tabla 1.10: Diferencias entre Líder y Jefe

Pasar de Jefe a Líder significa reconocer tres áreas de resultados igualmente importantes:

- **Tareas:** Donde se trata de lograr resultados específicos, en la calidad, con calidad y la oportunidad requeridas. El reto: Asegurar una mayor productividad y calidad.
- **Grupos:** Donde se requiere ocuparse de los individuos a su cargo para asegurar su capacitación, motivación y progreso. El reto: Promover una mayor preparación, involucramiento y responsabilidad.
- **Equipos:** Donde debe asegurarse su integración y desarrollo, así como un clima de comunicación y colaboración. El reto: Es conseguir una mayor participación y trabajo en equipo.

1.5.2.1 Tipos de Liderazgo

Los tipos de liderazgo se pueden listar como sigue:

- **Administración de club campestre:** El interés por las necesidades de los empleados para lograr relaciones satisfactorias, conduce a un ambiente y ritmo de comodidad y sociabilidad muy agradable en el trabajo.



- **Estilo de mando caracterizado por trabajo en equipo:** La realización del trabajo se debe a la dedicación plena de los empleados. La interdependencia basada en un interés común de la empresa, conduce a relaciones de respeto y confianza.
- **Estilo de mando basado en el hombre-organización:** Es posible obtener un nivel adecuado de producción estableciendo un equilibrio entre la necesidad de obtener unidades y la de mantener la moral del personal en un nivel satisfactorio.
- **Estilo de mando empobrecido:** Su preocupación por las necesidades de la organización o las personas que colaboran en ellas es mínima. No se involucra. Cumple con los requisitos de su trabajo y se involucra lo menos posible con la gente. Hace el mínimo necesario para acumular antigüedad sin ninguna consideración real por hacer una contribución. La necesidad de conservar el trabajo personal es lo que conduce a la motivación negativa.
- **Estilo de mando autoridad – obediencia:** Condiciones de trabajo organizadas de forma que la producción se consigue a través de una mínima interferencia del elemento humano.

1.5.3 Presentaciones Efectivas

Para realizar presentaciones efectivas hay que tomar en cuenta lo siguiente:

- **Cantidad de Información.** Las diapositivas son solamente una ayuda audio visual, por lo que deben contener sólo información clave que debe ser ampliada durante la presentación oral. No existe una regla específica sobre la cantidad de información adecuada para cada diapositiva, pero por lo general se recomienda que no incluya más de 12 líneas.
- **Uso del Color.** Diseñar diapositivas agradables, con buena combinación entre los colores del fondo y el texto de la misma. Para una lectura más fácil, debe existir un contraste entre el texto y el fondo. Se recomienda utilizar colores oscuros en el fondo y claros para el texto o viceversa. Evita los colores muy brillantes o muy vivos en combinación ya que podrían ser molestos para la vista y poco legibles.
- **Uso de Multimedia.** En una presentación es posible incluir videos, sonido, transiciones de diapositivas, imágenes fijas y con movimiento, objetos voladores y gráficos.
 - No satures tu presentación con sonidos u objetos voladores, es mejor utilizar pocos para reforzar puntos importantes.





- Procura que cada imagen, animación, sonido, gráfica o video que utilices apoye y refuerce tu mensaje.
- Utiliza graficas sencillas, claras y fáciles de entender.

- **Tamaño y Tipo de letra**
 - **Tamaño:** Por lo general para presentaciones con una audiencia mediana es recomendable utilizar tamaños no menores a 24 puntos para que las personas no tengan dificultad tratando de leer sus diapositivas.

 - **Tipo de letra:** Para no recargar visualmente la presentación trate de utilizar solamente un tipo de letra, si lo considera muy necesario utilice como máximo dos tipos. Es mejor usar tipos de letra sencillos y fáciles de leer en vez de las *fuentes muy elaboradas o que simulan la letra cursiva*.

- **Uso de Viñetas.** El uso de viñetas resulta muy útil para presentar la información de una forma breve y clara. Estas van acompañadas de frases o palabras claves en vez de oraciones o párrafos completos para evitar leer o recitar las diapositivas a la audiencia.

- **Presentación Oral.** Se ha señalado que la opinión que tenemos de los demás suele basarse en tres características principales:
 - Contenido verbal = 7 %
 - Interés del discurso oral = 38 %
 - Lenguaje corporal = 55 %

1.5.4 Reuniones Efectivas

Primero que nada es administrar el tiempo destinado a la presentación, de la siguiente manera:

- Asumir el control de los requerimientos que te hacen en tu tiempo disponible.

- Asegurar que el uso que haces de tu tiempo – una distribución limitada – se ajuste de la mejor manera a tus metas y necesidades personales.

- Establecer tus metas personales. Si no están claras, no tienes un marco de referencia para poder distribuir el tiempo.



Los pasos a seguir para organizar una reunión de trabajo efectiva son:

1. Fijar objetivo de la reunión para que puedas prepararla y llevarla a cabo con mayor rapidez y acierto.
2. Hacer una lista detallada de asuntos por tratar para lograr reuniones más cortas y productivas.
3. Elaborar y distribuir con antelación la orden del día y la documentación, permite a los asistentes prepararse en forma adecuada.
4. Dirigir la reunión en sí, reducir su duración y duplicar su eficacia.

1.6. ADMINISTRACIÓN DE BASES DE DATOS

1.6.1 Tareas del administrador

Las principales responsabilidades y tareas del administrador de bases de datos (DBA) son:

- La correcta elección e instalación del Sistema Manejador de Bases de Datos Relacional (RDBMS).
- Creación y Definición del tamaño de la Base de Datos.
- Definición de los procedimientos de respaldo y de recuperación de la Base de Datos.
- Definición de los esquemas de seguridad y de integridad que tendrá la Base de Datos.
- La correcta administración de los usuarios de la Base de Datos.
- Supervisar el desempeño de las aplicaciones que corren bajo la base de datos.

La seguridad se refiere a la protección que se deberá tener contra accesos de usuarios no autorizados a la información contenida dentro de la base de datos. El objeto de datos que puede requerir protección, va desde la base de datos completa, de sólo algunas tablas, o tan sólo de una celda en específico dentro de una tabla. El alcance de la protección se conoce como granularidad. Diferentes usuarios pueden tener diferentes derechos sobre los mismos objetos. Los manejadores de bases de datos relacionales permiten que el administrador pueda restringir el acceso a ciertos datos que no competen con las funciones del usuario.

La seguridad se logra utilizando básicamente dos mecanismos:

- **Vista:** Permiten que se limite la visión del usuario a ciertas columnas o filas de determinadas tablas.
- **El sistema de privilegios:** En el cual el administrador puede conceder o revocar privilegios sobre los objetos de la base de datos a los distintos usuarios.



La definición de estos derechos, no es competencia del administrador, es más parte de las políticas de la empresa, él es un ejecutor y auditor de las mismas; los RDBMS tienen los mecanismos que le permiten al administrador implementar estas seguridades, usando principalmente órdenes SQL.

La integridad se refiere a los controles que se deben de diseñar para que las transacciones que realicen los usuarios cumplan con las condiciones preestablecidas por el administrador, y así se eviten posibles inconsistencias o errores de digitación para impedir que la base de datos tome estados inválidos.

La Integridad implica asegurarse que la información contenida en la base de datos está correcta; se debe verificar que las transacciones que realicen los usuarios autorizados para ello, cumplan con las condiciones que mantengan a la base de datos en un estado válido.

El administrador de la base de datos (DBA), cómo recién se expresó, debe supervisar el desempeño de las aplicaciones sobre la base de datos para que el usuario final tenga la información en forma oportuna. Hay muchos factores que pueden incidir en una baja en el desempeño, por ejemplo:

- Una organización física inadecuada de la base de datos.
- El crecimiento acelerado de las tablas que componen la base de datos.
- Un sistema de concurrencia mal diseñado.
- Cambios en el hardware o en el software.

Por lo anterior, el administrador de la base de datos debe estar monitoreando constantemente la base para hacerle los ajustes necesarios.

1.6.2 Concurrencia

Las bases de datos relacionales poseen un sistema de manejo de concurrencia, que permite que múltiples usuarios compartan los mismos recursos al mismo tiempo; los recursos son objetos de la base de datos.

Los objetos de un sistema de concurrencia son:

- Que el acceso a los objetos en forma simultánea, por los usuarios, sea coordinado.
- Consistencia de los datos: Un usuario siempre debe tener una vista consistente de los datos que está manipulando.



Cuando varias transacciones son ejecutadas correctamente en una base de datos compartida, sus ejecuciones deben ser sincronizadas. En otras palabras, el efecto de una transacción debe ser el mismo que sería si no existiera ninguna otra transacción que se estuviera ejecutando concurrentemente. El efecto de ejecutar diferentes transacciones al mismo tiempo, debe ser el mismo que si fueran ejecutadas en serie con algún orden; si esto se puede lograr, se dice que las transacciones son serializables.

1.6.3 Respaldo de Información

En todo sistema manejador de bases de datos existe la posibilidad de que ocurran fallas que generen pérdida de información, estas fallas pueden ocurrir por:

- Errores del usuario
- Actualización indebida de una tabla.
- Fallas en el Hardware: como por ejemplo un aterrizaje de algún disco duro
- Fallas en el Software: como por ejemplo, errores en el código de una aplicación.

Todos los manejadores de bases de datos ofrecen mecanismos de respaldo, que permiten hacer copias totales, parciales o incrementales de la base de datos. Los respaldos pueden ser de dos formas:

- En Línea: Mientras se está respaldando, los datos siguen estando disponibles para los usuarios; esto es muy útil para las bases de datos que deben estar en servicio las 24 horas del día.
- Fuera de Línea: Requiere que la base de datos esté fuera de servicio, mientras se está respaldando.

1.6.4 Recuperación

El sistema de recuperación que ofrecen los manejadores de bases de datos, permite, que en el caso de que llegara a ocurrir alguna falla, la base de datos pueda ser restaurada, con un mínimo impacto para el usuario; es decir, que se puedan recuperar todas las transacciones que se habían hecho, hasta momentos antes de que la falla haya ocurrido. Dicha recuperación puede ser:

- Estática; es decir, que la base de datos se restaura hasta el momento en que se encontraba cuando se tomó la última copia.

El comando **dump tran** <nombre de la base> **to** <ruta de respaldo>; este comando hace un respaldo sólo del log de transacciones, y realiza la limpieza del mismo después de la copia.



- Dinámica; en está, no sólo se restaura la base con la copia más reciente que se tenga, sino que también es capaz de recuperar las transacciones que se hayan realizado desde ese entonces. El comando **dump database** <nombre de la base> **to** <ruta de respaldo>; este comando respaldará tanto la base de datos así como también el log de transacciones.

La restauración de un respaldo se realiza en caso de que el sistema sufra alguna falla y se dañe la base de datos, siguiendo el siguiente procedimiento:

- Se borra la base dañada.
- Se crea de nuevo la base con el comando **CREATE DATABASE**.
- Se recupera el respaldo con el comando **load dump** <nombre de la base> **from** <ruta de respaldo>.
- Se pone en línea la base con **ONLINE DATABASE** <nombre de la base>.

1.6.5 Creación e instalación de la Base de Datos

Antes de crear la Base de Datos, el administrador debe decidir el tamaño que tendrá la base de datos, la ubicación y el espacio necesario, si un dispositivo de log es necesario, y si así fuera deberá determinar el tamaño de éste.

Para calcular el tamaño de la base de datos, se debe calcular el tamaño de cada tabla y al final hacer la suma de todas ellas. Se debe de tener en cuenta que el tamaño se debe de expresar en páginas; 512 páginas equivalen a 1024 Kb, o lo que es lo mismo 1 MB. Para realizar el cálculo, debemos de tener:

- Un registro completo y saber el tamaño de éste, expresado en bytes.
- Realizar una estimación de cuantos registros habrá en 5 o 6 años, como mínimo.

Una vez que tengamos los datos anteriores, se deberá realizar una multiplicación de éstos y al resultado se le deberá aumentar un 25% para los índices que pueda tener cada tabla. Al resultado aumentarle de nuevo un 25% de espacio para los controladores, parches u otros software de actualización que se le pudiera instalar a la base de datos.

Para el caso en el que se esté determinando el tamaño de una base de datos con alta concurrencia y transaccional, se le deberá de aumentar ahora un 35% más.

Se deberán redondear las cifras y convertir el tamaño a Megabytes (MB), una vez que se tenga esté valor se multiplica por 512, para por fin obtener el valor en páginas.



Todo lo anterior es el procedimiento que debe de conocer el administrador de la base de datos, sin embargo, existe una forma rápida de hacerlo, existe un procedimiento almacenado, el cual es:

sp_estspace <nombre de la tabla>,<número de registros estimados>

Con este procedimiento almacenado, se deberá obtener el mismo valor calculado por el administrador.

1.6.6 Propietarios y Usuarios de una Base de Datos

Un propietario de una base de datos, es el creador de la base de datos, o la persona a quien se le haya transferido la propiedad de la base de datos.

Al propietario de la base de datos se le ha otorgado la autorización de que adicione y/o retire usuarios de la base de datos, esto con el comando **sp_adduser**.

También puede otorgar o revocar permisos a usuarios para crear objetos en la base de datos y ejecutar comandos con **grant**.

Un usuario que crea objetos en una Base de Datos es dueño de esos objetos, generalmente es el dueño de la Base de Datos, pero el DBO puede otorgar permisos de crear objetos a otro usuario. A los propietarios de los objetos de la base de datos se les todos los permisos de acceso para sus objetos.

La administración de los usuarios es muy fácil de llevar a cabo, ya que existen roles predeterminados, estos se les asignan a los usuarios según el tipo de rol que se les quiera asignar; tenemos tres tipos de roles, los cuales son:

El rol SA (Administrador del Sistema)

- Para este rol están predeterminados el manejo de los discos.
- El crear dispositivos de la Base de Datos.
- Puede crear, modificar y eliminar BD.
- Modifica, bloquea y desbloquea los logins.
- Modifica la configuración del servidor.
- Puede bajar y levantar el servidor; así como también matar procesos.

El rol SSO (Oficial de Seguridad del Sistema)

- Este rol puede bloquear o desbloquear logins, así como la creación de los mismos.



- Cambia passwords.
- Crea, otorga y revoca roles a los usuarios de la base de datos.
- También maneja el sistema que audita la base de datos.
- El rol Oper (Operador), este rol solo crea y recupera respaldos.

1.7. Buenas prácticas en la función de la administración

Las estructuras organizacionales en las áreas de Tecnologías de la Información (TI) difieren de organización a organización. Son un elemento importante para que todos los empleados tengan un conocimiento claro de la responsabilidad y autoridad de cada uno de sus integrantes. Adicionalmente las descripciones de puestos proporcionan a los integrantes del departamento de Sistemas de Información una clara definición de sus roles y responsabilidades.

De acuerdo al manual de revisión de los Auditores de Sistemas de Información Certificados (CISA) las responsabilidades del administrador de bases de datos son (ISACA):

- Custodia información de la organización.
- Define y Mantiene la estructura de los datos en el sistema corporativo de Base de Datos.
- Debe comprender a la empresa, datos de usuario y las reglas de éstos.
- Definir e iniciar los procedimientos de respaldo y recuperación.
- Es el responsable del diseño real, definición y mantenimiento de las Bases de Datos corporativas.
- Especificar la definición física de los datos y cambiarla para su mejor desempeño.
- Seleccionar e implementar herramientas de optimización de la Base de Datos.
- Probar y evaluar las herramientas de programadores.
- Dar soporte técnico a programadores sobre la estructura de la Base de Datos.
- Implementar controles de definición, accesos, actualizaciones y concurrencia.
- Monitorear el uso, recopilar estadísticas de desempeño y ajustar la Base de Datos.
- Es el responsable de la seguridad y clasificación de la información de los datos compartidos, almacenados en los sistemas de Bases de Datos.

Concretamente, todo lo anterior se resume a que el administrador de la base de datos debe de cuidar la integridad, seguridad, accesos, respaldos y el control de la información.



1.7.1 Procesos

Según COBIT, un proceso o procedimiento; es una descripción de una manera particular de lograr algo, es una forma pre-establecida de hacer las cosas, en sí es una serie de pasos que se siguen en un orden regular definido, garantizando un enfoque consistente y repetitivo hacia las actividades.

Y según la ISO 9001-2000, un proceso; es una actividad u operación que recibe entradas y las convierte en salidas. Casi todas las actividades y operaciones relacionadas con un servicio o producto son procesos.

1.7.2 Estándares

Un estándar es aquello que sirve como prototipo, modelo, norma o patrón de referencia. Es una especificación que reglamenta procesos y productos, reglas o directrices para realizar actividades. Las normas o estándares pasan por un proceso de redacción y aprobación por organismos autorizados reconocidos internacionalmente.

Existen organismos de estandarización en Tecnologías de la Información, los cuales son:

- **ISO:** Organización Internacional para la Estandarización. Es la encargada de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Esta organización nace después de la segunda guerra mundial. La ISO 2000 es el estándar internacional para la administración de servicios de TI. La ISO es en sí misma parte de un todo más grande en donde se alinea con ITIL. En la figura 1.15 se puede apreciar la unión de ISO 2000 con ITIL.



Figura 1.15: Unión de ISO 2000 con ITIL.



La implementación de la ISO 2000 conlleva varios beneficios y ventajas, éstos dependen de organización a organización, pero en términos generales son:

- Alineamiento de los servicios de TI a la estrategia de la organización.
 - Creación de un marco formal para proyectos de mejora de servicios.
 - Provee un tipo de comparación con mejores prácticas.
 - Crea ventajas competitivas por medio de la promoción de servicios consistentes y efectivos en costo.
 - Al requerir propiedad y responsabilidad en todos los niveles, crea una filosofía y cultura progresiva.
 - Soporta el intercambio de proveedores de servicio y colaboradores por la creación de procesos operacionales entre organizaciones.
 - Reduce el riesgo y el costo en términos de recepción de servicios externos.
-
- **IEEE**: Instituto de Electricidad y Electrónica de Ingenieros. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, dedicada entre otras cosas a la estandarización.
 - **ITU**: Unión Internacional de Telecomunicaciones; es la organización más importante de las Naciones Unidas en los que concierne a las tecnologías de la información y la comunicación.
 - **COBIT** (Control de Objetivos para la Información Tecnologías Relacionadas): Busca, desarrolla, publica y promueve un autoritario y actualizado conjunto internacional de objetivos de control de tecnologías de la información, generalmente aceptadas, para el uso diario por parte de gestores de negocios y auditores. COBIT utiliza el enfoque en objetivos y necesidades del negocio.

Los activos de información (aplicaciones, la misma información, infraestructura y las personas) están sometidos a una serie de amenazas del entorno. Dichas amenazas son valoradas por la administración de la organización para establecer salvaguardas o controles. Las debilidades en los controles son conocidas como vulnerabilidades. De estos conceptos se desprenden el análisis y la administración de riesgos.



De acuerdo con las guías para la administración de la seguridad de TI un riesgo es “El potencial de que una amenaza dada explote vulnerabilidades de un activo o grupos de activos para causar pérdidas o daños a tales activos.” El impacto o la severidad relativa del riesgo es proporcional al valor del negocio, de la pérdida o daño y a la frecuencia estimada de la amenaza.

El control es una medida para reducir los riesgos; se materializan en las políticas, procedimientos, prácticas y estructuras organizacionales, para lograr sus objetivos y para que los eventos no deseados sean prevenidos o detectados y corregidos.

Existen tres clases de control, las cuales se describen en la tabla 1.11:

CLASE	FUNCIÓN	EJEMPLOS
Preventivo	Detectar problemas antes de que surjan. Monitorear las operaciones y las entradas. Intentar predecir problemas potenciales antes de que ocurran y se hagan ajustes. Prevenir que ocurran errores, omisiones o actos maliciosos.	Emplea sólo personal certificado. Segregar funciones. Controlar el acceso a instalaciones físicas. Usar documentos bien diseñados. Establecer procedimientos para la autorización de transacciones.
Detectivo	Usar controles que detecten y reporten cuando ocurra un error, omisión o acto malicioso.	Puntos de verificación en los trabajos de producción. Verificación duplicada de cálculos. Funciones de auditoría interna.
Correctivo	Minimizar el impacto de una amenaza. Remediar los problemas descubiertos por controles detectivos. Identificar la causa de los problemas. Corregir errores que surjan de los problemas. Modificar los sistemas de procesamiento para minimizar las ocurrencias del problema en el futuro.	Planes de contingencia. Procedimientos de respaldo.

Tabla 1.11: Clases de Control

Cada uno de los procesos de TI de COBIT tiene un objetivo de control de alto nivel y un número de objetivos de control detallados. Como un todo, representan las características de un proceso bien administrado.

Los controles efectivos reducen el riesgo, aumentan la probabilidad de la entrega de valor y aumentan la eficiencia debido a que habrá menos errores y un enfoque administrativo más consistente.

En la figura 1.16 se muestran los procesos de control de COBIT.



Figura 1.16: Procesos de control de COBIT

- **ITIL** (Biblioteca de Infraestructura de Tecnología de la Información): Es una recopilación de las mejores prácticas tanto del sector público como del sector privado respecto a la gestión de servicios de Tecnologías de Información. Se tienen una amplia gama de libros en los cuales se encuentran documentados todos los procesos referentes a la provisión de servicios de la tecnología de la información hacia las organizaciones. ITIL está dividido en diez procesos, mismos que están divididos en cinco procesos operacionales y cinco procesos tácticos, además de incluirse dentro de los procesos operacionales una función que es la de service desk.

Las ventajas de utilizar ITIL son:

- Reducción del costo de desarrollo de prácticas y procedimientos.
- Aumento de la satisfacción de los clientes.
- Mejora en los flujos de comunicación entre el personal de tecnologías de información y los clientes o usuarios.
- Aumento de la productividad así como también de las capacidades y experiencia de los colaboradores.
- Incremento de la calidad del servicio y apoyo a la operación de la organización.

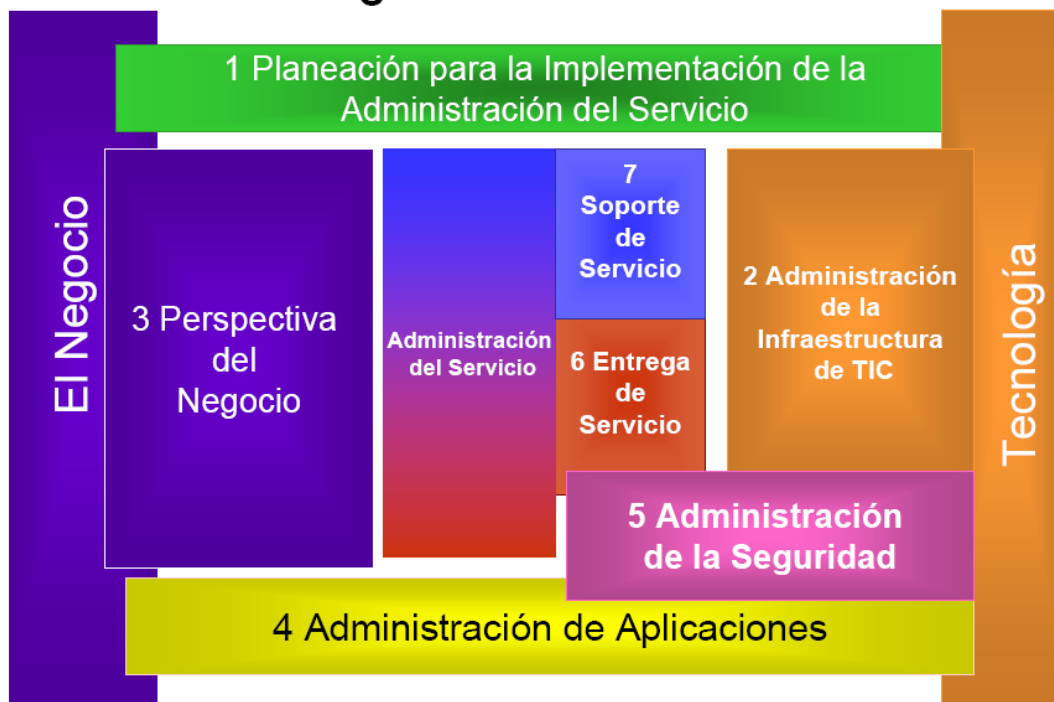
Los principios que lleva ITIL son:

- Alinear los servicios de TI con las necesidades actuales y futuras de la organización.
- Mejora la calidad de la entrega de los servicios de TI.
- Reducir los costos de la provisión de los servicios.
- Crear nuevas oportunidades de expansión para la organización.



- Ser escalable, adaptándose a las particularidades y complejidad de las organizaciones.
- Ser independiente de la tecnología y ser más descriptivo que prescriptivo.

¿Qué cubre ITIL?



Los servicios de TI; los servicios de TI es un conjunto de recursos de la infraestructura de TI, que cumplen con una o más necesidades de sus clientes, están enfocados al cumplimiento de los objetivos del negocio y es percibido por el cliente como una solución que apoya su función en el negocio.

Los elementos que componen la infraestructura de TI son: Hardware, Software, Procedimientos, Documentos y Recursos Humanos.

1.7.3 Auditoría Informática

A finales del siglo XX, los sistemas de TI han constituido las herramientas más poderosas para cualquier organización; apoyan la toma de decisiones, y en los últimos años han generado un alto grado de dependencia, y por ende una elevada inversión en Tecnologías de la Información.

Debido a la importancia que tienen los sistemas de TI en el funcionamiento de una organización, existe la Auditoría Informática. Hoy en día debemos pensar en el auditor como un elemento imprescindible para



una sana operación de las instituciones, su papel ha pasado de ser un detector de problemas, a un agente de cambio, identificador de oportunidades y emisor de propuestas de valor. Su compromiso profesional va más allá de fungir como un mecanismo detectivo.

Las bases de datos deben estar sometidas a controles, ya que las computadoras y los centros de procesamiento de datos son blancos apetecibles para el espionaje, la delincuencia y el terrorismo.

La auditoría informática se define como “Un proceso evolutivo que mediante técnicas y procedimientos aplicados en una organización por personal independiente a la operación de la misma, evalúa la función de tecnología de información y su aportación al cumplimiento de los objetivos institucionales; emite una opinión al respecto y efectúa recomendaciones para mejorar el nivel de apoyo al cumplimiento de dichos objetivos”.

Existen dos tipos de auditorías: las internas y las externas; las primeras son realizadas por la propia organización o en nombre de ésta. Y las externas incluyen las que se denominan generalmente auditorías de segunda y tercera parte. Las auditorías de segunda parte las realizan las partes que tienen interés en la organización, por ejemplo los clientes. Las auditorías de tercera parte las realizan organizaciones auditoras externas e independientes, como por ejemplo las que ofrecen registro o certificación de conformidad.

Las auditorías deben ser planificadas, documentadas, con objetivos bien definidos y con una metodología conocida por todos.

Los principios de la auditoría son:

- **Conducta Ética:** Aquí son esenciales la confianza, integridad, confidencialidad y discreción.
- **Independencia:** El auditor debe de ser objetivo durante todo el proceso de la auditoría, para asegurar que los hallazgos y conclusiones se basarán solamente en evidencias.
- **Evidencia:** La evidencia de la auditoría es verificable. Se basa en muestras de información disponible, ya que una auditoría se realiza durante un periodo de tiempo limitado y con recursos limitados.

1.8. Seguridad en Base de Datos

La seguridad consiste en distinguir las principales vulnerabilidades de seguridad de nuestro entorno de trabajo. En principio es responsabilidad del diseñador de bases de datos, él es quien toma las acciones necesarias al momento de crear la base de datos, tomando en cuenta el volumen de las transacciones y las



restricciones que tiene que especificar en el acceso a los datos; debe de tomar y aprovechar al máximo los elementos ofrecidos por el RDBMS.

1.8.1 Seguridad de la Información

La seguridad de la información tiene como fin la protección de la información y de los sistemas de información del acceso, uso, divulgación, interrupción o destrucción no autorizada.

La seguridad de la información protege a la información de una amplia gama de amenazas, a fin de garantizar la continuidad del negocio. No importa la forma en que se adquiere la información, o los medios por los cuales se distribuye o almacena, siempre debe de ser protegida en forma adecuada.

Es muy importante la seguridad de la información, ya que es un recurso que, como el resto de los importes activos comerciales, tiene valor para una organización y por consiguiente debe de ser debidamente protegida. Dicha información puede existir en muchas formas; puede ser impresa o escrita en papel, almacenada electrónicamente, puede ser transmitida por correo electrónico, o simplemente expuesta en imágenes o en una conversación.

La seguridad en cómputo son los mecanismos tecnológicos que protegen los sistemas de cómputo y todo lo asociado con ellos.

Entonces, la seguridad en cómputo está enfocada a sistemas de cómputo y redes de datos. Mientras que la seguridad de la información está enfocada al tratamiento y uso de la información, involucrando sistemas de cómputo, redes de datos, personas y procesos.

El objetivo de la seguridad es lograr adquirir, almacenar, procesar y transmitir información, preservando los servicios de:

- Confidencialidad, esto quiere decir, que la información sólo la conozcan quienes tienen derecho a ello.
- Integridad, esto es, que la información no sea alterada sin autorización.
- Disponibilidad, esto quiere decir, que los usuarios legítimos puedan usar la información cuando la requieran.

La seguridad no puede ser la meta de una organización, ni un aspecto más a considerar. La seguridad no se resuelve con un producto; y nunca nos encontraremos con un sistema 100 % seguro.



Eugene Spafford distinguido científico e investigador y uno de los principales expertos en seguridad informática, citó *“El único sistema totalmente seguro es aquel que está apagado, desconectado de la red, guardado en una caja fuerte de titanio, encerrado en un bunker de concreto, rodeado por gas venenoso y cuidado por guardias muy bien armados y muy bien pagados. Aún así, no apostaría mi vida por él.”*

Los errores más comunes que se comenten en cuanto a la seguridad son:

- El designar a personas no capacitadas para mantener la seguridad, el no capacitarlas ni darles tiempo de capacitarse.
- No entender la relación que existe entre la seguridad y los problemas de funcionamiento. Entienden la seguridad física, pero no ven las consecuencias de una mala seguridad informática.
- No entender cuánto dinero vale la información y qué tanto depende de ella la reputación corporativa.
- Autorizar soluciones reactivas y parches de corto plazo, tan es así que los problemas reaparecen rápidamente.
- El suponer que los problemas desaparecerán si no se les hace caso.

Podemos confiar en que un sistema de cómputo es seguro si se puede confiar en que él y su software se comportarán como se espera que lo hagan, y que la información almacenada en él se mantendrá inalterada y accesible durante tanto tiempo como su dueño lo desee.

1.8.2 Ataques y sus diferentes tipos

Un ataque es cualquier acción que explota una vulnerabilidad. Existen diferentes tipos de ataques.

Los pasivos, en estos observa comportamientos o lee información, sin alterar ni el estado del sistema ni la información. Solo afecta la confidencialidad del sistema o de la información.

Los activos, es en estos donde los atacantes modifican o afectan la información o inclusive el estado del sistema. Afectan no sólo la confidencialidad sino también la integridad y la autenticidad de la información o del sistema.



Un atacante puede ser cualquier entidad que realiza un ataque, puede ser una persona, un proceso o hasta algún dispositivo.

Existen cuatro tipos de amenazas principales a los sistemas que explotan las vulnerabilidades de los activos en el sistema, los cuales se describen en la tabla 1.12.

Tipo de Amenaza	Definición
Interrupción	Un activo del sistema se pierde, se hace no disponible o inutilizable.
Intercepción	Alguna parte no autorizada logra acceso a un activo del sistema.
Modificación	Cuando una parte no autorizada logra acceso al activo del sistema y puede manipular ese activo.
Fabricación	Una parte no autorizada puede fabricar objetos falsos en un sistema.

Tabla 1.12 : Tipos de amenazas

1.8.3 Seguridad en Bases de Datos

La seguridad en Bases de Datos es muy importante pues con ella se protege el acceso a la información sensible y a los activos digitales de una organización. Los sistemas de bases de datos son sistemas extremadamente complejos y con gran dificultad para configurar y asegurar.

Una base de datos con un bajo nivel de seguridad compromete no solamente a la base de datos misma, sino también al sistema operativo y a otros sistemas relacionados.

Los objetivos de seguridad en base de datos comprenden medidas para evitar:

- La liberación inapropiada de información (Pérdida de secrecía y confidencialidad).
- La modificación inapropiada de datos (Pérdida de integridad).
- La negación del servicio (Pérdida de disponibilidad).
- El robo o fraude.

Las recomendaciones para tener buenas prácticas en seguridad de bases de datos son:

- Usar un sistema de detección de intrusos, especialmente en servidores de bases de datos en línea y de alto riesgo.
- Cambiar las contraseñas de las cuentas creadas por omisión durante la instalación y asignar contraseñas fuertes a las mismas.
- Deshabilitar las cuentas de invitado y las cuentas de demostración definidas durante la instalación. Y eliminar estas cuentas en la base de datos de producción.



- Mantener actualizado el RDBMS con las versiones más recientes de software y de los parches de seguridad liberados por el fabricante del software.
- No permitir que las aplicaciones consulten o manipulen directamente la base de datos mediante instrucciones SELECT, INSERT, UPDATE o DELETE. Usar procedimientos almacenados en su lugar.
- En las aplicaciones, restringir la ejecución de instrucciones SQL dinámico.
- Impedir que las aplicaciones acepten instrucciones de SQL de los usuarios y las ejecuten dentro de la base de datos.
- Habilitar la Auditoría de acceso al sistema operativo y al servidor de bases de datos. Revisar el registro de auditoría buscando los eventos fallidos de acceso y buscar tendencias con la finalidad de detectar posibles intrusos.
- Monitorear cuidadosamente los registros (logs) de error y de eventos y disparar automáticamente alertas relacionadas con la seguridad y con errores. Proteger los archivos de registro (logs) mediante permisos apropiados de sistema operativo.
- En ambientes de bases de datos distribuidas, eliminar el acceso a los servidores que no se utilicen. Utilizar cuentas de acceso con mínimos privilegios para los servidores relacionados.
- Almacenar los archivos utilizados para la carga masiva de datos o por lotes en un directorio con los permisos apropiados; y eliminar dichos archivos una vez que hayan sido utilizados.
- Para asegurar la replicación de datos sobre Internet o sobre una red de área amplia (WAN), se deberá implementar una red privada virtual (VPN).
- Definir y aplicar una política de respaldo periódico, almacenar los medios de respaldo en un lugar seguro. Realizar regularmente restauraciones de la base de datos a partir de los respaldos.

1.9. Performance and Tuning

El término Performance se refiere al número de transacciones en un lapso determinado de tiempo, que la base de datos puede ejecutar. Mientras que el término Tuning se refiere al ajuste y modificación de parámetros para optimizar el performance y el rendimiento del servidor.



El Performance & Tunning describe un grupo de actividades para optimizar y homogeneizar la relación de una base de datos; se refiere al diseño de la base de datos, a la selección del mejor Sistema Manejador de Bases de Datos Relacional (RDBMS), al Sistema Operativo donde correrá el RDBMS y al CPU donde se ejecuta.

El objetivo principal del Performance & Tunning es el de maximizar el uso de los recursos del sistema para realizar el trabajo tan eficiente y rápidamente como sea posible. La mayoría de los sistemas están diseñados para administrar el trabajo de forma eficiente, pero es posible mejorar el rendimiento del sistema mediante la personalización de la configuración y en consecuencia del RDBMS. Para lograr el máximo rendimiento y la optimización deseada en un servidor se deben de configurar como mínimo tres aspectos; la memoria, el disco duro y el límite de recursos para cada usuario.

La creación de Cachés de memoria exclusivos para algunas tablas que contienen información fija, optimiza las consultas hacia la base de datos. Una de estas tablas con información fija, pueden ser los catálogos de referencia. Cuando se hace una consulta, el RDBMS va directamente al disco duro y hace un acierta cantidad de lecturas físicas; después esas consultas las sube al caché de memoria y de ahí realiza el mismo número de lecturas lógicas. Con la creación de cachés y la asignación de éstos a tablas, el subir las consultas al caché sólo se hará la primera vez, y las siguientes sólo va a leer directamente desde el caché. En conclusión, al asociar un caché a una tabla se ahorra tiempo y lecturas hacia la base de datos.

1.10. Modelo Orientado a Objetos

Los elementos de un sistema de Información son:

- Software
- Hardware
- Personas
- Documentación
- Procesos
- Base de Datos (Información).

1.10.1 Paradigma Orientado a Objetos

Paradigma es una forma o patrón de pensamiento, que establece reglas y límites, un paradigma es una forma de resolver problemas. Un Objeto es una cosa tangible o conceptual, es un elemento que tiene características y comportamiento.



El Paradigma Orientado a Objetos es una filosofía para el desarrollo de sistemas basado en el modelado del mundo real a través de la identificación de objetos. Surgió inicialmente como un enfoque para la programación pero se ha extendido a todo el ciclo de desarrollo de sistemas.

Las ventajas del paradigma Orientado a Objetos son:

- Se pueden reutilizar.
- Su desarrollo es más rápido.
- Son de mayor calidad y de fácil mantenimiento.
- Tienen una gran modularidad y de costo reducido.
- Utilizan mejores estructuras de información.

1.10.3 Lenguaje de Modelado Unificado (UML)

UML es un estándar en la industria de construcción de software, que permite modelar y documentar la arquitectura de una aplicación. Para comprender mejor lo que es UML, es recomendable definir lo que es un modelo y el porqué de modelar.

Se define a un modelo como la representación de algo real, el cual capta los aspectos importantes de lo que estamos representando, desde cierto punto de vista, y simplifica u omite el resto. Los modelos pueden ser físicos, gráficos, matemáticos, etcétera.

Debemos de modelar, porque con esto nos permite dividir un problema complejo en problemas menores, nos permite visualizar un sistema desde varias perspectivas y con ello podemos entender y dimensionar el problema, logrando poder probar varias soluciones. Al modelar podemos abstraer las características, componentes y estructura de algo a construir; podemos detectar fallas, inconsistencias y prevenir cambios. Es menos costoso construir un modelo que un sistema.

Por lo tanto, el lenguaje de modelado unificado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

El UML fue desarrollado en un esfuerzo para simplificar y consolidar las notaciones de desarrollo orientado a objetos que habían surgido principalmente de Booch, Rumbaugh y Jacobson. En el año de 1996 surge la versión 1.0 del UML, un año después surge el OMG (Grupo de Gestión de Objetos). Los beneficios de UML son:

- Facilita la comunicación con otros.
- Es un estándar en la industria de construcción de software.
- Permite modelar y documentar la arquitectura de una aplicación.



- Existen herramientas en el mercado para modelar y generar código a partir de UML.

La figura 1.17 se ilustra cómo es un diagrama de UML.

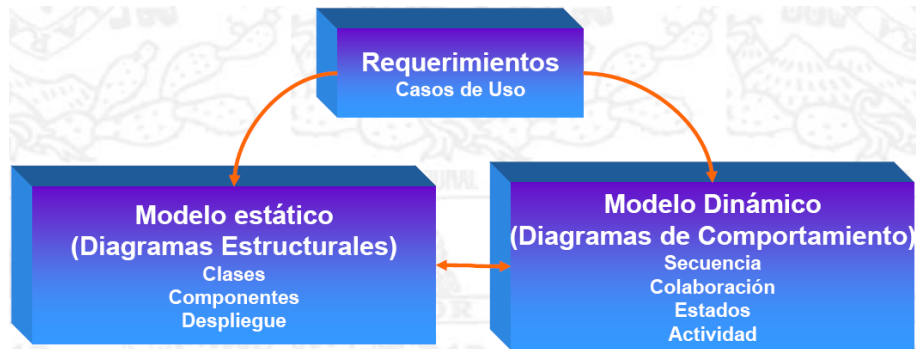


Figura 1.17 : Diagrama UML

Un requerimiento es una condición o característica que se debe satisfacer. En el contexto de desarrollo de software, un requerimiento es una característica identificable expresada en términos de funcionalidad o desempeño que un sistema debe poseer para lograr su objetivo. Existen diferentes niveles de requerimiento, los cuales se ilustran en la figura 1.18.



Figura 1.18 : Niveles de Requerimiento

También existen diferentes tipos de requerimientos; los cuales son:

- Requerimientos funcionales; estos requerimientos describen lo que el sistema debe de hacer, es decir, especifican acciones que el sistema debe ser capaz de realizar (funcionalidad).
- Requerimientos no funcionales; estos requerimientos describen atributos del sistema o atributos del ambiente del sistema.



Hay distintas fuentes de requerimientos, como por ejemplo, las conversaciones de un cliente con su usuario, entrevistas con expertos, experiencias de sistemas anteriores, conocimiento personal del dominio.

El Lenguaje de Modelado Unificado, comprende la realización de los siguientes diagramas:

- **Diagrama de Casos de Uso:** Los casos de uso nos permiten modelar y especificar los requerimientos funcionales de un sistema. Un diagrama de casos de uso modela las posibles formas en que un sistema puede ser usado. Los casos de uso en conjunto representan toda la funcionalidad del sistema que se esté modelando.
- **Diagrama de Actividad:** Los diagramas de actividad describen el comportamiento de un sistema y procesos en paralelo. Es la forma en que el sistema actúa y reacciona a su entorno. Los diagramas de actividad muestran, las secuencias de actividades, las acciones paralelas y las dependencias entre ellas. Se recomienda utilizar los diagramas de actividad en:
 - Análisis del comportamiento de casos de uso, basta con diagramar el flujo principal y en caso necesario, algunos flujos alternos.
 - Cuando los procesos no son secuenciales y pueden ser representados en forma paralela.
- **Diagrama de Clases:** Representan las características estructurales, es decir, los requisitos funcionales de un sistema, los servicios que el sistema debe proporcionar a sus usuarios finales. Los diagramas de clases se consideran la columna vertebral de los métodos orientados a objetos. El término “clase” se utiliza como una descripción de un grupo de objetos que comparten propiedades (atributos), comportamiento (operaciones) y relaciones con otros objetos (asociaciones).
- **Diagrama de Secuencia:** establecen el comportamiento del sistema, ilustran las interacciones de objetos que están ordenadas de acuerdo a la secuencia de tiempo de los mensajes intercambiados.
- **Diagramas de Colaboración:** Estos diagramas muestran el flujo de datos entre objetos. Es otra forma de representar los mensajes intercambiados por un conjunto de objetos a través de sus ligas.
- **Diagrama de Estado:** Describe el comportamiento dinámico de los objetos en un cierto plazo, desde su creación hasta su fin. Muestran la historia de vida de una clase, los eventos que causan una transición de un estado a otro y las acciones que resultan de un cambio de estado.
- **Diagrama de Componentes:** Son un grupo de objetos o partes más pequeñas en las que se divide un subsistema, que interaccionan entre ellos y se combinan para dar un servicio. Un componente es una unidad de código fuente que sirve como bloque constructor para la estructura física de un sistema. Muestran la organización y distribución de clases y objetos en componentes y paquetes



de software. Se pueden utilizar para modelar código fuente, modelar versiones ejecutables, modelar bases de datos físicas y conocer que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

- **Diagramas de Distribución:** Los diagramas de distribución son creados para mostrar los diferentes nodos (procesadores y dispositivos en el sistema). Los nodos modelan la topología del hardware sobre el que se ejecuta el sistema. En la figura 1.19 se observan los diferentes nodos en el diagrama de distribución.



Figura 1.19 : Diagrama de distribución.

1.11. Tópicos Avanzados de Bases de Datos

1.11.1 Minería de Datos

La minería de datos es un paso consistente en el uso de algoritmos concretos que generan una enumeración de patrones a partir de los datos procesados, tal como se ilustra en la figura 1.20.

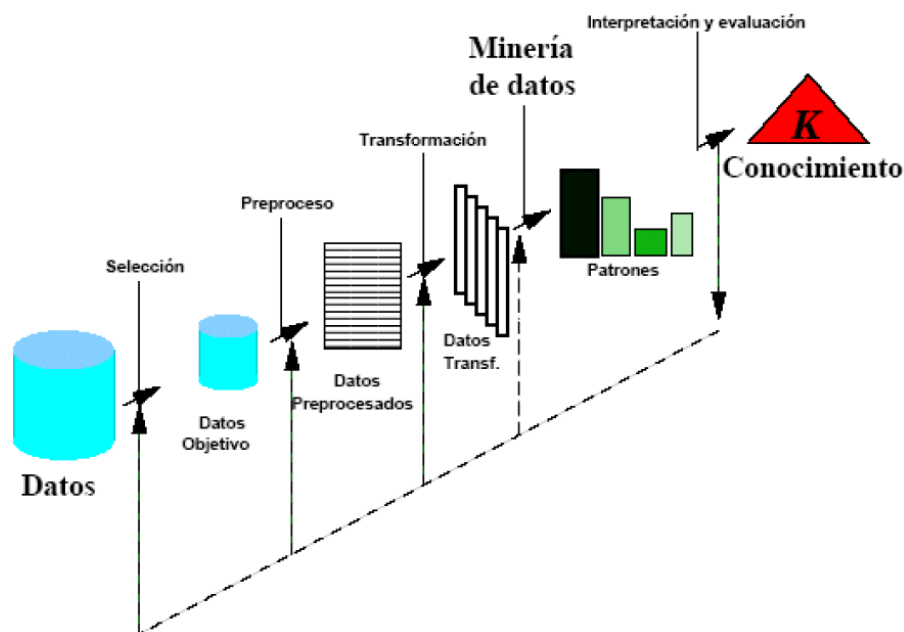


Figura 1.20: Minería de Datos

La minería de datos tiene cuatro etapas principales, las cuales son:

- **Determinación de los objetivos:** Delimitar los objetivos que el cliente desea bajo la orientación del especialista en Minería de Datos.
- **Procesamiento de los datos:** Se refiere a la selección, la limpieza, el enriquecimiento, la reducción y la transformación de las bases de datos.
- **Determinación del modelo:** Se comienza realizando un análisis estadístico de los datos, y después se lleva a cabo una visualización gráfica de los mismos para tener una primera aproximación. Según los objetivos planteados y la tarea que debe llevarse a cabo, pueden utilizarse algoritmos desarrollados en diferentes áreas de la inteligencia artificial.
- **Análisis de los resultados:** Verifica si los resultados obtenidos son coherentes y los coteja con los obtenidos por el análisis estadístico y de visualización gráfica. El cliente si son novedosos y si le aportan un nuevo conocimiento que le permita considerar sus decisiones.

La Minería de Datos se divide en dos modelos:

- **Modelo Predictivo:** realiza una predicción acerca de los valores de los datos usando la comprensión de los resultados encontrados desde diferentes datos. Las tareas de este modelo incluyen la clasificación, regresión, análisis de series de tiempo y predicción.



- **Modelo Descriptivo:** Éste modelo identifica patrones o relaciones en los datos. A diferencia del modelo predictivo, sirve como un modelo para explorar las propiedades de los datos examinados, y no para predecir nuevas propiedades. El agrupamiento (clustering), sumarización, reglas de asociación y descubrimiento de secuencias son vistas como tareas del modelo descriptivo.

La figura 1.21 ilustra los modelos de la minería de datos.



Figura 1.21 : Modelos de minería de datos.

Las tareas básicas de Minería de Datos son:

- **Clasificación:** Traza mapas de datos dentro de grupos o clases predefinidos.
- **Regresión:** Es usada para un mapa de un detalle de un dato para la predicción de un valor real de una variable.
- **Predicción:** Muchas aplicaciones de minería de datos del mundo real se pueden ver, prediciendo el dato futuro basado sobre los estados pasados y actuales del dato.
- **Sumarización:** Son mapas de datos dentro de subseries o subconjuntos con simples descripciones asociadas. Deriva o extrae información descriptiva acerca de la base de datos.



- **Agrupamiento:** Es un proceso de dividir un conjunto de datos en grupos mutuamente excluyentes de tal manera que cada miembro de un grupo esté lo “más cercano” posible a otro, y grupos diferentes estén lo “más lejos” posible uno del otro.
- **Reglas de asociación:** Establece asociaciones en base a los perfiles de los clientes sobre los cuales se está realizando la minería de datos.

1.11.2 Datawarehouse

Datawarehouse es una combinación de conceptos y tecnología que cambian significativamente la manera en que es entregada la información a la gente de negocios. El objetivo principal es satisfacer los requerimientos de información internos de la empresa para una mejor gestión, con eficiencia y facilidad de acceso. Las características de datawarehouse son:

- **Orientado al Tema:** Una primera característica de DW es que la información se clasifica en base a los aspectos que son de interés para la empresa. Siendo así, los datos están en contraste con los clásicos procesos orientados a las aplicaciones.
- **Integrado:** El aspecto más importante del ambiente DW es que la información encontrada al interior está siempre integrada.
- **De tiempo variante:** Toda la información de DW es requerida en algún momento. Esta característica básica de los datos en un depósito, es muy diferente de la información encontrada en el ambiente operacional.
- **No volátil:** La información es útil sólo cuando es estable. Los datos operacionales cambian sobre una base momento a momento. La perspectiva más grande, esencial para el análisis y la toma de decisiones, requiere una base de datos estable.

La estructura de Datawarehouse se divide en cuatro grupos, los cuales son:

Detalle de datos actuales: En gran parte, el interés más importante radica en el detalle de los datos actuales, debido a que refleja las ocurrencias más recientes, las cuales son de gran interés. Es voluminoso, ya que se almacena al más bajo nivel de granularidad. Casi siempre se almacena en disco, el cual es de fácil acceso, aunque su administración sea costosa y compleja.

Detalle de datos antiguos: Son aquellos que se almacenan sobre alguna forma de almacenamiento masivo, no es frecuente su acceso y se almacena a un nivel de detalle, consistente con los datos detallados



actuales. Mientras no sea prioritario el almacenamiento en un medio de almacenaje interno, a causa del gran volumen de datos unido al acceso no frecuente de los mismos, es poco usual utilizar el disco como medio de almacenamiento.

Datos ligeramente resumidos: Es aquella que proviene desde un bajo nivel de detalle encontrado al nivel del detalle actual. Este nivel del DW casi siempre se almacena en disco.

Metadata: Técnicamente se define a metadata como una descripción del contenido. Es decir que lo que se busca con metadata es una manera de adjuntar a una información, no una identificación, sino una descripción de lo que es. Esta descripción debe de cumplir por lo menos con dos requisitos; por un lado ser accesible desde distintos entornos y por otro lado ser clara y precisa.

El término Datawarehouse se utiliza indistintamente para hablar de la arquitectura en sí como también para uno de los componentes que la conforman, específicamente el que se tiene relación con el almacenamiento físico de los datos. La arquitectura de Datawarehouse comprende los siguientes aspectos:

- Datos operacionales.
- Extracción de datos.
- Transformación de datos.
- Carga de datos.
- Datawarehouse
- Herramientas de acceso al componente de almacenamiento físico.

En la figura 1.22 se puede observar la arquitectura de Datawarehouse.

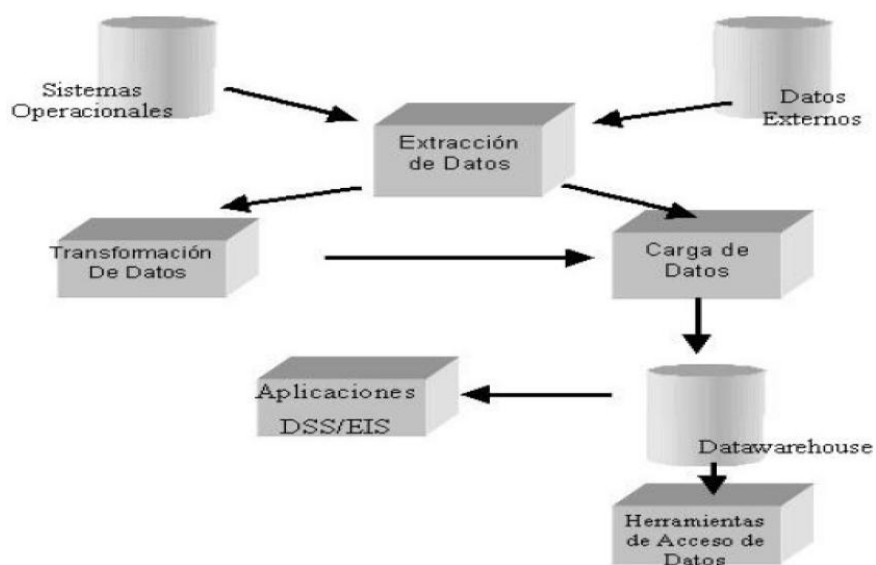


Figura 1.22: Arquitectura de Datawarehouse



1.11.3 Bases de Datos Multidimensionales

Las bases de datos multidimensionales se utilizan principalmente para crear aplicaciones OLAP y pueden verse como bases de datos de una sola tabla, su peculiaridad es que por cada dimensión tienen un campo (o columna), y otro campo por cada métrica o hecho. En general la estructura básica está definida por dos elementos; las tablas y los esquemas.

1.11.4 Bases de Datos Inteligentes

Las bases de datos inteligentes representan una tecnología para la administración de la información que fue evaluada como resultado de la integración de las bases de datos tradicionales con otros campos como lo son:

- Programación orientada a objetos.
- Sistemas expertos.
- Recepción de información en línea.
- Hypermedia

Para ilustrar lo anterior se muestra la figura 1.23.



Figura 1.23 : Bases de datos inteligentes.

Las características de las bases de datos inteligentes son:

- Modelo de datos basado en el conocimiento y orientado a objetos.



- Bases de datos integradas y motor de inferencia.
- Búsquedas sensitivas al contexto y/o de estructura sensitiva.
- Soporte de múltiples medios de almacenamiento.
- Administración inteligente de versión, recuperación y resistencia.
- Soporte de transacciones y concurrencia.
- Optimización de query's.



CAPÍTULO 2: PLANTEAMIENTO DEL PROBLEMA Y POSIBLE SOLUCIÓN

2.1 Historia de la telecomunicación en México y de la empresa TELCEL

La comunicación en el hombre ha tenido una función poderosa, la de expresar sus sentimientos y satisfacer sus necesidades. La humanidad ha desplegado gran energía para comunicarse, ha utilizado y desarrollado diferentes tipos de señales que agrupa en diferentes tipos de lenguajes y expresiones. En el futuro el sistema de telefonía celular con tecnología digital y el empleo de convergencia de servicios celulares se consolidarán como el medio de telecomunicación universal.

Una de las características más apreciadas en la comunicación es la de poder ejercerla con libertad de movimiento. Los sistemas de comunicación celular son una de las máximas expresiones de la movilidad en las telecomunicaciones.

El físico británico James Clerk Maxwell estableció en la década de 1860 la teoría de las ondas electromagnéticas, teoría que sustentó los desarrollos de la transmisión inalámbrica. Posteriormente Guglielmo Marconi, inventó el sistema de radio en el año de 1896, con este acontecimiento iniciaría la era de las telecomunicaciones.

El sistema celular sobrepasa las características de comunicación telefónica, permite establecer comunicación en cualquier momento y en cualquier lugar dentro del área de servicio celular. Este moderno sistema de telecomunicación revolucionó a la telefonía convencional, dejando atrás los cables y sustituyéndolos por frecuencias de radio, dando la opción de servicio telefónico móvil.

El término celular se refiere a la forma hexagonal en que están agrupadas las zonas de servicio. Organizaciones especializadas, proveedores de equipos y servicios han debatido en la implantación de tecnologías con mayores capacidades y funcionalidades, de tal forma que favorezcan en mercado de las telecomunicaciones. A través del tiempo, el desarrollo tecnológico y la lucha por la dominación de mercados han generado importantes estándares y tecnologías. Los sistemas celulares pueden clasificarse considerando el tipo de acceso, la cobertura, la tasa de transmisión de datos y eficiencia en el manejo del espectro.

En el mes de febrero del año 1956 se funda “PUBLICIDAD TURISTICA S.A.” comercializando directorios telefónicos, sección blanca y sección amarilla. En el año de 1974 se cambia la razón social a “DIRECTORIOS PROFESIONALES S.A. DE C.V. (DIPSA); en esta empresa se editan directorios como el directorio azul por calles, directorio de la construcción, turismo y otros. Para el año de 1977, se solicitó a la Secretaría de Comunicaciones y Transportes (SCT) una concesión para instalar, operar y explotar un sistema de radiotelefonía móvil en el distrito federal. Pocos años después en 1984, cambia



nuevamente de razón social a “Radiomóvil DIPSA S.A. de C.V.”, con su marca “TELCEL” se propone utilizar los sistemas más avanzados y colocar a México como un país con los servicios de la más alta calidad. En años más recientes, en 2008 para ser exactos “TELCEL” alcanzó una marca histórica llegando a 50 millones de clientes, todo eso no se hubiese logrado si tras la marca TELCEL no existieran cientos de empleados trabajando día con día brindando sus conocimientos y habilidades para que cada persona que cuente con un celular cuente con el mejor servicio de telefonía celular.

2.2 Sistema administrador de nómina para TELCEL

Hoy en día el tener un buen manejo del departamento de nómina de cualquier empresa es fundamental; ya que éste departamento es el encargado del manejo de lo mas importante dentro de cualquier empresa, el capital humano. Es necesario contar con un sistema en el cual este depositada la información referente al personal que trabaja en la empresa, el salario de cada empleado, la fecha en que fue contratado, etcétera.

Es por ello que se plantea un sistema que administre la base de datos de los empleados que laboran en una empresa; en este caso se dirige hacia una empresa de telecomunicaciones como lo es Telcel, este sistema de administración de la base de datos propone una base de datos formada por tres tablas: tabla departamento, tabla cargo, y tabla empleado; cada una de ellas con diferentes campos para poder tener un mejor manejo de la información. Las tablas tendrán los campos que se muestran en la tabla 2.1.

Nombre de la Tabla	Campos
Departamento	id_departamento nombre
Cargo	Id_cargo nombre
Empleado	id_empledo nombre apellido_paterno apellido_materno edad sexo sueldo porcentaje_comision fecha_contratacion id_departamento id_cargo

Tabla 2.1: Tablas y sus campos para la base de datos.



CAPÍTULO 3: IMPLANTACIÓN DEL SISTEMA

3.1 Funciones del Sistema

El sistema de administración de base de datos para el departamento de nómina de Telcel, se muestra a continuación.

Para ingresar al sistema, se debe de proporcionar el nombre de usuario asignado a la base de datos y la contraseña, a manera de ejemplo se ilustra cómo se accede al sistema con el usuario SYSTEM; la pantalla de acceso al sistema se ilustra en la figura 3.1.



Figura 3.1 : Pantalla de acceso al sistema.

Una vez que se ha ingresado el nombre de usuario y su contraseña, el sistema muestra una pantalla de home en la cual se encuentran los menús principales, dichos menús se muestran en la figura 3.2.



Figura 3.2 : Home del sistema.

Cada uno de los menús, lista opciones dependiendo de lo que se requiera realizar. Para el caso de usuarios; despliega un menú en el cual se pueden elegir diferentes opciones como puede ser: alta de usuario, baja de usuario, modificación de usuario, permisos a un usuario, y los usuarios existentes; el menú Usuarios se ilustra en la figura 3.3. Para el menú Tablas e índices, despliega un menú en el cual se puede elegir dar de alta una tabla, o en caso contrario dar de baja una tabla, o asignarle permisos a una tabla; el menú Tablas e Índices se ilustra en la figura 3.4. En el caso del menú Bases de datos, se pueden elegir tres distintas opciones, las cuales nos permiten crear una base de datos, borrar una base de datos, o realizar una consulta por medio del lenguaje SQL; el menú Bases de Datos se ilustra en la figura 3.5.



Figura 3.3: Submenú Usuarios.



Figura 3.4: Submenú Tablas e Índices



Figura 3.5: Submenú Bases de Datos

Al seleccionar una opción de los diferentes menús nos abre una nueva ventana. A continuación veremos cada pantalla de los tres menús principales.



Para dar de alta un usuario se da un clic en el menú Usuario y se selecciona Alta, al hacer esto nos abrirá una nueva ventana en donde se tiene que llenar un formulario en el cual nos pide el nombre del nuevo usuario, la contraseña de este usuario y la confirmación de la contraseña, por último el rol que se le asignará al nuevo usuario, lo anterior se ilustra en la figura 3.6.

Nombre de Usuario	<input type="text"/>
Contraseña	<input type="password"/>
Confirmar Contraseña	<input type="password"/>

Rol de Usuario: Connect

Crear Usuario Cancelar

Figura 3.6: Alta de usuario.

Si se quisiera dar de baja un usuario se da un clic en el menú Usuario y se selecciona Baja, al dar clic nos abre una ventana en la cual se selecciona el usuario a dar de baja, la pantalla de baja de usuario se ilustra en la figura 3.7.

Usuario: SYSTEM

ELIMINAR USUARIO DE LA BASE DE DATOS

Usuario: ATD

Eliminar Cancelar

Figura 3.7: Baja de usuario



Para modificar un usuario solo hay que dar clic en el menú Usuario y elegir la opción Modificación, se abrirá una nueva ventana en donde hay que elegir el usuario a modificar, se tiene que ingresar una nueva contraseña para el usuario seleccionado y confirmar la contraseña ingresada, por último se debe de seleccionar el nuevo rol del usuario a modificar; la pantalla de modificación de usuario se ilustra en la figura 3.8.

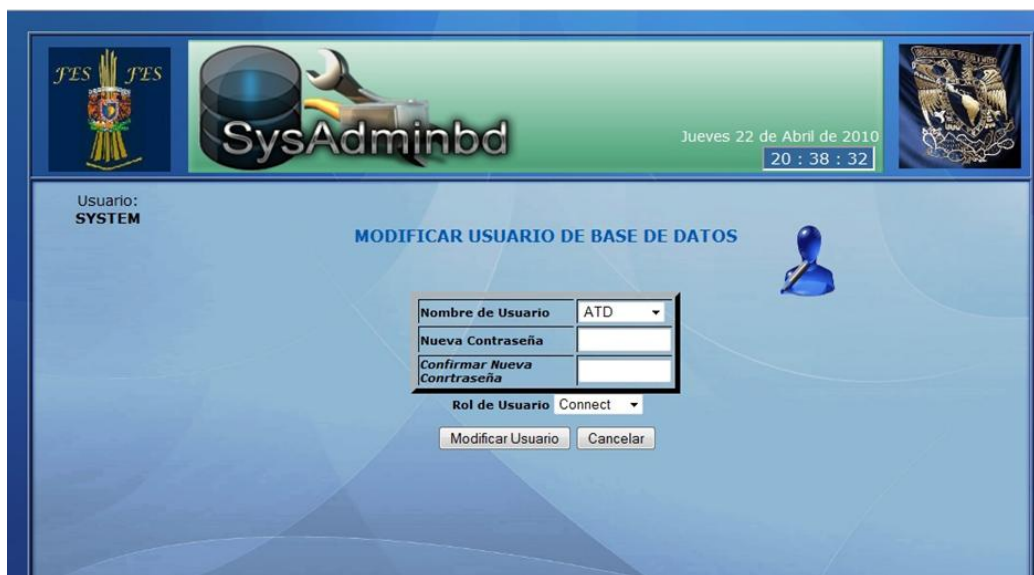


Figura 3.8: Modificación de usuario

En la figura 3.9 se ilustra la pantalla de permisos a un usuario, en ésta se puede modificar el tipo de usuario que se desee ya sea solo de conexión a la base de datos, o que se requiera que el usuario se le asigne el perfil de DBA, otro rol que se le puede asignar a un usuario es el de recurso el cual le permite al usuario administrar los recursos de la base de datos.



Figura 3.9: Permisos a usuarios



En la última opción del menú Usuarios nos muestra una ventana en la cual se listan los usuarios que existen en la base de datos, nos muestra el nombre de los usuarios, la fecha en que fueron creados, el ID del usuario y el rol que tiene cada usuario; la pantalla de usuarios se puede observar en la figura 3.10.

USUARIO	ID	CREADO	TABLESPACE	ROL
ATD	36	21/04/10	USERS	DBA
HR	33	07/02/06	USERS	CONNECT
HR	33	07/02/06	USERS	RESOURCE
SYSTEM	5	07/02/06	SYSTEM	DBA

Figura 3.10: Usuarios de la Base de datos

Otro de los menús principales es el de Tablas e Índices, este menú despliega tres opciones a elegir: alta de una tabla, baja de una tabla o permisos a una tabla. En la figura 3.11 se puede observar la pantalla que muestra el sistema cuando se selecciona “Alta” en el menú Tablas e Índices, en ésta nueva pantalla se tiene que llenar un formulario para especificar el número de campos de la nueva tabla, el nombre que se le dará a esa tabla; el formulario también nos pide que ingresemos el nombre de cada columna de la nueva tabla, así como el tipo de datos, longitud, decimal, si es nulo o no el campo y la llave primaria de la nueva tabla.

Porporcione el Número de Campos de la Tabla:

Nombre de la Tabla

Nombre Columna	Tipo de Dato	Longitud	Decimal	¿Es Nulo?	¿Es Llave Primaria?
<input type="text"/>	Number	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text"/>	Number	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text"/>	Number	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>

Crear Tabla Cancelar

Figura 3.11: Formulario para crear una nueva tabla.



Para dar de baja una tabla de la base de datos, solo basta con seleccionar la opción “Baja” del menú Tablas e Índices, se abrirá una nueva ventana en la cual solo basta con seleccionar la tabla que se desee eliminar; lo anterior se ilustra en la figura 3.12.



Figura 3.12: Eliminar tablas.

Si se necesita dar permisos a un usuario sobre una tabla, se debe de seleccionar la opción “permisos” del menú Tablas e Índices, esta opción abrirá una nueva pantalla en la cual se debe de seleccionar la tabla deseada el usuario al que se le van a dar los permisos sobre la tabla, y el tipo de permiso, puede ser SELECT, UPDATE, INSERT, DELETE, o ALL que los cuatro anteriores. La figura 3.13 ilustra la pantalla de permisos a una tabla.



Figura 3.13: Permisos a una tabla.

El último bloque de menús es el de “Bases de Datos”. Es este menú se presentan tres opciones: Bases de Datos existentes, Borrar y Consultas SQL. La figura 3.14 ilustra la ventana que se abre al seleccionar la



primera opción, en esta ventana se muestran las bases de datos que existen, indicando el ID de la base de datos, el nombre, la versión, el host_name y el estatus de la base de datos.



Figura 3.14: Bases de datos existentes

Para eliminar una base de datos, se selecciona la opción borrar del menú “Base de Datos” ésta opción nos abrirá una nueva ventana en la cual se puede seleccionar la base de datos que se quiera borrar; lo anterior se ilustra en la imagen 3.15.



Figura 3.15: Borrar una base de datos.

Por último, el sistema permite realizar consultas SQL, esto es posible seleccionando la opción “Consultas SQL” del menú Bases de Datos, al hacer clic en la opción se abre una nueva ventana en la cual hay un cuadro de entrada de texto, y es ahí donde se puede escribir el query que se desee. La figura 3.16 ilustra la ventana de consultas SQL.



Figura 3.16: Crear consultas SQL.

3.2 Tablas de la base de datos de Telcel

Las tablas de la base de datos para el departamento de nómina de Telcel tienen la información que aparece en las tablas 3.1, 3.2 y 3.3., dichas tablas contienen la información para la tabla “Departamento”, “Cargo” y “Empleado” respectivamente.

Id_departamento	nombre
1	Ventas
2	Recursos Humanos
3	Diseño
4	Sistemas
5	Administración
6	Redes
7	Calidad

Tabla 3.1: Tabla Departamento

Id_cargo	nombre
1	Empleado
2	Jefe de Departamento
3	Coordinador
4	Ejecutivo de cuenta
5	Subdirector
6	Director

Tabla 3.2: Tabla Cargo



id_employado	nombre	apellido_paterno	apellido_materno	edad	sexo	sueldo	porcentaje_comision	fecha_contratacion	id_departamento	id_cargo
1	Gilberto	Aparicio	Guerrero	25	H	7000	2	20/03/2001	4	1
2	Javier	Rojas	Hernández	30	H	7500	2	05/03/2001	4	1
3	Alma	Guerra	Zarate	19	M	10000	2	10/05/2000	4	1
4	Carolina	Fernández	Fernández	32	M	15000	2	01/12/2003	4	2
5	Alberto	Ibáñez	Becerril	32	H	25000	2	16/05/2004	4	2
6	Carlos	López	López	32	H	27000	2	16/05/2003	4	3
7	Arturo	Peña	Higareda	45	H	18000	5	12/10/2002	1	4
8	Javier	Bárceñas	Moreno	29	H	18000	10	20/03/2005	1	4
9	Ernesto	Almaraz	García	32	H	20000	2	15/01/2007	1	2
10	Susana	García	Santos	31	M	18000	2	30/11/2002	3	1
11	Fernanda	Ramírez	Jiménez	18	M	18000	2	30/08/2003	3	1
12	Roxana	Guerrero	Vázquez	25	M	26500	2	15/02/2009	3	2
13	Manuel	Alavez	Núñez	35	H	12600	2	08/04/2005	2	2
14	Angeles	Nolasco	Carmona	29	M	6000	2	15/07/2000	5	1
15	Teresa	Álvarez	Martínez	37	M	12000	2	28/10/2009	5	2
16	Alberto	González	Guizar	56	H	26500	2	01/05/2003	NULL	5
17	Ricardo	Pérez	Rosas	58	H	50500	2	05/10/2001	NULL	6
18	Martha Elena	Ramírez	Torres	25	M	26500	2	17/11/2004	3	2
19	Elvira	Ríos	Montes	50	M	18000	7	18/07/2005	1	4

Tabla 3.3: Tabla Empleado



CONCLUSIONES

El principal objetivo de la Universidad Nacional Autónoma de México es la formación de profesionales útiles a la sociedad, capaces de identificar, conceptualizar y resolver los problemas más variados de los ámbitos nacional, regional y local. Este objetivo es alcanzado día a día gracias a la gran calidad de la enseñanza en la Universidad, la cual es innegable y se reafirma en cada clase, curso o taller en cualquiera de sus modalidades. En esta ocasión el Diplomado de Bases de Datos impartido por la Dirección General de Servicios de Computo Académico de la UNAM no fue la excepción, en cada una de sus clases se rebaso mis expectativas como alumno y se cumplió al 100% cada uno de los objetivos planteados.

A lo largo de cada uno de los módulos del diplomado fui adquiriendo el conocimiento, las habilidades, estrategias y técnicas, para la administración de bases de datos.

El ser administrador de bases de datos requiere de una gran preparación profesional, se deben de tener conocimientos sólidos en aspectos fundamentales, se requiere de mucha práctica, concentración y esfuerzo. Para ser un buen administrador de bases de datos primero hay que trabajar el RDBMS a nivel usuario, para que con esto se practique la manipulación de la información y sus datos a través de las consultas SQL, después de esto podemos pasar a administrar una base de datos, pero como lo mencioné anteriormente para ser un buen administrador de bases de datos se requiere de tiempo y práctica.

He aprendido que un buen administrador de bases de datos no puede estancarse con los conocimientos que se le enseñaron en un principio, sino que tiene que actualizarse y conocer las tendencias para estar al corriente de lo más nuevo en el mundo de los RDBMS's y de los sistemas de seguridad para los mismos. El estar actualizados es una tarea que no debe tener fin.

Finalmente quiero dejar claro que el diplomado forma a profesionales altamente capacitados sin olvidar la parte humana y nos enseña a trabajar en equipo para lograr el éxito de la empresa u organización en donde se labore y claro también tener éxito en el aspecto personal.



GLOSARIO

- **Abstracción** Permite omitir las propiedades y acciones de un objeto y dejar sólo aquellas que nos interesan para una situación particular.
- **Atributo** Las entidades se componen de atributos que son cada una de las propiedades o características que tienen las entidades. Cada ejemplar de una misma entidad posee los mismos atributos, tanto en nombre como en número, diferenciándose cada uno de los ejemplares por los valores que toman cada uno de dichos atributos.
- **Clase** Es una descripción de un grupo de objetos con propiedades comunes (atributos), comportamiento común (operaciones) y relaciones comunes con otros objetos. Una clase sirve como plantilla para crear objetos. Una clase también es una definición abstracta de un objeto.
- **Clave** El modelo Entidad - Relación exige que cada entidad tenga un identificador, se trata de un atributo o conjunto de atributos que identifican de forma única a cada uno de los ejemplares de la entidad.
- **Dominio** Se define dominio como un conjunto de valores que puede tomar un determinado atributo dentro de una entidad.
- **Encapsulamiento** En general, hace referencia a cualquier tipo de ocultamiento. El encapsulamiento es una propiedad que permite asegurar que el contenido de la información de un objeto está oculto al mundo exterior, y por tanto protege los datos.
- **Entidad** Se puede definir como entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o que puede llegar a existir y del cual deseamos guardar información, por ejemplo, un profesor, un alumno o bien una materia. Las entidades las podemos clasificar en entidades débiles y entidades regulares.
- **Entidad Débil** Son aquellas entidades en las que su existencia depende de la existencia de ejemplares en otras entidades.
- **Entidad Regular** Son aquellas entidades que existen por sí mismas, es decir, la existencia de un ejemplar de una entidad no depende de la existencia de otros ejemplares en otra entidad.
- **Firma y Protocolo** El nombre, tipos de parámetros y tipos de resultados de una operación se denomina firma (Signature). Al conjunto de firmas de todas las operaciones de un objeto se denomina protocolo.



- **Herencia** La clase en general es llamada superclase y la especialización de ésta es llamada subclase; es decir que las clases hijas conservan la estructura y comportamiento de la clase padre.
- **Interrelación** Se entiende por interrelación a la asociación, vinculación o correspondencia entre entidades. Al igual que las entidades, las interrelaciones se pueden clasificar en regulares y débiles, esto de acuerdo al tipo de entidad que estén asociando, entidades regulares o entidades débiles, con otra de cualquier tipo.
- **Jerarquía** Un sistema se compone por subsistemas (más pequeños) relacionados que tienen a su vez sus propios subsistemas, y así sucesivamente.
- **Mensaje** El mensaje posee un emisor, un receptor y una acción. Es la petición de un servicio, para que los objetos de un sistema trabajen en conjunto, un objeto (cliente) envía a otro una llamada para realizar una operación y el objeto receptor (proveedor) ejecutará la operación.
- **Objeto** Un objeto es la representación de una entidad, ya sea física, conceptual o de software. Un objeto representa un elemento identificable con ciertas características (atributos) y que pueden realizar un conjunto de acciones (operaciones). Un objeto debe de tener estado, comportamiento e identidad. Un objeto es también una instancia de una clase.
- **Persistencia** Es la conservación de la información en algún medio de almacenamiento con la posibilidad de recuperarla en cualquier momento. La persistencia permite que los objetos existan fuera del programa en ejecución.
- **RDBMS** Por sus siglas en inglés (Relational Database Management System), Sistema Manejador de Bases de Datos Relacional.