

Universidad Nacional Autónoma de México  
Facultad de Estudios Superiores Aragón



## “PROPUESTA VÍA WEB DE LA ADMINISTRACIÓN DE BASES DE DATOS”

Cursos y seminarios de actualización y capacitación profesional

**Autor:** César Núñez Hernández.

**Asesor:** Ing. Silvia Vega Muytoy.

México D.F., 2009



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.

**INTRODUCCIÓN ..... 3**

**1 INFORME GENERAL DEL DIPLOMADO ..... 4**

**1 INFORME GENERAL DEL DIPLOMADO ..... 4**

**1.1- SISTEMAS DE INFORMACIÓN Y EL MODELO DE DATOS RELACIONAL..... 4**

1.1.1 CONCEPTOS GENERALES DE BASES DE DATOS ..... 4

1.1.2 CARACTERÍSTICAS DE LAS BASES DE DATOS ..... 5

1.1.3 MODELO RELACIONAL ..... 5

1.1.4 ESTRUCTURA DEL MODELO RELACIONAL ..... 6

1.1.5 DOMINIO Y ATRIBUTO ..... 6

1.1.6 BASES DE DATOS RELACIONALES ..... 7

1.1.7 NORMALIZACIÓN ..... 8

**1.2- SISTEMAS MANEJADORES DE BASES DE DATOS RELACIONALES (RDBMS)..... 9**

1.2.1 RDBMS..... 9

1.2.2 ESQUEMAS DE SEGURIDAD EN EL RDBMS ..... 10

1.2.3 SQL ANSI ..... 10

1.2.4 REGLAS DE CODD ..... 12

1.2.5 SISTEMAS MANEJADORES DE BASES DE DATOS COMERCIALES Y LIBRES..... 13

1.2.6 TIPOS DE DATOS USADOS POR LOS RDBMS..... 15

**1.3- SQL (STRUCTURE QUERY LANGUAGE) ..... 17**

1.3.1 SQL Y ANSI..... 17

1.3.2 DEFINICIÓN DE DATOS. .... 17

1.3.3 ESTRUCTURAS DE CONTROL DE FLUJO. .... 19

**1.4- ACCESO A DATOS A TRAVÉS DE LA PROGRAMACIÓN DE CLIENTES ..... 20**

1.4.1. PHP ..... 20

1.4.2 ENVÍO DE PARÁMETROS ..... 21

1.4.3 RECEPCIÓN DE PARÁMETROS ..... 21

1.4.4 CONEXIÓN A LAS BASES DE DATOS ..... 22

**1.5- FUNDAMENTOS DE SISTEMAS OPERATIVOS..... 23**

1.5.1 SISTEMA OPERATIVO LINUX..... 23

1.5.2 INTERPRETE DE COMANDOS ..... 23

1.5.3 RESPONSABILIDADES DEL ADMINISTRADOR ..... 24

1.5.4 ADMINISTRACIÓN DE CUENTAS DE USUARIO Y GRUPOS ..... 25

**1.6- HABILIDADES DIRECTIVAS PARA ADMINISTRADORES ..... 29**

1.6.1 HABILIDADES DIRECTIVAS ..... 29

1.6.2 LÍDER..... 29

1.6.3 COMUNICACIÓN ..... 30

1.6.4 LENGUAJE CORPORAL ..... 31

1.6.5 LOS TRES TIPOS DE PERSONAS..... 32

1.6.6 REUNIONES EFECTIVAS ..... 33

**1.7- ADMINISTRACIÓN DE BASES DE DATOS ..... 33**

1.7.1 CONCEPTOS GENERALES. .... 33

1.7.2 ADMINISTRAR LA ESTRUCTURA DE LA BASE DE DATOS..... 34

1.7.3 ADMINISTRAR EL SISTEMA MANEJADOR DE BASE DE DATOS. ....	34
1.7.4 ESTABLECER EL DICCIONARIO DE DATOS. ....	35
1.7.5 ASEGURAR LA CONFIABILIDAD DE LA BASE DE DATOS .....	35
1.7.6 SEGURIDAD DE LA BASE DE DATOS.....	35
1.7.7 OBJETIVOS DEL ADMINISTRADOR DE LA BASE DE DATOS. ....	35
1.7.8 FUNCIONES BÁSICAS DEL ADMINISTRADOR DE BASES DE DATOS. ....	36
<b>1.8 BUENAS PRÁCTICAS EN LA FUNCIÓN DE LA ADMINISTRACIÓN .....</b>	<b>38</b>
1.8.1 AUDITORÍA INFORMÁTICA .....	38
1.8.2 FUNCIONES DEL ADMINISTRADOR DE BASES DE DATOS .....	40
1.8.3 ISACA.....	41
1.8.4 CONTROL OBJECTIVES FOR INFORMATION AND RELATED TECHNOLOGY (COBIT) .....	41
1.8.5 MEJORES PRÁCTICAS EN LAS BASES DE DATOS, CONTROLES DE BASES DE DATOS .....	44
1.8.6 INFRAESTRUCTURA DE TECNOLOGÍAS DE INFORMACIÓN, <i>FRECUENTEMENTE ABREVIADA ITIL</i> .....	44
<b>1.9- SEGURIDAD EN BASES DE DATOS .....</b>	<b>45</b>
1.9.1 SEGURIDAD .....	45
<b>1.10- PERFORMANCE AND TUNING.....</b>	<b>48</b>
1.10.1 PERFORMANCE AND TUNING .....	48
<b>1.11- MODELO ORIENTADO A OBJETOS .....</b>	<b>48</b>
1.11.1 UML (LENGUAJE UNIFICADO DE MODELADO) .....	49
<b>1.12- TÓPICOS AVANZADOS DE BASES DE DATOS .....</b>	<b>49</b>
1.12.1 DATAWAREHOUSE .....	49
1.12.2 LOS ELEMENTOS BÁSICOS DE UN DATA WAREHOUSE .....	49
1.12.3 LOS PROCESOS BÁSICOS DEL DATA WAREHOUSE.....	50
 <b><u>2. PROPUESTA VÍA WEB DE LA ADMINISTRACIÓN DE BASES DE DATOS</u></b> <b>..... 52</b>	
 2.1 CARACTERÍSTICAS GENERALES DEL PROYECTO .....	52
2.2 PROPUESTA DE UN PROYECTO .....	54
 <b><u>CONCLUSIONES .....</u></b>	<b>57</b>
 <b><u>BIBLIOGRAFÍA .....</u></b>	<b>58</b>

## **INTRODUCCIÓN**

Una base de datos es un sistema para archivar información en computadora cuyo propósito general es mantener información y hacer que esté disponible cuando se solicite.

Las bases de datos son áreas de la computación que ha recibido mucha atención, debido a sus múltiples aplicaciones: bibliotecas, automatización de oficinas, ingeniería de software, diccionarios automatizados y en general cualquier programa orientado a mantener y recuperar información. Su recuperación, actualización y manejo es relativamente simple con el uso de cualquier manejador de bases de datos.

Cuando se habla de información se refiere a documentos cuya estructura es declarada explícitamente de algún modo, con etiquetas o elementos que mediante la sintaxis dan un sentido a la información con la que escribimos sentencias, como se hace en los lenguajes de programación.

Este trabajo consta de dos capítulos; el capítulo uno contiene una breve descripción de los visto en los doce módulos que consta el Diplomado de Administración de Bases de Datos y en el capítulo dos se describe el proyecto final del diplomado y se realiza una propuesta de un proyecto para la administración de la información.

## 1 INFORME GENERAL DEL DIPLOMADO

El contenido del siguiente informe es acerca de los 12 módulos que conforma el diplomado de Administración de Bases de Datos, así como una breve descripción del proyecto realizado durante el transcurso del diplomado aplicando lo visto durante las 240 horas.

### 1.1- Sistemas de Información y el Modelo de Datos Relacional

**Objetivo:** Se identificarán todos los elementos que conforman una base de datos y la diferencia del manejo de archivos de dato; Será posible conceptualizar hechos del mundo real como un modelo de datos, utilizando la teoría relacional para su representación.

#### 1.1.1 Conceptos Generales de bases de datos

En la tabla 1.1.1 se presentan los conceptos generales de bases de datos.

<b>Dato</b>	Es la unidad mínima de información, un hecho aislado que no tiene ningún significado o la representación simbólica (numérica, alfabética, etc.), de un atributo o característica de una entidad.
<b>Información</b>	Es un acontecimiento o una serie de acontecimientos, que llevan un mensaje y que al ser percibida por el receptor mediante alguno de sus sentidos, amplía sus conocimientos, en esta relación sólo el destinatario puede evaluar el significado y utilidad de la información recibida.
<b>Bases de Datos</b>	Colección integrada, estructurada y generalizada de datos, atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios.

**Tabla 1.1.1 conceptos generales de bases de datos**

### 1.1.2 Características de las bases de datos

Las bases de datos deben contener las características que se muestran en la tabla 1.1.2:

<b>Redundancia:</b> La redundancia de datos se refiere a la existencia de información repetida en diferentes tablas dentro de una base de datos. Dentro de una base de datos la redundancia debe ser mínima y controlada.
<b>Consistencia:</b> Los problemas de consistencia de los datos se deben a la redundancia de éstos, ya que al hacer modificaciones es probable que haya incongruencias al no modificar todos los datos que se repiten.
<b>Integridad:</b> Para que los datos sean íntegros deben ser consistentes y válidos. Si existe inconsistencia entre 2 datos que representan lo mismo, es un ejemplo de falta de integridad dado por la existencia de redundancia en los datos. Para lograr la integridad de la base de datos se debe mantener la redundancia al mínimo y controlada, establecer llaves o índices primarios y hacer validaciones entre las tablas relacionadas y al insertar los datos.
<b>Seguridad:</b> La seguridad implica asegurar que los usuarios están autorizados para llevar a cabo lo que tratan de hacer. Se refiere principalmente al control de acceso, modificación o definición de los datos o la base de datos. Algunos sistemas operativos proporcionan cierto nivel de seguridad pero principalmente debe radicar en el SGBD, esto también para evitar la dependencia de entidades externas.

**Tabla 1.1.2 Características de las bases de datos**

### 1.1.3 Modelo Relacional

El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Su idea fundamental es el uso de “relaciones”. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados “tuplas”. Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, esto es, pensando en cada relación como si fuese una tabla que está compuestas por *registros* (cada fila de la tabla sería un registro o *tupla*), y *columnas* (también llamadas *campos*).

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene mayor relevancia (a diferencia de otros modelos como el jerárquico y el de red). La información puede ser recuperada o almacenada por medio de “consultas” que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un

conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Para manipular la información se utiliza un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional y el Cálculo relacional. *El Álgebra relacional* permite describir la forma de realizar una consulta, en cambio, *el Cálculo relacional* sólo indica lo que se desea devolver.

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

### 1.1.4 Estructura del Modelo Relacional

La relación es el elemento básico del modelo relacional y se representa por una tabla. En la tabla 1.1.3 se listan algunos términos usados y sus equivalencias:

Relación	Tabla
Tupla	Fila
Atributo	Columna
Número de Tuplas	Cardinalidad
Número de atributos	Grado
Dominio	Colección de valores
Clave Primaria	Identificador único para la tabla

**Tabla 1.1.3 términos usados y sus equivalencias en las bases de datos**

### 1.1.5 Dominio y Atributo

**Dominio:** Un Dominio es un conjunto de valores del mismo tipo e indivisibles, caracterizados por un nombre y un tipo de datos. Por ejemplo:

Dominio de Nacionalidades: chilena, francesa, norteamericana

**Atributo:** Un atributo describe a las entidades y constituye una parte específica de información para un dominio.

Una tabla no puede tener dos atributos con el mismo nombre, sin embargo, existen los dominios compuestos los cuales están formados por otros dominios, esto permite tener un nombre diferente y aplicar restricciones específicas para cada dominio.

**Relación:** Matemáticamente una relación definida sobre los  $n$  dominios  $D_1, D_2, \dots, D_n$ , es un subconjunto del producto cartesiano de estos dominios donde cada elemento de la relación (tupla), es una serie de  $n$  valores ordenados.

**Claves:** En una relación, una clave es un conjunto de atributos que identifican de manera única y mínima a cada tupla. Dentro de una tabla existen claves candidatas que se clasifican en tres grupos:

**1. Clave primaria:** Es la clave que permite identificar las tuplas de la relación de forma única.

**2. Clave alternativas:** Son aquellas claves que no han sido escogidas como claves primarias, pero que también podrían identificar de manera única a una tupla.

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

**Restricciones:** Las restricciones es aquello que no esta permitido ya sea por el propio modelo (inherentes), como no tener tuplas repetidas, o por el usuario, como la validación de los datos que son ingresados.

### 1.1.6 Bases de datos relacionales

Una base de datos relacional es una base de datos en donde todos los datos que el usuario puede ver están organizados como tablas de valores, y en donde todas las operaciones de la base de datos operan sobre estas tablas.

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

- Cada "tabla" contiene sólo un tipo de registros.
- Los campos no tienen un orden específico, de izquierda a derecha.
- Los registros no tienen un orden específico, de arriba hacia abajo.
- Cada campo tiene un solo valor.

#### 1.1.6.1 Diseño de las bases de datos relacionales

El primer paso para crear una base de datos, es planificar el tipo de información que se quiere almacenar, teniendo en cuenta cual es la información disponible y cual la que se necesita. Planificar la base de datos, en especial de las tablas, es lo que va a permitir manejar correctamente la información.

En el diseño de la estructura de una tabla se describen los campos que componen el registro y los datos que contendrán esos campos. Se deben determinar los campos necesarios y definirlos correctamente con un nombre, especificando su tipo y su longitud.

### 1.1.7 Normalización

Para poder llevar las entidades y sus relaciones a un formato de tabla, es necesario hacer el proceso de normalización que consiste en acomodar los datos en un grupo de tablas que representan a las entidades, además de sus características y sus relaciones de forma adecuada, esto para asegurar que el modelo conceptual de la base de datos funcionará correctamente.

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 1.1.1 se muestra las formas de normalización de una base de datos

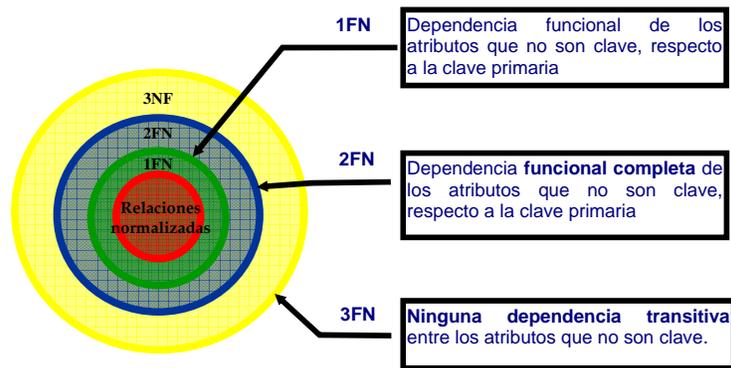


Figura 1.1.1 normalización de una base de datos

Las ventajas de la normalización son las siguientes:

- Evita anomalías al insertar, modificar o borrar.
- Mejora la independencia de datos.
- No pone restricciones no naturales en la estructura de los datos.
- Están encaminadas a eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas.

En la tabla 1.1.4 se describen las 3 formas de normalización de una base de datos.

<b>Primera Forma Normal (1FN):</b>	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.
<b>Segunda Forma Normal (2FN)</b>	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
<b>Tercera Forma Normal (3FN)</b>	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.

Tabla 1.1.4 descripción de la normalización de las bases de datos

## 1.2- Sistemas Manejadores de Bases de Datos Relacionales (RDBMS)

**Objetivo:** 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.

### 1.2.1 RDBMS

La interfaz entre la base de datos física y los usuarios del sistema se denomina, manejador de bases de datos (MBD) o Sistema Manejador de Bases de Datos Relacional RDBMS (Relational Data Base Management System).

Un RDBMS es el conjunto de programas que permiten la definición, manipulación y controlan el acceso de 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 con redundancia mínima.
- 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 del almacenamiento físico de los datos ya que lo importante para estos usuarios es la información contenida.
- Los datos son compartidos por diferentes usuarios y programas de aplicación; existe un mecanismo común para inserción, actualización, eliminación y consulta de los datos.
- Cuando sea necesario actualizar o hacer una recuperación de los datos, los métodos utilizados serán capaces de conservar la integridad, seguridad y confidencialidad de los mismos.
- Tanto datos como procedimientos pueden ser transportables a través de diferentes SGBD.

Conceptualmente, cuando un usuario realiza alguna petición, en un RDBMS se presenta lo siguiente:

- El usuario solicita la petición a la base de datos utilizando algún sublenguaje de datos como SQL.
- El RDBMS interpreta esa solicitud y la analiza.
- El RDBMS verifica que correspondan las partes externa/conceptual, el esquema conceptual, la parte conceptual/interna y la estructura de almacenamiento.

- El DBMS realiza las operaciones que se requieran sobre la base de datos y regresa una respuesta al usuario.

Gráficamente, las operaciones requeridas por un RDBMS se representa en la figura 1.2.1:

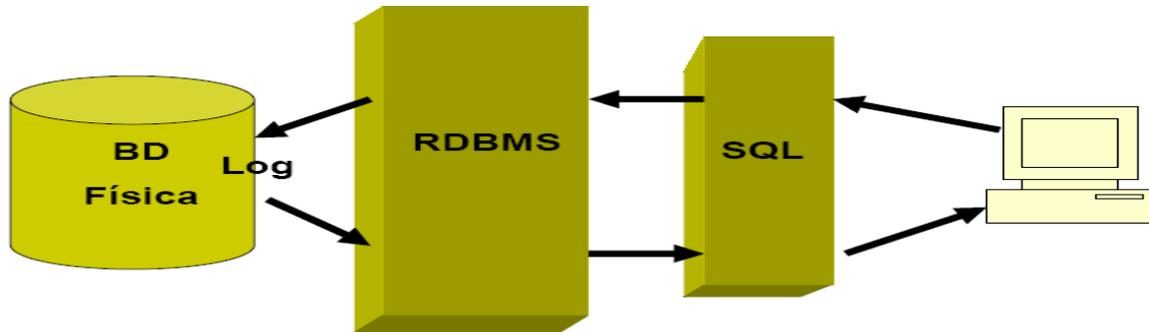


Figura 1.2.1 operaciones requeridas por un RDBMS

### 1.2.2 Esquemas de seguridad en el RDBMS

Para poder acceder dentro del Sistema Manejador de Base de Datos (DBMS) es necesario pasar varias capas de seguridad como en la figura 1.2.2:

- **Seguridad a Nivel Servidor:** El usuario debe tener una cuenta válida en la capa del servidor (DBMS).
- **Seguridad a Nivel de Base de Datos:** El usuario debe ser un usuario válido dentro de la capa de la base de datos.
- **Seguridad a Nivel de Permisos sobre Objetos y Comandos:** El usuario deberá tener permiso dentro de la capa de los datos.



Figura 1.2.2 capas de seguridad de un RDBMS

### 1.2.3 SQL ANSI

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 SQL99, sin embargo internamente son convertidos a un tipo de dato que si 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 1.2.1 se muestra en la primera columna, el tipo de dato como se especifica en el ANSI SQL y en las demás columnas se indica el tipo de datos equivalente que cumple con dicho estándar.

Tipo en SQL99	MySQL	PostgreSQL	Oracle	Sybase	Ms SQL Server	Descripción
tinyint	tinyint			tinyint	tinyint	
smallint	smallint	smallint	smallint (lo convierte a number)	smallint	smallint	Entero con signo de 2 bytes
int, integer	int, integer	integer	int (lo convierte a number)	int	int	Entero con signo de 4 bytes
float	float		carácter()	float	float	Número de punto flotante
double	double	Double precision	Double precision (lo convierte a carácter)	double precision	Double precision (se convierte en carácter)	Número Doble
real		Real	real (Lo convierte a carácter)	real	real	Número Real
Numeric (p,d)	Numeric (p,d)	Numeric (p,d)	number(p,d)	numeric(p,d)	decimal(p,d)	Numérico con precisión p y d decimales
character varying (n)	varchar(n)	varchar(n)	varchar2(n)	varchar(n)	varchar(n)	Carácter de longitud variable
char, carácter (n)	char(n)	char(n)	char(n)	char(n)	char(n)	Cadena de caracteres de longitud fija
date	date	date				Fecha sin hora del día
time	time	time				Hora del día
timestamp p	timestamp	timestamp	date	datetime	datetime	Fecha y hora del día
blob	blob	bytea	blob	image	image	Binary large object
clob	text	text	clob	text	text	Character large object

**Tabla 1.2.1 tipos de datos estándar del SQL**

En algunos casos existe en el manejador el nombre del tipo de dato estándar, más no cumple con las especificaciones que debería. Por ejemplo, como se muestra en la tabla 1.2.1, el tipo Date existe en Oracle, más no cumple con las especificaciones del Date estándar, porque almacena hora y fecha en dicho campo y no por separado. Lo mismo sucede para el tipo de datos Timestamp de Sybase y Ms SQL Server, aunque existe la palabra como tal, timestamp no tiene ninguna relación con el estándar.

### 1.2.4 Reglas de Codd

Las **12 reglas de Codd** son un sistema de reglas propuestas por Edgar F. Codd, del modelo relacional para las bases de datos, diseñado para definir qué requiere un sistema de administración de base de datos.

Codd se percató de que existían bases de datos en el mercado las cuales decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar estas tablas literalmente normalizadas; entonces éste publicó 12 reglas que un verdadero sistema relacional debería tener aunque en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse "más relacional" cuanto más siga las reglas que se muestran en la tabla 1.2.2.

<b>Regla de la información</b>	Toda la información en la base de datos es representada unidireccionalmente, por valores en posiciones de las columnas dentro de filas de tablas.
<b>Regla del acceso garantizado</b>	Todos los datos deben ser accesibles sin ambigüedad.
<b>Tratamiento sistemático de valores nulos</b>	El sistema de gestión de base de datos debe permitir que haya campos nulos.
<b>Catálogo dinámico en línea basado en el modelo relacional</b>	El sistema debe soportar un catálogo en línea, el catálogo relacional deber ser accesible a los usuarios autorizados. Es decir, los usuarios deben poder tener acceso a la estructura de la base de datos (catálogo).
<b>Regla del sublenguaje de datos completo</b>	<p>El sistema debe soportar por lo menos un lenguaje relacional que;</p> <ol style="list-style-type: none"> <li>1. Tenga una sintaxis lineal.</li> <li>2. Puede ser utilizado recíprocamente y dentro de programas de uso.</li> </ol> <p>Soporte operaciones de definición de datos, operaciones de manipulación de datos (actualización así como la recuperación), seguridad e integridad y operaciones de administración de transacciones.</p>
<b>Regla de actualización de vistas</b>	Todas las vistas que son teóricamente actualizables deben ser actualizables por el sistema.
<b>inserción, actualización y borrado de alto nivel</b>	El sistema debe soportar suministrar datos en el mismo tiempo que se inserte, actualiza o este borrando. Esto significa que los datos se pueden recuperar de una base de datos relacional en los sistemas contruidos de datos de filas múltiples y/o de tablas múltiples.
<b>Independencia física de datos</b>	Los cambios en el nivel físico (cómo se almacenan los datos, si en arreglos o en las listas encadenadas, etc.) no debe requerir un cambio a una solicitud basada en la estructura.
<b>Independencia lógica de datos</b>	Los cambios al nivel lógico (tablas, columnas, filas, etcétera) no

	deben requerir un cambio a una solicitud basada en la estructura. La independencia de datos lógica es más difícil de lograr que la independencia física de datos.
<b>Independencia de integridad</b>	Las limitaciones de la integridad se deben especificar por separado de los programas de la aplicación y se almacenan en la base de datos. Debe ser posible cambiar esas limitaciones sin afectar innecesariamente las aplicaciones existentes.
<b>Independencia de distribución</b>	La distribución de las porciones de la base de datos a las varias localizaciones debe ser invisible a los usuarios de la base de datos. Los usos existentes deben continuar funcionando con éxito: <ol style="list-style-type: none"> <li>1. cuando una versión distribuida del SGBD se introdujo por primera vez</li> </ol> cuando se distribuyen los datos existentes se redistribuyen en todo el sistema.
<b>Regla de la no subversión</b>	La regla de la no subversión, si el sistema proporciona una interfaz de bajo nivel (de registro a la vez) y luego de que esa interfaz no se pueda utilizar para subvertir el sistema, por ejemplo: sin pasar por seguridad relacional o limitación de integridad.

**Tabla 1.2.2 Las 12 reglas de Codd**

### 1.2.5 Sistemas Manejadores de Bases de Datos comerciales y libres

El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica. En la tabla 1.2.3 se enlista los RDBMS más comunes con sus características.

RDBMS	Características	Sistema Operativo	Procesador	RAM	Disco Duro
Microsoft SQL Server	Soporte de transacciones. Escalabilidad, estabilidad y seguridad. Soporta procedimientos almacenados. Permite trabajar en modo cliente-servidor	Windows 2000 Windows XP Windows 2003 y 2008	Pentium IV	128 Mb	280 Mb
Sybase	Administrador lógico de recursos y tareas Múltiples esquemas de bloqueo de datos. Copias de respaldo en línea y de alto rendimiento. Soporte a múltiples herramientas de desarrollo y lenguajes de programación, como PowerBuilder, Visual Basic, Java, C, PHP, etc. Soporte a múltiples protocolos de conectividad, como Open Client (propio de Sybase), ODBC, OLE DB, ADO.NET y JDBC.	HP-UX AIX IRIX Linux Windows	Pentium III	128 Mb	240 Mb
PostgreS QL	Llaves foráneas o Claves Foráneas ( <i>foreign keys</i> )	Unix Linux	Pentium III	128 Mb	140 Mb

	Disparadores ( <i>triggers</i> ) Vistas. Integridad transaccional. Herencia de tablas. Tipos de datos y operaciones geométricas.	Windows			
MySQL	Un amplio subconjunto de ANSI SQL 99, y varias extensiones. Soporte a multiplataforma Procedimientos almacenados Triggers Cursores Vistas actualizables Soporte completo para Unicode	Linux Windows FreeBSD UNIX IBM-AIX MAC OSX	Procesador de 200 MHz	128 Mb	200 Mb
Oracle	Apoya en el diseño y optimización de modelos de datos. Asistir a los desarrolladores con sus conocimientos de SQL y de construcción de procedimientos almacenados y <i>triggers</i> , entre otros. Apoya en la definición de estándares de diseño y nomenclatura de objetos. Documentar y mantener un registro periódico de las mantenciones, actualizaciones de hardware y software, cambios en las aplicaciones y, en general, todos aquellos eventos relacionados con cambios en el entorno de utilización de una base de datos.	Solaris HP-UX AIX Linux Unix Windows NT/2000/XP	Pentium IV	1 Gb	400 Mb
DB2 IBM	Permite el manejo de objetos grandes La definición de datos y funciones por parte del usuario Chequeo de integridad referencial SQL recursivo soporte multimedia: texto, imágenes, video, audio; backup/recuperación on-line y offline.	IBM AIX Sun Solaris HP-UX Linux Windows NT/200/XP/2003	Pentium III	512 Mb	120 Mb

**Tabla 1.2.3 RDBMS más comunes**

### 1.2.6 Tipos de datos usados por los RDBMS

Los tipos de datos SQL se clasifican en 13 tipos de datos primarios y de varios sinónimos válidos reconocidos por dichos tipos de datos dependiendo del RDBMS. En la tabla 1.2.4 se enlistan los tipos de datos más comunes en los RDBMS

Tipo de Dato	Sinónimos	Tamaño	Descripción
Binary	Varbinary Binary Varying Bit varying	1 byte por caracter	Se puede almacenar cualquier tipo de datos en un campo de este tipo. Los datos no se traducen (por ejemplo, a texto). La forma en que se introducen los datos en un campo binario indica cómo aparecerán al mostrarlos.
Bit	Boolean Logical Yes no	1 byte	Valores Sí y No, y campos que contienen solamente uno de dos valores.
Byte	Integer Tinyint	1 byte	Un número entero entre 0 y 255.
Counter	Autoincrement		Se utiliza para campos contadores cuyo valor se incrementa automáticamente al crear un nuevo registro.
Money	Currency	8 bytes	Un número entero comprendido entre -922.337.203.685.477,5808 y 922.337.203.685.477,5807.
Datetime	Date Time	8 bytes	Una valor de fecha u hora entre los años 100 y 9999
Decimal	Numeric DEC	17 bytes	Un tipo de datos numérico exacto con valores comprendidos entre $10^{28} - 1$ y $-10^{28} - 1$ . Puede definir la precisión (1 -28) y la escala (0 -precisión definida). La precisión y la escala predeterminadas son 18 y 0, respectivamente.
Real	Single Float IEESINGLE	4 bytes	Un valor de coma flotante de precisión simple con un intervalo comprendido entre $-3,402823E38$ y $-1,401298E-45$ para valores negativos, y desde $1,401298E-45$ a $3,402823E38$ para valores positivos, y 0.
FLOAT	Double Float number	8 bytes	Un valor de coma flotante de precisión doble con un intervalo comprendido entre $-1,79769313486232E308$ y $-4,94065645841247E-324$ para valores negativos, y desde $4,94065645841247E-324$ a $1,79769313486232E308$ para valores positivos, y

			0.
Smallint	Short integer	2 bytes	Un entero corto entre -32.768 y 32.767.
Integer	Long Integer 4	4 bytes	Un entero largo entre -2.147.483.648 y 2.147.483.647.
Image	Longbinary General	Lo que se requiera	Desde cero hasta un máximo de 2.14 gigabytes.
Text	Longtext Longchar Memo Note ntext	2 bytes por carácter	Desde cero hasta un máximo de 2.14 gigabytes.
Char	TEXT(n) ALPHANUMERIC CHARACTER STRING VARCHAR CHARACTER VARYINGNCHAR NATIONAL CHARACTERNATIONAL CHARNATIONAL CHARACTER VARYINGNATIONAL CHAR VARYING	2 bytespor carácter. (Consulte las notas).	Desde cero a 255 caracteres.

**Tabla 1.2.4 tipos de datos más comunes en los RDBMS**

### 1.3- SQL (Structure Query Language)

**Objetivo:** El participante utilizará el lenguaje SQL para extraer la información necesaria en el desarrollo de las aplicaciones en su entorno de trabajo a través de la elaboración de las consultas que se puedan requerir y de la programación de los procedimientos a utilizar; así mismo, entenderá la ventaja de conocer las normas ANSI SQL vistas anteriormente y la sintaxis de las principales instrucciones del SQL dentro de la migración de consultas y procedimientos de un RDBMS a otro.

#### 1.3.1 SQL y ANSI

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.

Actualmente la versión soportada por la mayoría de las bases de datos es el ANSI SQL99 también conocido como SQL3.

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.

SQL es un lenguaje fácil de entender, ya que su estructura utiliza palabras en inglés, lo que lo hace fácil de aprender y utilizar y las instrucciones se enfocan a qué buscar, dejando al RDBMS la tarea de cómo recuperar la información.

#### 1.3.2 Definición de datos.

**DDL o Lenguaje de Definición de Datos:** Se utiliza para crear, eliminar o modificar tablas, índices, vistas, triggers, procedimientos; es decir, 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 editar la información de la base de datos, esto implica: seleccionar, insertar, borrar, modificar.

Las instrucciones que se manejan son:

- 1.- **select.** Permite realizar consultas a la base de datos.
- 2.- **insert.** Se utiliza para agregar registros a una tabla.
- 3.- **update.** Sirve para modificar los valores de los campos de una tabla.
- 4.- **delete.** Utilizado para eliminar los valores de los campos de una tabla.

**DCL o Lenguaje de Control de Datos:** Se utiliza para definir los permisos de seguridad para acceder y editar a los elementos de la base de datos, es decir, permitir o revocar el acceso.

Las instrucciones relacionadas son:

**grant.** Permite otorgar permisos a los usuarios sobre los objetos de la base de datos y las operaciones sobre ellos.

**revoke.** Permite quitar permisos sobre los objetos de la base de datos y las operaciones sobre los mismos.

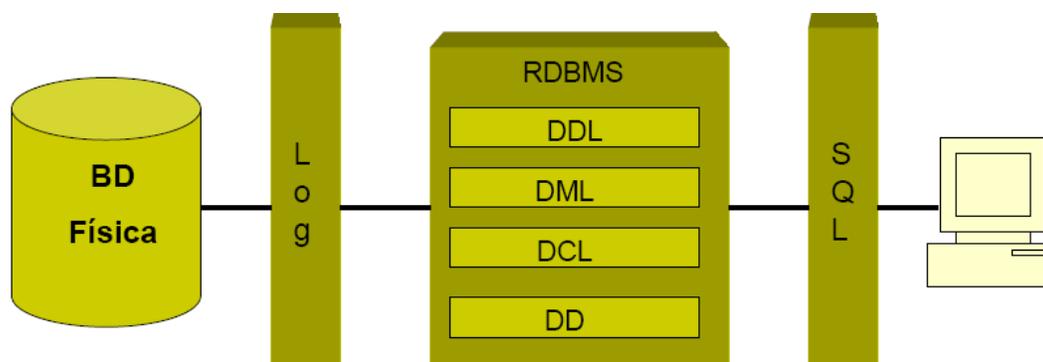
**DD o Diccionario de Datos:** Lo que contiene el diccionario son “datos acerca de los datos”, llamados comúnmente meta datos, el diccionario de datos dice como están definidos los objetos de la base de datos. Todos los esquemas de la base de datos (externo, conceptual e interno), se almacenan físicamente en el diccionario.

El diccionario puede formar parte de la base de datos que describe, y por tanto, incluir su propia descripción.

Sus principales funciones son las siguientes:

- Describe todos los elementos del sistema.
- Indica los mismos significados a todos los elementos del sistema.
- Es la documentación de las características del sistema.
- Permite analizar los detalles para determinar cómo deben realizarse los cambios.
- Encuentra errores y omisiones del sistema.

En la figura 1.3.1 se muestra un esquema de como se estructura la definición de datos



**Figura 1.3.1 Definición de datos**

### 1.3.3 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.

1. IF-THEN-END IF
2. IF-THEN-ELSE-END IF
3. FOR LOOP
4. WHILE LOOP
5. ETIQUETAS

### 1.3.4 Procedimientos Almacenados

Un procedimiento almacenado es un conjunto de comandos de SQL que pueden ser compilados y almacenados en el servidor. Una vez realizado esto, los clientes no necesitan volver a teclear todas las instrucciones sino únicamente hacer referencia al procedimiento. Esto mejora el rendimiento del servidor, ya que la instrucción de SQL solamente es revisada una sola vez y menos información debe ser enviada entre el cliente y el servidor. En la tabla 1.3.1 se muestra que RDBMS manejan procedimientos almacenados.

	MySQL	PostgreSQL	Oracle	Sybase	Ms SQL Server
Soporte para Procedimientos Almacenados	No	Si	Si	Si	Si

**Tabla 1.3.1 RDBMS manejan procedimientos almacenados.**

El lenguaje que se emplea para programar los procedimientos almacenados, varía de un RDBMS a otro, y existen algunos que permiten programar en más de un lenguaje.

## 1.4- Acceso a Datos a través de la Programación de Clientes

### Objetivo.

El participante utilizará distintos elementos que brindan de los lenguajes de programación estudiados, para la consulta y manipulación de información en las bases de datos.

#### 1.4.1. PHP

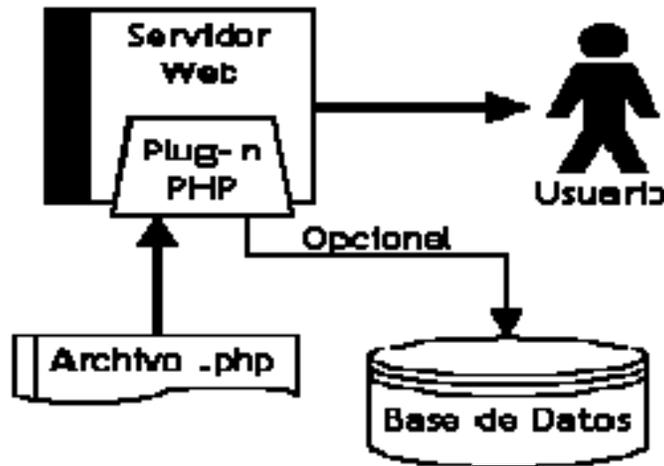
PHP (Hypertext Preprocessor) es un lenguaje de programación que permite la generación dinámica de contenidos en un servidor Web.

PHP permite integrar fragmentos de código dentro de una página HTML y realizar operaciones de una forma fácil y eficaz sin tener que hacer programas en un lenguaje distinto al HTML como podría ser C, Java, PERL, etc. En la tabla 1.4.1 se muestran las características y ventajas del lenguaje PHP

Características	Ventajas
Es un potente y robusto lenguaje de programación integrado en documentos HTML.	Código fuente abierto: permite hacer modificaciones en su código, escrito en lenguaje C.
Dispone de librerías de conexión con la mayoría de los manejadores de base de datos.	Gratuito
Tiene soporte para los diferentes protocolos de comunicación en Internet (HTTP, IMAP, FTP, LDAP, SNMP, etc.).	Portable y con versiones para múltiples plataformas (Windows 95, 98, NT, 2000, Unix, Linux, etc.).
	Eficiente: consume muy pocos recursos en el servidor
	Alta velocidad de desarrollo: Tiene librerías muy útiles y documentadas que ahorran trabajo al programador.

**Tabla 1.4.1 Características y ventajas del lenguaje PHP**

PHP es una extensión para servidores Web. Lo que hace es ponerse "entre" el servidor y el cliente. En la figura 1.4.1 se muestra de manera general el funcionamiento del lenguaje PHP.



**Figura 1.4.1 funcionamiento de PHP**

PHP toma código dentro de las páginas, lo ejecuta en el servidor y envía el resultado al cliente. El cliente no puede visualizar el código del programa, sólo su resultado. Además, por ser un lenguaje de scripting, los programas no se compilan, sino sólo se interpretan; esto significa que es más lento en ejecutarse que, por ejemplo, un programa en C, pero al mismo tiempo los cambios en el código PHP tienen efecto de inmediato.

No todos los archivos son interpretados por el plug-in de PHP, sino sólo aquellos que hayan sido definidos en la configuración del servidor como tales, puesto que la interpretación de código es un proceso que toma un tiempo mayor al que se emplea en enviar una página tal como está. Es común utilizar una o varias de las siguientes extensiones para informar al servidor que debe interpretarlas: .php, .phtml, .php3 y .php4.

#### 1.4.2 Envío de parámetros

Para poder enviar la información contenida en los formularios se utilizan dos métodos:

- Método GET: Sirve para solicitar y enviar información que se visualiza en la URL, es decir, cualquiera puede ver el contenido de las variables que se envía.
- Método POST: En este método las variables no se envían por el URL, sino que van dentro del mensaje, es decir, las variables van ocultas para el usuario.

#### 1.4.3 Recepción de parámetros

PHP guarda las variables que se envían a través de estos métodos en arreglos. Se tiene acceso a las variables ocupando el nombre como clave del arreglo:

```
$_GET["var1"]
```

```
$_POST["var1"]
```

### 1.4.4 Conexión a las Bases de datos

Una de las principales ventajas de trabajar con páginas dinámicas es poder almacenar su contenido en bases de datos para poder organizar, actualizar y buscarlo más fácilmente.

Función	Sintaxis
<b>mysql_connect</b>	int <b>mysql_connect</b> ( [string [hostname], string[username], string[password])  Abre una conexión con un servidor Mysql
<b>mysql_close</b>	int <b>mysql_close</b> ( [int link_identifier]) Cierra una conexión con Mysql, devuelve true si se cerro correctamente, falso en caso de error.
<b>mysql_query</b>	int <b>mysql_query</b> ( string query , int [link_identifier])  Envía una consulta a Mysql. Devuelve true o false, para indicar que las operaciones se han ejecutado con éxito, y para la sentencia SELECT, un identificador del resultado.
<b>mysql_fetch_row</b>	array <b>mysql_fetch_row</b> ( int result_id ), Devuelve un tabla con los valores de los campos de la fila actual de la consulta y obtiene los registros como un array enumerado, comenzando en cero.
<b>mysql_fetch_array</b>	array <b>mysql_fetch_array</b> (int result); está es una versión extendida de mysql_fetch_row, y regresa los datos en un arreglo asociativo.
<b>mysql_free_result</b>	int <b>mysql_free_result</b> (int result); libera la memoria utilizada en la consulta.

**Tabla 1.4.2 funciones SQL para conectarse a través de PHP**

En la tabla 1.4.2 se muestran las funciones más comunes de SQL para conectarse a las bases de datos a través de PHP.

## **1.5- Fundamentos de Sistemas Operativos**

### **Objetivo:**

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 el RDBMS

### **1.5.1 Sistema operativo Linux**

Una de las partes que integran una computadora es el hardware que, junto con el software, trabajan para realizar las diferentes tareas.

El hardware de la computadora está formada de varios componentes diferentes, como el CPU, memoria, los dispositivos de entrada y salida y el disco duro, cada uno con un propósito específico.

Para que estos componentes trabajen juntos es necesario que sean administrados por un sistema operativo.

El sistema operativo es un conjunto de programas que administran los recursos del sistema.

El sistema operativo interpreta las instrucciones del usuario o de las aplicaciones y dice a la computadora qué hacer. Maneja las entradas y las salidas, guarda el seguimiento de los datos almacenados en el disco y se comunica con los periféricos como el monitor, disco duro, impresoras, módems, etcétera.

Los tres principales componentes del sistema operativo Linux son:

- El kernel: es el núcleo del sistema operativo. Es el programa maestro que administra todos los recursos de la computadora.
- El Shell: es una interfaz entre el usuario y el kernel. Su función es interpretar las instrucciones o comandos que recibe y después ejecutarlas.
- El árbol de directorios: se refiere al conjunto de directorios donde se almacena toda la información del sistema operativo.

### **1.5.2 Interprete de comandos**

Parte fundamental de un sistema operativo encargada de ejecutar las órdenes básicas para el manejo del sistema. También se denomina shell. Suelen incorporar características tales como control de procesos, redirección de entrada/salida y un lenguaje de órdenes para escribir programas por lotes o (scripts).

Los principales intérpretes de comandos son:

- command.com: perteneciente a los sistemas MS-DOS y derivados como Windows 95 y 98.

- ksh: El korn shell de UNIX. C Shell (csh): similar en sintáxis al lenguaje C.
- Bourne Shell
- GNU Bourne Again Shell (bash): variante del Bourne Shell clásico para Linux con características avanzadas como control de procesos, historial de ordenes y edición similar a Emacs entre otras.
- Tcsh: versión de C Shell para Linux con funcionalidades similares a bash.
- Zsh: pequeño intérprete similar al Bourne Shell.

En la figura 1.5.1 se muestra los elementos que se involucran para el funcionamiento de la línea de comandos



### 1.5.1 Esquema de elementos involucrados en la línea de comandos

### 1.5.3 Responsabilidades del Administrador

La siguiente lista muestra algunas de las tareas de las que es responsable el administrador del sistema:

- Administración de cuentas de usuario y grupos
- Mantenimiento de la seguridad del sistema
- Configurar dispositivos
- Instalar y particionar unidades de disco
- Manejar y administrar sistemas de archivos
- Calendarizar tareas relacionadas al sistema

- Configurar archivos de inicialización del sistema
- Instalación del sistema operativo
- Administrar paquetes de software y parches
- Realizar tareas de respaldo y recuperación
- Administración de la recuperación del sistema
- Etcétera.

## 1.5.4 Administración de cuentas de usuario y grupos

### 1.5.4.1 Cuentas de usuario

Una importante tarea del administrador del sistema es configurar las cuentas de los usuarios para que tengan acceso al sistema. Cada cuenta de usuario se conforma de 5 componentes principales:

- Nombre de usuario
- Contraseña
- Directorio HOME de usuario
- Shell
- Archivos de inicialización del usuario

Cada usuario debe tener una cuenta de usuario para poder ingresar al sistema.

### 1.5.4.2 La cuenta Root

La cuenta y contraseña del usuario root son configuradas durante la instalación. Esta cuenta es usada por los administradores del sistema para realizar tareas administrativas.

La información de las cuentas de los usuarios se almacena en los siguientes archivos:

- /etc/passwd: Contiene los usuarios autorizados para ingresar al sistema
- /etc/shadow: Contiene las contraseñas cifradas de los usuarios.
- /etc/group: Define los grupos que hay en el sistema.

En la tabla 1.5.1 se muestran las Cuentas por default en un S.O tipo linux

<b>Nombre login</b>	<b>ID usuario</b>	<b>Descripción</b>
Root	0	Cuenta de súper usuario, no tiene restricciones y sobrescribe los login, protecciones y permisos. Este usuario tiene acceso a todo el sistema.
Bin	1	Cuenta administrativa que es dueña de la mayoría de los comandos
daemon	2	Cuenta del sistema que controla los procesos en segundo plano
Adm	3	Cuenta administrativa que es dueña de distintos archivos de sistema

**Tabla 1.5.1 cuentas por default de un sistema operativo tipo Linux**

### 1.5.4.3 Creación de usuarios

Para agregar cuentas de usuario se utiliza el comando `useradd`, el cual crea los registros en el `/etc/passwd` y en `/etc/shadow`.

La sintaxis para agregar un usuario es la siguiente:

```
useradd [ -u uid ][ -g gid ][ -G gid [,gid,.. ]][ -d dir ][ -m ][ -s shell ][ -c comment ]  
nombre de usuario
```

Opciones:

- u id : Configura el id único para cada usuario
- g grupo: especifica un gid o nombre de grupo predefinido
- G grupo: define los grupos secundarios para el usuario
- d directorio: define la ruta hacia el directorio del usuario
- m: crea el directorio en el caso que no exista
- s shell: Define la ruta al shell que el usuario tendrá predefinido
- c comentario: Usado para poner el nombre del usuario
- e expiración: configura la fecha de expiración de la cuenta en el formato (mm/dd/yy)
- f inactivo: configura el número de días inactivos permitidos para una cuenta de usuario

### 1.5.4.4 Creación de grupos

Para crear grupos es necesario tener la cuenta de root y utilizar el comando `groupadd`. Este comando agrega nuevos grupos en el archivo `/etc/group`. La sintaxis para crear grupos es la siguiente:

```
groupadd [ -g gid [ -o ] ] nombregrupo
```

Opciones:

- g gid : Asigna el id de grupo para el nuevo grupo
- o : Permite que el gid se duplique

Ejemplo:

```
groupadd -g 301 grupo1
```

### 1.5.4.5 Administración de Software

Todo el software en un sistema Linux está dividido en paquetes RPM los cuales pueden ser instalados, actualizados o eliminados.

El Administrador de paquetes (RPM) es un sistema de empaquetado que trabaja en Linux y otros sistemas como UNIX y que está a la disposición de cualquiera, ya que es libre.

#### 1.5.4.6 RPM's

RPM facilita las actualizaciones de sistema para el usuario y mantiene una base de datos de los paquetes instalados puede hacer consultas y verificaciones poderosas en su sistema. Si prefiere una interfaz gráfica, puede utilizar Herramienta de administración de paquetes para ejecutar muchos comandos RPM.

RPM tiene cinco modos de operación básicos (sin contar la construcción de paquetes): instalación, desinstalación, actualización, consulta y verificación.  
Instalación de RPM

Los paquetes RPM normalmente tienen nombres de archivo como foo-1.0-1.i386.rpm. El nombre de archivo incluye el nombre de paquete (foo), versión (1.0), lanzamiento (1) y arquitectura (i386). La instalación de un paquete es tan simple como teclear el siguiente comando en el intérprete de comandos de shell:

```
rpm -ivh foo-1.0-1.i386.rpm
```

#### 1.5.5 Windows 2003 server

Es un sistema operativo de la familia Windows de la marca Microsoft para servidores que salió al mercado en el año 2003. Está basada en tecnología NT y su versión del núcleo NT es la 5.2.

En términos generales, Windows Server 2003 se podría considerar como un Windows XP modificado, no con menos funciones, sino que estas están deshabilitadas por defecto para obtener un mejor rendimiento y para centrar el uso de procesador en las características de servidor, por ejemplo, la interfaz gráfica denominada *Luna* de Windows XP viene desactivada y viene con la interfaz clásica de Windows. Sin embargo, es posible volver a activar las características mediante comandos *services.msc*. En internet existen varios trucos para hacerlo semejante a Windows XP.

##### 1.5.5.1 Características de Windows 2003 Server

Sus características más importantes del sistema operativo Windows 2003 Server son:

- Sistema de archivos NTFS:
  1. cuotas
  2. cifrado y compresión de archivos, carpetas y no unidades completas.
  3. permite montar dispositivos de almacenamiento sobre sistemas de archivos de otros dispositivos al estilo unix
- Gestión de almacenamiento, backups. Incluye gestión jerárquica del almacenamiento, consiste en utilizar un algoritmo de caché para pasar

los datos menos usados de discos duros a medios ópticos o similares más lentos, y volverlos a leer a disco duro cuando se necesitan.

- Windows Driver Model: Implementación básica de los dispositivos más utilizados, de esa manera los fabricantes de dispositivos sólo han de programar ciertas especificaciones de su hardware.
- ActiveDirectory Directorio de organización basado en LDAP, permite gestionar de forma centralizada la seguridad de una red corporativa a nivel local.
- Autenticación Kerberos5
- DNS con registro de IP's dinámicamente
- Políticas de seguridad

Con estas características que tiene Windows 2003, se puede tener la seguridad que al administrar las funciones se tendrían protegidos los datos que contenga dicho sistema.

#### **1.5.5.2 Servidores**

Los servidores que maneja Windows 2003 son:

- Servidor de archivos
- Servidor de impresiones
- Servidor de aplicaciones
- Servidor de correo (SMTP/POP)
- Servidor de terminal
- Servidor de Redes privadas virtuales (VPN) (o acceso remoto al servidor)
- Controlador de Dominios (mediante Active Directory)
- Servidor DNS
- Servidor DHCP
- Servidor de Streaming de Video
- Servidor WINS

Lo anterior son los principales servidores que maneja Windows 2003.

## **1.6- Habilidades Directivas para Administradores**

### **Objetivo**

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.

### **1.6.1 Habilidades directivas**

Las personas dedicadas a diversos aspectos relacionados con la computación deben tener claro que su trabajo forma parte de un sistema que integra recursos técnicos, humanos y materiales destinados a lograr objetivos comunes.

Administrar es alcanzar un objetivo mediante el esfuerzo humano coordinado, de ahí la necesidad de desarrollar la habilidad para comunicarse con las personas con las que se colabora, comprender las necesidades del grupo en lo individual y lo general, descubrir las posibilidades técnicas y humanas para obtener lo mejor de cada persona y el mejor funcionamiento grupal.

### **1.6.2 Líder**

Inspira confianza en la gente: Cambiar la actitud tradicional de intervenir para dar instrucciones y corregir a otra persona. La persona debe ser capaz de inspirar una visión común, modelar con el ejemplo, reconocer y recompensar todos los esfuerzos, son las únicas vías para ganar la confianza de la gente que lleva a cargo y por tanto, reforzar consistentemente el cambio. Las características principales de un líder son:

- Centrarse en las personas: Olvidar la tendencia de considerar a las personas como cosas o simplemente como recursos humanos y responder a las necesidades que cada individuo tiene como ser humano.
- Desarrollar un equipo: Derribar las barreras entre funciones y personas, trabajar de manera interdisciplinaria, aumentar la participación y ser capaz de utilizar y fomentar los mejores recursos del grupo, son habilidades indispensables de un líder empeñado en mejorar la calidad y productividad.
- Generar un clima motivador: Los procesos de calidad y el buen servicio se basan en la participación de la gente y estimular su iniciativa para que actúen con responsabilidad sin necesidad de vigilancia.

Entonces, se puede señalar que liderazgo es el proceso de influir, guiar o dirigir a los miembros del grupo hacia el éxito en la consecución de metas y objetivos organizacionales.

### 1.6.3 Comunicación

Comunicarse es hacer conocer a otras personas nuestros conocimientos o pensamientos.

Para que exista comunicación debe existir un emisor, un receptor, un mensaje y un canal a través del cual se enviará el mensaje. Si alguno de estos elementos falla, se dice que se ha producido una interferencia y no podrá establecerse la comunicación. En la figura 1.6.1 se muestra el contexto para una buena comunicación.

Para enviar un mensaje se debe utilizar un código común al emisor y al receptor. El código puede ser:

**Lingüístico:** es el código más completo, ya que la lengua oral o escrita puede ser comprendida por un número mayor de personas que otros tipos de códigos.

**No lingüístico:** los gestos, sonidos y señales constituyen este tipo de código, el cual es menos preciso que el anterior. De acuerdo a la intención del emisor, los mensajes pueden clasificarse en: expresivos, informativos y apelativos.



#### 1.6.1 contexto para una buena comunicación.

Antes de comenzar con la comunicación se debe tener en cuenta lo siguiente:

- Apariencia
- Voz
- Volumen
- Tono
- Entonación
- Pronunciación
- Conocer a la audiencia

### 1.6.4 Lenguaje corporal

El lenguaje corporal, que no es más que todo lo que tú transmites por medio de movimientos o gestos, delata completamente tus sentimientos o percepción acerca de la persona con la que está interactuando.

Cuando se conversa con una o varias personas, reflejas y envías miles de señales y mensajes a través del comportamiento. Así que hay que prestar atención y sacarle provecho a los siguientes datos, porque tanto en la vida laboral como en la personal, serán de gran provecho. En la tabla 1.6.1 se muestran las señales reflejadas a través del comportamiento.

ACTO	LO QUE REFLEJA
Acariciarse la quijada	Toma de decisiones
Entrelazar los dedos	Autoridad
Dar un tirón al oído	Inseguridad
Mirar hacia abajo	No creer en lo que se escucha
Frotarse las manos	Impaciencia
Apretarse la nariz	Evaluación negativa
Golpear ligeramente los dedos	Impaciencia
Sentarse con las manos agarrando la cabeza por detrás	Seguridad en sí mismo y superioridad
Inclinar la cabeza	Interés
Palma de la mano abierta	Sinceridad, franqueza e inocencia
Caminar erguido	Confianza y seguridad en sí mismo
Pararse con las manos en las caderas	Buena disposición para hacer algo
Jugar con el cabello	Falta de confianza en sí mismo e inseguridad
Comerse las uñas	Inseguridad o nervios
La cabeza descansando sobre las manos o mirar hacia el piso	Aburrimiento
Unir los tobillos	Aprensión
Manos agarradas hacia la espalda	Furia, ira, frustración y aprensión
Cruzar las piernas, balanceando ligeramente el pie	Aburrimiento
Brazos cruzados a la altura del pecho	Actitud a la defensiva

Caminar con las manos en los bolsillos o con los hombros encorvados	Abatimiento
Manos en las mejillas	Evaluación
Frotarse un ojo	Dudas
Tocarse ligeramente la nariz	Mentir, dudar o rechazar algo

**Tabla 1.6.1 Señales que se reflejan a través del comportamiento**

Cuando se conoce a una persona, se ha de intentar conseguir que se forme una primera buena impresión de nosotros. En los primeros minutos de conversación la persona se crea una idea de nosotros, que luego es muy difícil de cambiar.

Por estos motivos se recomienda:

- Ir bien vestido
- Ir aseado
- Presentarse siempre con una sonrisa
- Mirar a la persona a los ojos
- Dar un apretón de manos adecuado
- Mostrar interés por la persona
- Ser puntual
- Tratar bien a la recepcionista. El tono en que nos anuncie puede influir en la primera impresión.

### 1.6.5 Los tres tipos de personas

Antes de pasar a analizar distintos aspectos del lenguaje corporal, hay 3 tipos de personas, según la forma en que tienen de pensar. Debido a su forma de pensar, adoptarán un tipo de lenguaje verbal y corporal.

Los tres tipos de personas son:

1. **Cinestésicas o Kinestésicas**. Piensan a través de sentimientos. Usan lenguaje relacionado con los sentimientos: siento que no está bien, tengo la impresión de que esto está bien. Son sensibles al tacto y al olfato. Son personas que hacen las cosas despacio. Son personas que se acercan mucho a otras al hablar.
2. **Visuales**. Piensan a través de imágenes. Usan lenguaje relacionado con la vista: ya veo lo que quiere decir, no lo tengo claro, podríamos enfocar el tema. Son personas que hacen las cosas rápido, ya que las imágenes son rápidas.

No se acercan mucho al interlocutor (necesitan ver todos los movimientos).

3. **Auditivas**. Piensan a través de sonidos. Usan expresiones del tipo: me suena extraño, ya he escuchado lo que ha dicho. Hablan pausados, dan mucha información, y son reflexivos. Se acercan más al interlocutor que los visuales.

En realidad, todos somos a la vez visuales, auditivos y kinestésicos. Aunque predominará una o dos de las formas de pensar. Y sobre todo en momentos de tensión, se usará la forma de pensar que predomine. Lo ideal es que cuando hay comunicación con alguien se debe adaptar el lenguaje, velocidad y expresiones al tipo de persona que es nuestro interlocutor, así nuestro lenguaje llegará más claro. Por ejemplo formular una pregunta adaptada a la personalidad del otro: ¿Ves lo que digo? (visual), ¿Cómo te suena lo que digo? (auditivo), ¿Cómo te sientes en relación a esto? (kinestésico)

### 1.6.6 Reuniones Efectivas

Administrar el tiempo

- 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.

Teniendo una buena administración del tiempo es posible dirigir mucho mejor al grupo de trabajo que se tiene a cargo.

## 1.7- Administración de Bases de Datos

### Objetivo

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.

#### 1.7.1 Conceptos Generales.

**Administrador de la Base de Datos.** Es la persona encargada de definir y controlar las bases de datos, además proporciona asesoría a los desarrolladores, usuarios y personas que lo requieran. Es la persona o equipo de personas profesionales responsables del control y manejo del sistema de base de datos, generalmente tiene(n) experiencia en DBMS, diseño de bases de datos, Sistemas operativos, comunicación de datos, hardware y programación.

La característica más importante que debe poseer es un conocimiento profundo de las políticas y normas de la empresa, así como el criterio de la empresa para aplicarlas en un momento dado. La responsabilidad general del DBA es facilitar el desarrollo y el uso de la Base de Datos dentro de las guías de acción definidas por la administración de los datos.

El Administrador de Bases de Datos es responsable primordialmente de:

- Administrar la estructura de la Base de Datos.
- Administrar la actividad de los datos.
- Administrar el Sistema Manejador de Base de Datos.
- Establecer el Diccionario de Datos.
- Asegurar la confiabilidad de la Base de Datos.
- Confirmar la seguridad de la Base de Datos.

Estas responsabilidades hacen que el administrador de bases de datos lleve a cabo la administración correcta del sistema con una reducción de fallas en éste.

### **1.7.2 Administrar la estructura de la Base de Datos.**

Esta responsabilidad incluye participar en el diseño inicial de la base de datos y su puesta en práctica así como controlar, y administrar sus requerimientos, ayudando a evaluar alternativas, incluyendo los DBMS a utilizar y ayudando en el diseño general de la bases de datos.

Una vez diseñada las bases de datos, es puesta en práctica utilizando productos del DBMS, procediéndose entonces a la creación de los datos (captura inicial). El DBA participa en el desarrollo de procedimientos y controles para asegurar la calidad y la alta integridad de la BD.

### **1.7.3 Administrar el Sistema Manejador de Base de Datos.**

Existe una gran actividad al interior de un DBMS. La concurrencia de múltiples usuarios requiere la estandarización de los procesos de operación; el DBA es responsable de estas especificaciones y de asegurarse que éstas lleguen a quienes concierne. Todo el ámbito de la base de datos se rige por estándares, desde la forma de como se captura la información (tipo de dato, longitud, formato), como es procesada y presentada. El nivel de estandarización alcanza hasta los aspectos más internos de la base de datos; como se accesa a un archivo, como se determinan los índices primarios y auxiliares, registros, etc.

El DBA debe procurar siempre que los estándares que serán aplicados beneficien también a los usuarios, privilegiando siempre la optimización en la operación del DBMS y el apego de las políticas de la empresa. Entre las funciones del DBA se encuentra la de revisar los estándares periódicamente para determinar su operatividad, ajustarlos, ampliarlos o cancelarlos y hacer que éstos se cumplan.

#### **1.7.4 Establecer el Diccionario de Datos.**

Cuando se definen estándares sobre la estructura de la base de datos, se deben de registrar en una sección del diccionario de datos a la que todos aquellos usuarios relacionados con ese tipo de proceso pueden acceder. Este metadato debe precisar información que indique con claridad el tipo de datos que serán utilizados, sus ámbitos de influencia y sus limitantes de seguridad.

#### **1.7.5 Asegurar la Confiabilidad de la Base de Datos**

Se trata de realizar un sistema de bases de datos lo suficientemente robusto para que sea capaz de recuperarse frente a errores o usos inadecuados. Se deben utilizar gestores con las herramientas necesarias para la reparación de los posibles errores que las bases de datos pueden sufrir, por ejemplo tras un corte inesperado de luz.

#### **1.7.6 Seguridad de la Base de Datos.**

Entre las características mas utilizadas por el DBA para tratar de resolver o minimizar la seguridad de las bases de datos se encuentran las siguientes:

- Restringir el acceso a los procedimientos para ciertos usuarios.
- Restringir al acceso a los datos para ciertos usuarios procedimientos y/o datos.
- Evitar la coincidencia de horarios para usuarios que comparten.

Las técnicas de recuperación son otra función esencial del DBA al administrar la actividad de datos. A pesar de que el DBMS lleva a cabo una parte del proceso de recuperación, los usuarios determinan en forma critica la operatividad de esos sistemas de protección. El DBA debe anticipar fallas y definir procedimientos estándares de operación; los usuarios deben saber que hacer cuando el sistema este caído y que es lo primero que debe realizarse cuando el sistema este puesto en marcha nuevamente. El personal de operación deberá saber como iniciar el proceso de recuperación de la BD que copias de seguridad utilizar; como programar la reejecución del tiempo perdido y de las tareas pendientes; es importante también establecer un calendario para llevar a cabo estas actividades sin afectar a otros sistemas dentro de la organización que hagan uso de los mismos recursos de computo.

#### **1.7.7 Objetivos del Administrador de la Base de Datos.**

1. **Mantener la Integridad de los Datos.** Una base de datos debe protegerse de accidentes tales como los errores en la entrada de los datos o en la programación, del uso mal intencionado de la base de datos y de los fallos del hardware o del software que corrompen los datos. La protección contra accidentes, que ocasiona inexactitudes en los datos, es parte del objetivo de garantizar la integridad de los datos. Estos accidentes incluyen los fallos durante el procesamiento de las transacciones, los errores lógicos que

infringen la suposición de que las transacciones preservan las restricciones de consistencia de la base de datos y las anomalías debido al acceso concurrente en la base de datos (acceso concurrente). La integridad, se encarga de asegurar que las operaciones ejecutadas por los usuarios sean correctas y mantengan la consistencia de la base de datos.

2. **Mantener la Seguridad de los Datos.** La protección de la base de datos de usos mal intencionados o no autorizados se denomina seguridad de los datos. La seguridad se encarga de limitar a los usuarios a ejecutar únicamente las operaciones permitidas.
  
3. **Mantener la Disponibilidad de los Datos.** La posibilidad de fallos de hardware o de software requiere procedimientos de recuperación de la base de datos. Tiene que proporcionar medios para el restablecimiento de las bases de datos que se hayan corrompido por desperfectos del sistema, a un estado uniforme.

### **1.7.8 Funciones Básicas del Administrador de Bases de Datos.**

#### **1.7.8.1 Creación de Bases de Datos y Tablas.**

Creando Bases de Datos:

- Localización de las bases de datos.
- Tipo de base de datos (modo de direccionamiento).

Creando Tablas:

- Seleccionando tipos de datos.
- Tablas fragmentadas o no fragmentadas.
- Localización de la tabla.
- Determinación del espacio en disco.
- Modo de aseguramiento de candados.

Al tener en claro los datos y tipos de datos se crea de la base de datos y sus tablas para tener así un mejor control de información.

#### **1.7.8.2 Especificación de las Restricciones de Integridad de los Datos.**

Estos son algunos métodos para asegurar la integridad de los datos:

- Privilegios:
  - Base de datos.

- Tabla.
- Columna
- Integridad de identidad, semántica y referencial.
- Vistas.

Las restricciones de integridad se mantienen en una estructura especial del sistema que consulta el gestor de la base de datos cada vez que se tiene lugar una actualización en el sistema

#### **1.7.8.3 Administrar la Concurrencia.**

Existen dos tipos de control de la concurrencia:

- Concurrencia de Lectura: (Instrucción SELECT)
- Administrada a través de los niveles de aislamiento.
- Concurrencia de Actualización: Instrucciones INSERT, DELETE y UPDATE.

La administración de la concurrencia involucra como los datos son consultados y actualizados en un ambiente multiusuario.

#### **1.7.8.4 Optimización del Acceso a Datos.**

- Índices.
- Estadísticas de actualización.
- Distribución de datos.

#### **1.7.8.5 Definir el Esquema Conceptual.**

Es tarea del administrador de datos decidir con exactitud cual es la información que debe mantenerse en la base de datos, una vez identificado los datos a almacenar en un nivel abstracto, el dba debe crear a continuación el esquema conceptual correspondiente, empleando el DDL conceptual.

#### **1.7.8.6 Definir el Esquema Interno.**

El dba debe definir la representación de la información en la base de datos almacenada (diseño físico). Debe crear la definición de estructura de almacenamiento correspondiente (esquema interno) con el DDL interno y definir la correspondencia entre los esquemas interno y conceptual.

#### **1.7.8.7 Vincularse con los Usuarios.**

El dba debe encargarse de la comunicación con los usuarios, garantizar la disponibilidad de los datos que requieren y escribir y/o ayudar a los usuarios a escribir los esquemas externos necesarios, empleando el DDL externo aplicable.

#### **1.7.8.8 Procedimientos de Respaldo y Recuperación.**

El dba debe definir un plan de recuperación adecuado que incluya descarga o vaciado periódico de la base de datos en un medio de almacenamiento de respaldo, y procedimientos para cargar otra vez la base de datos a partir del vaciado más reciente cuando sea necesario.

### 1.7.9 Tipos de usuarios

- DBA, están permitidas todas las operaciones, conceder privilegios y establecer usuarios. Usuario con derecho a crear, borrar y modificar objetos y que además puede conceder privilegios a otros usuarios sobre los objetos que ha creado. Privilegios sobre los objetos, añadir nuevos campos, indexar, alterar la estructura de los objetos, etc.
- Programadores de aplicaciones. Los profesionales en computación que interactúan con el sistema por medio de llamadas en DML (Lenguaje de Manipulación de Datos), las cuales están incorporadas en un programa escrito en un lenguaje de programación (Por ejemplo, Cobol, PL/I, Pascal, C, etc.).
- Usuarios sofisticados. Los usuarios sofisticados interactúan con el sistema sin escribir programas. En cambio escriben sus preguntas en un lenguaje de consultas de base de datos.
- Usuarios especializados. Algunos usuarios sofisticados escriben aplicaciones de base de datos especializadas que no encajan en el marco tradicional de procesamiento de datos.
- Usuarios ingenuos. Los usuarios no sofisticados interactúan con el sistema invocando a uno de los programas de aplicación permanentes que se han escrito anteriormente en el sistema de base de datos, podemos mencionar al usuario ingenuo como el usuario final que utiliza el sistema de base de datos sin saber nada del diseño interno del mismo por ejemplo: un cajero.

En pocas palabras se puede definir a los usuarios como toda persona que tenga todo tipo de contacto con el sistema de base de datos desde que éste se diseña, elabora, termina y se usa.

## 1.8 Buenas Prácticas en la Función de la Administración

### Objetivo

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.

### 1.8.1 Auditoría informática

La auditoría informática se define como un examen crítico que se realiza con el fin de evaluar la eficacia y eficiencia de una sección, un organismo, una entidad, etcétera.

Los principales objetivos que constituyen a la auditoría Informática son el control de la función informática, el análisis de la eficiencia de los Sistemas Informáticos que comporta, la verificación del cumplimiento de la Normativa general de la empresa en este ámbito y la revisión de la eficaz gestión de los recursos materiales y humanos informáticos.

El auditor informático ha de velar por la correcta utilización de los amplios recursos que la empresa pone en juego para disponer de un eficiente y eficaz Sistema de Información. Claro está, que para la realización de una auditoría informática eficaz, se debe entender a la empresa en su más amplio sentido, ya que una Universidad, un Ministerio o un Hospital son tan empresas como una Sociedad Anónima o empresa Pública. Todos utilizan la informática para gestionar sus "negocios" de forma rápida y eficiente con el fin de obtener beneficios económicos y de costos.

La auditoría interna es la realizada con recursos materiales y personas que pertenecen a la empresa auditada. Los empleados que realizan esta tarea son remunerados económicamente. La auditoría interna existe por expresa decisión de la Empresa, o sea, que puede optar por su disolución en cualquier momento.

Por otro lado, la auditoría externa es realizada por personas afines a la empresa auditada; es siempre remunerada. Se presupone una mayor objetividad que en la Auditoría Interna, debido al mayor distanciamiento entre auditores y auditados.

La auditoría informática interna cuenta con algunas ventajas adicionales muy importantes respecto de la auditoría externa, las cuales no son tan perceptibles como en las auditorías convencionales. La auditoría interna tiene la ventaja de que puede actuar periódicamente realizando Revisiones globales, como parte de su Plan Anual y de su actividad normal. Los auditados conocen estos planes y se habitúan a las auditorías, especialmente cuando las consecuencias de las Recomendaciones habidas benefician su trabajo.

En una empresa, los responsables de Informática escuchan, orientan e informan sobre las posibilidades técnicas y los costos de tal Sistema. Con voz, pero a menudo sin voto, Informática trata de satisfacer lo más adecuadamente posible aquellas necesidades. La empresa necesita controlar su Informática y ésta necesita que su propia gestión esté sometida a los mismos Procedimientos y estándares que el resto de aquella. La conjunción de ambas necesidades cristaliza en la figura del auditor interno informático.

En cuanto a empresas se refiere, solamente las más grandes pueden poseer una Auditoría propia y permanente, mientras que el resto acuden a las auditorías externas. Puede ser que algún profesional informático sea trasladado desde su puesto de trabajo a la Auditoría Interna de la empresa cuando ésta existe. Finalmente, la propia Informática requiere de su propio grupo de Control Interno, con implantación física en su estructura, puesto que si se ubicase dentro de la estructura Informática ya no sería independiente. Hoy, ya existen varias organizaciones Informáticas dentro de la misma empresa, y con diverso grado de autonomía, que son coordinadas por órganos corporativos de Sistemas de Información de las Empresas.

Una Empresa o Institución que posee auditoría interna puede y debe en ocasiones contratar servicios de auditoría externa. Las razones para hacerlo suelen ser:

- Necesidad de auditar una materia de gran especialización, para la cual los servicios propios no están suficientemente capacitados.
- Contrastar algún Informe interno con el que resulte del externo, en aquellos supuestos de emisión interna de graves recomendaciones que chocan con la opinión generalizada de la propia empresa.
- Servir como mecanismo protector de posibles auditorías informáticas externas decretadas por la misma empresa.
- Aunque la auditoría interna sea independiente del Departamento de Sistemas, sigue siendo la misma empresa, por lo tanto, es necesario que se le realicen auditorías externas como para tener una visión desde afuera de la empresa.

La auditoría informática, tanto externa como interna, debe ser una actividad exenta de cualquier contenido o matiz "político" ajeno a la propia estrategia y política general de la empresa. La función auditora puede actuar de oficio, por iniciativa del propio órgano, o a instancias de parte, esto es, por encargo de la dirección o cliente.



### 1.8.1 Proceso de auditoría informática

#### 1.8.2 Funciones del administrador de bases de datos

De acuerdo al manual de revisión de los Certified Information Systems Auditor (CISA), Certificado para auditores de sistemas de información, las responsabilidades del administrador de bases de datos son:

1. Custodia información de la organización.
2. Define y mantiene la estructura de los datos en el sistema corporativo de BD.
3. Debe comprender a la empresa, datos de usuario y las relaciones de éstos.
4. Responsable de la seguridad y clasificación de la información de los datos compartidos, almacenados en los sistemas de BD.

5. Responsable del diseño real, definición y mantenimiento de las BD corporativas.
6. Especificar la definición física de los datos y cambiarla para su mejor desempeño.
7. Seleccionar e implementar herramientas de optimización de la BD.
8. Probar y evaluar las herramientas de programadores.
9. Dar soporte técnico a programadores sobre estructura de la BD.
10. Implementar controles de definición, acceso, actualización y concurrencia.
11. Monitorear el uso, recopilar estadísticas de desempeño y ajustar la BD.
12. Definir e iniciar los procedimientos de respaldo y recuperación.

Pueden usarse diversos mecanismos para el reforzamiento de la segregación de funciones:

- Acceso a datos
- Autorización de transacciones
- Custodia de activos
- Formas de autorización
- Tablas o registros de autorización de usuarios

Teniendo estos mecanismos para la base de datos se puede tener más controlado el acceso y manejo de información.

### **1.8.3 ISACA**

Existe una asociación internacional denominada Information Systems Audit and Control Association, Auditoría de sistemas de información y asociaciones de control (ISACA por sus siglas en inglés); comenzó en 1967 con un grupo pequeño de profesionales con actividades similares –verificar controles en los sistemas de cómputo que se estaban convirtiendo en parte crítica en las operaciones de sus organizaciones. En 1969 se conformaron como la asociación de auditores EDP con el fin de cubrir las necesidades únicas, diversas y de alta tecnología en el naciente campo de la TI.

La misión de ISACA consiste en mejorar el reconocimiento de la profesión de auditoría y control de las TI a través de la elaboración de materiales y marcos de trabajo, así como capacitación y certificación de sus miembros a través de la fundación (Information Systems Audit and Control Foundation)

### **1.8.4 Control Objectives for Information and Related Technology (COBIT)**

Un marco de trabajo conocido como Objetivos de Control para la Información y la Tecnología Relacionada (Control Objectives for Information and related Technology-COBIT®) sirve como guía para la buena práctica de la auditoría de las TI, emitido por el IT Governance Institute. COBIT es un marco de control o sistema de control interno que tiene el objetivo de que las TI's sean exitosas en satisfacer los requerimientos de la organización.

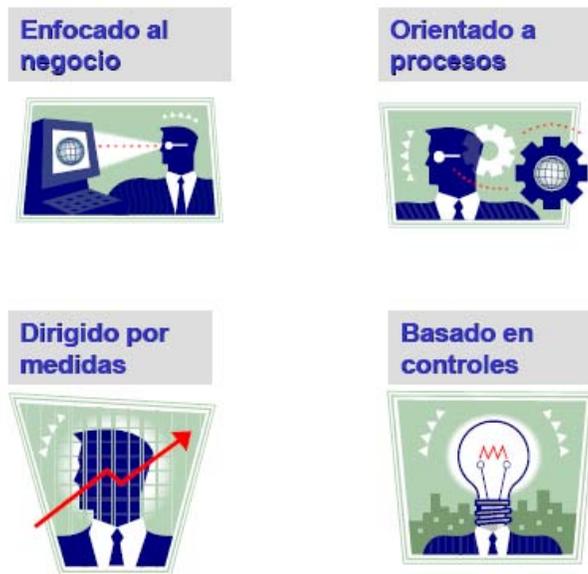
COBIT es un marco de control generalmente aplicable y aceptado internacionalmente como buena práctica para controles de TI.

Provee una serie de buenas prácticas a través de un marco de dominio y procesos y presenta actividades en una estructura lógica y manejable.

Las buenas prácticas de Cobit:

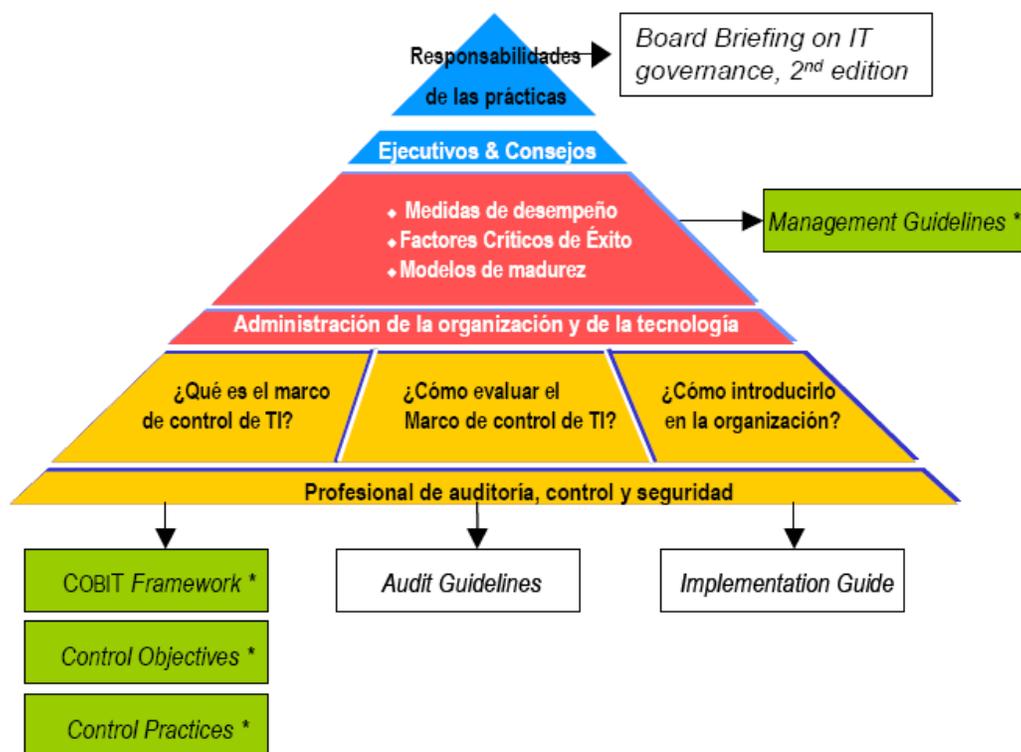
- Representan el consenso de expertos fuertemente enfocados en el control y menos en la ejecución.
- Ayudan a optimizar las inversiones en TI, aseguran la entrega de servicios y proveen una medida en contra de la cual juzgar cuando las cosas van mal.

COBIT cubre cuatro características básicas de las cuales se muestran en la figura 1.8.2 :



### 1.8.2 características básicas de COBIT

La estructura documental de COBIT respecto de las áreas y niveles de organización puede verse representada en la imagen 1.8.3:



### 1.8.3 Niveles de organización COBIT

La forma de operar de COBIT puede verse representada en la imagen 1.8.4:



### 1.8.4 Forma de operación de COBIT

### **1.8.5 Mejores prácticas en las bases de datos, controles de bases de datos**

Es crítico que la integridad y disponibilidad de la base de datos se mantenga; esto se asegura a través de los siguientes controles:

- Establecer y obligar la definición de estándares, políticas y procedimientos
- Establecer e implementar el respaldo de datos y los procedimientos de recuperación para asegurar la disponibilidad de la base de datos.
- Establecer los niveles necesarios de controles de acceso para los elementos de datos, tablas y archivos para prevenir acceso inadvertido o no autorizado.
- Establecer controles para asegurar que sólo el personal autorizado pueda actualizar la base de datos.
- Establecer controles para manejar los problemas de concurrencias, tales como múltiples usuarios deseando actualizar los mismos elementos de datos al mismo tiempo (por ejemplo, el uso de commit, el bloqueo de registros o archivos).
- Establecer controles para asegurar la precisión, totalidad y consistencia de los elementos de datos y las relaciones en las bases de datos. Es importante que estos controles estén contenidos en las definiciones de tablas o columnas.
- Usar puntos de verificación de las bases de datos para minimizar la pérdida de datos y los esfuerzos de recuperación para reiniciar el proceso después de una falla del sistema.
- Ejecutar la optimización de las bases de datos para reducir espacio de disco no utilizado y verificar las relaciones de datos definidas.
- Seguir los procedimientos de reestructuración cuando se hagan cambios lógicos, físicos y de procedimientos.
- Usar herramientas de reporte de desempeño de las bases de datos para monitorear y mantener la eficiencia de la base de datos.
- Minimizar la habilidad para usar medios que no sean sistemas de aplicación o fuera de los procedimientos de seguridad para acceder a los datos de la base.

Al realizar este tipo de control en bases de datos se mantiene mucho mejor la base de datos, mas rápida y con una mayor integridad de datos.

### **1.8.6 Infraestructura de Tecnologías de Información, *frecuentemente abreviada ITIL***

Es un marco de trabajo de las mejores prácticas destinadas a facilitar la entrega de servicios de tecnologías de la información (TI). ITIL resume un extenso conjunto de procedimientos de gestión ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI. Estos procedimientos son independientes del proveedor y han sido desarrollados para servir como guía que abarque toda infraestructura, desarrollo y operaciones de TI.

La biblioteca de infraestructura de TI (ITIL) toma este nombre por tener su origen en un conjunto de libros, cada uno dedicado a una práctica específica dentro de la gestión de TI. Tras la publicación inicial de estos libros, su número creció rápidamente (dentro la versión 1) hasta unos 30 libros. Para hacer a ITIL más accesible (y menos costosa) a aquellos que deseen explorarla, uno de los objetivos del proyecto de actualización ITIL versión 2 fue agrupar los libros según unos conjuntos lógicos destinados a tratar los procesos de administración que cada uno cubre. De esta forma, diversos aspectos de los sistemas de TIC, de las aplicaciones y del servicio se presentan en conjuntos temáticos.

Aunque el tema de Gestión de Servicios (Soporte al Servicio y Entrega de Servicios) es el más ampliamente difundido e implementado, el conjunto de mejores prácticas ITIL provee un conjunto completo de prácticas que abarca no sólo los procesos y requerimientos técnicos y operacionales, sino que se relaciona con la gestión estratégica, la gestión de operaciones y la gestión financiera de una organización moderna.

Los ocho libros de ITIL y sus temas son:

Gestión de Servicios de TI

1. **Prestación de Servicios**
2. **Soporte al Servicio**

Otras guías operativas

3. **Gestión de la infraestructura de TI**
4. **Gestión de la seguridad**
5. **Perspectiva de negocio**
6. **Gestión de aplicaciones**
7. **Gestión de activos de *software***

Para asistir en la implementación de prácticas ITIL, se publicó un libro adicional con guías de implementación (principalmente de la Gestión de Servicios):

8. **Planeando implementar la Gestión de Servicios**

Adicional a los ocho libros originales, más recientemente se añadió una guía con recomendaciones para departamentos de TIC más pequeños:

## **1 .9- Seguridad en Bases de Datos**

### **Objetivo**

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.

#### **1.9.1 Seguridad**

La información es uno de los activos más importantes de las entidades, y de más especial en algunos sectores de actividad.

Es indudable que cada día las entidades dependen de mayor medida de la información y de la tecnología, y que los sistemas de información están más soportadas por la tecnología, frente a la realidad de hace pocas décadas.

Por otra parte, hace unos años la protección era más fácil, con arquitecturas centralizadas y terminales no inteligentes, pero hoy en día los entornos son realmente complejos, con diversidad de plataformas y proliferación de redes, no sólo internas sino también externos, incluso con enlaces internacionales.

Entre las plataformas físicas (hardware) pueden estar: máquinas grandes y máquinas departamentales y personales, solos o formando parte de red, e incluso PCS portátiles. Esta diversidad acerca la información a los usuarios, si bien hace mucho más difícil proteger los datos, especialmente porque los equipos tienen filosofías y sistemas operativos diferentes, incluso a veces siendo del mismo fabricante.

Hay tres aspectos principales, como distintas vertientes de la seguridad.

1. **La confidencialidad:** se cumple cuando solo las personas autorizadas (en un sentido referirnos también a sistemas) pueden conocer los datos o la información correspondiente.
2. **La integridad:** consiste en que sólo las personas autorizadas puedan variar (modificar o borrar) los datos. Además deben quedar pistas para control posterior y para auditoria.
3. **La disponibilidad:** se cumple si las personas autorizadas pueden acceder a tiempo a la información.

El SGBD facilita normalmente mecanismos para prevenir los fallos (subsistema de control), para detectarlos una vez que se han producido (subsistema de detección) y para corregirlos después de haber sido detectados (subsistema de recuperación).

Aspectos fundamentales de la seguridad

- **Confidencialidad.** No develar datos a usuarios no autorizados. Comprende también la privacidad (protección de datos personales).
- **Accesibilidad.** La información debe estar disponible.
- **Integridad.** Permite asegurar que los datos no han sido falseados.

La seguridad en las bases de datos abarca varios temas:

- Cuestiones éticas y legales relativas al derecho a tener acceso a cierta información.

- Cuestiones de política en el nivel gubernamental, institucional o corporativo relacionadas con la información que no debe estar disponible para el público.
- Cuestiones relacionadas con el sistema.
- Necesidad en algunas organizaciones de identificar múltiples niveles de seguridad y de clasificar los datos y los usuarios según estos niveles.

El SGBD debe proveer técnicas que permitan a ciertos usuarios tener acceso a porciones selectas de una base de datos sin tener acceso al resto. Por lo regular un SGBD cuenta con un subsistema de seguridad de autorización de la base de datos que se encarga de garantizar la seguridad de porciones de la base de datos contra el acceso no autorizado.

Existen dos tipos de mecanismos de seguridad:

- Discrecionales, se usan para otorgar privilegios a los usuarios.
- Obligatorios, sirven para imponer seguridad de múltiples niveles clasificando los datos y los usuarios en varias clases de seguridad e implementando después la política de seguridad apropiada de la organización.

Otro problema de seguridad es el acceso a una base de datos estadística, la cual sirve para proporcionar información estadística a partir de diversos criterios. Los usuarios de bases de datos estadísticas están autorizados para usarlas para obtener información estadística sobre una población pero no para tener acceso a información confidencial detallada sobre individuos específicos. La seguridad en bases de datos estadísticas debe cuidar que la información sobre individuos no sea accesible. En ocasiones es posible deducir ciertos hechos relativos a los individuos a partir de consultas, esto tampoco debe permitirse.

Otra técnica de seguridad es el cifrado de datos que sirve para proteger datos confidenciales que se transmiten por satélite o algún tipo de red de comunicaciones. Asimismo el cifrado puede proveer protección adicional a secciones confidenciales de una base de datos. Los datos se codifican mediante algún algoritmo de codificación. Un usuario no autorizado tendrá problemas para descifrar los datos codificados, pero un usuario autorizado contará con algoritmos para descifrarlos.

Entre las obligaciones del DBA está otorgar privilegios a los usuarios y clasificar los usuarios y los datos de acuerdo con la política de la organización. Las órdenes privilegiadas del DBA incluyen los siguientes tipos de acciones:

1. Creación de cuentas
2. Concesión de privilegios.
3. Revocación de privilegios.
4. Asignación de niveles de seguridad.

La acción 1 de la lista sirve para controlar el acceso al SGBD en general, la 2 y la 3 para controlar las autorizaciones discrecionales y la 4 controla la autorización obligatoria

## **1 .10- Performance and Tuning**

### **Objetivo**

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

### **1.10.1 Performance and Tuning**

El performance and tuning es generalmente una aplicación informática, pero los mismos métodos se pueden aplicar a los mercados económicos, las burocracias o de otros sistemas complejos. La motivación para esta actividad se denomina un problema de rendimiento, que puede ser real o previsto. La mayoría de los sistemas responden al aumento de la carga con un cierto grado de disminución de rendimiento. Una capacidad del sistema para aceptar mayor carga es la escalabilidad, y la modificación de un sistema para manejar una mayor carga es sinónimo de rendimiento de ajuste.

Sistemático de ajuste sigue estos pasos:

1. Evaluar el problema y establecer los valores numéricos que clasificar el comportamiento aceptable.
2. Medir el desempeño del sistema antes de la modificación.
3. Identificar la parte del sistema que es fundamental para mejorar el rendimiento. Esto se llama cuello de botella.
4. Modificar la parte del sistema para eliminar los cuellos de botella.
5. Medir el rendimiento del sistema después de la modificación.

Un problema de rendimiento se puede identificar por lo lento que parezca el sistema. Esto generalmente ocurre porque un sistema de alta carga, causa una sobrecarga de parte del sistema para llegar a un límite en su capacidad de respuesta creando un cuello de botella.

Muchas técnicas se utilizan para mejorar el rendimiento. Entre ellos se encuentran la optimización de código, el equilibrio de la carga, almacenamiento de caché, espacio de disco duro, etc.

## **1 .11- Modelo Orientado a Objetos**

### **Objetivo**

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.

### **1.11.1 UML (Lenguaje Unificado de Modelado)**

**UML**, por sus siglas en inglés, *Unified Modeling Language* es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, sólo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

## **1.12- Tópicos Avanzados de Bases de Datos**

### **Objetivo**

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

#### **1.12.1 DatawareHouse**

El Data WareHouse Es un repositorio de datos de muy fácil acceso, alimentado de numerosas fuentes, transformadas en grupos de información sobre temas específicos de negocios, para permitir nuevas consultas, análisis, reportear y decisiones.

#### **1.12.2 Los elementos básicos de un Data Warehouse**

- Sistema fuente: sistemas operacionales de registros donde sus funciones son capturar las transacciones del negocio. A los sistemas fuentes también se le conoce como Legacy System.

- Área de tráfico de datos: es un área de almacenamiento y grupo de procesos, que limpian transforman, combinan, remover los duplicados, guardan, archivan y preparan los datos fuente para ser usados en el Data Warehouse.
- Servidor de presentación: la máquina física objetivo en donde los datos del Data Warehouse son organizados y almacenados para queries directos por los usuarios finales, reportes y otras aplicaciones.
- Modelo dimensional: una disciplina específica para el modelado de datos que es una alternativa para los modelos de entidad – relación.
- Procesos de negocios: un grupo de actividades de negocio que hacen el Data Warehouse.
- Data Mart: un subgrupo lógico del Data Warehouse completo.
- Data Warehouse: búsquedas fuentes de datos de la empresa. Y es la unión de todos los data marts que la constituyen.
- Almacenamiento operacional de datos: es el punto de integración por los sistemas operacionales. Es el acceso al soporte de decisiones por los ejecutivos.
- OLAP: actividad general de búsquedas para presentación de texto y números del Data Warehouse, también un estilo dimensional específico de búsquedas y presentación de información y que es ejemplificada por vendedores de OLAP.
- ROLAP: un grupo de interfases de usuarios y aplicaciones que le dan a la base de datos relacional un estilo dimensional.
- MOLAP: un grupo de interfases de usuarios, aplicaciones y propietarios de tecnología de bases de datos que tienen un fuerte estilo dimensional.
- Aplicaciones para usuarios finales: una colección de herramientas que hacen los queries, analizan y presentan la información objetivo para el soporte de las necesidades del negocio.
- Modelado de aplicaciones: un sofisticado tipo de cliente de Data Warehouse con capacidades analíticas que transforma o digiere las salidas del Data Warehouse.
- Meta Data: toda la información en el ambiente del Data Warehouse que no son así mismo los datos actuales.

Estos son los elementos que contiene el data warehouse para poder mostrar la información procesada y poder realizar un análisis con los datos introducidos.

### **1.12.3 Los procesos básicos del Data Warehouse**

- Extracción: este es el primer paso de obtener la información hacia el ambiente del Data Warehouse.

- Transformación: una vez que la información es extraída hacia el área de tráfico de datos, hay posibles pasos de transformación como; limpieza de la información, tirar la basura que no sirve, seleccionar únicamente los campos necesarios para el Data Warehouse, combinar fuentes de datos, haciéndolas coincidir por los valores de las llaves, creando nuevas llaves para cada registro de una dimensión.
- Carga: al final del proceso de transformación, los datos están en forma para ser cargados.

Con estos procesos al desplegar la información sería de una manera más entendible y de acuerdo con los datos proporcionados sería la complejidad del manejo de la información.

## 2. PROPUESTA VÍA WEB DE LA ADMINISTRACIÓN DE BASES DE DATOS

### 2.1 Características generales del proyecto

El proyecto consiste en una aplicación vía Web que administra, crea y eliminar bases de datos, tablas, usuarios y realiza respaldos de la información contenida.

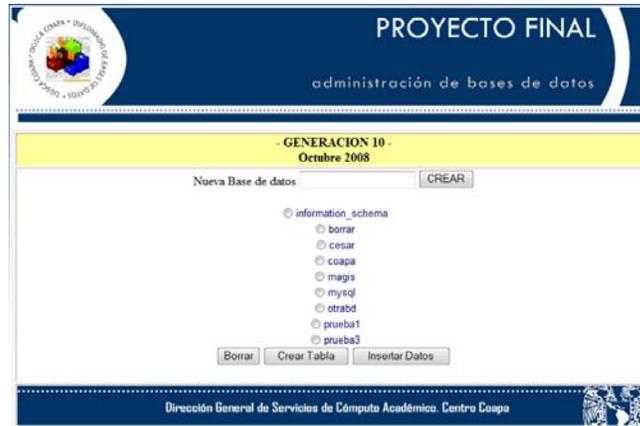
El la figura 2.1.1 se muestra el acceso al sistema colocando el nombre del servidor, nombre del usuario y la contraseña para poder acceder a la aplicación.

#### 2.1.1 acceso al sistema

En la figura 2.1.2 se muestra el menú de administración, esta contiene la administración de bases de datos, Administración de logias y la realización de respaldos.

#### 2.1.2 menú de administración.

En el menú de creación de bases de datos como se muestra la figura 2.1.3, se coloca el nombre de la base de datos a crear y allí mismo muestra las bases de datos que ya han sido creadas en el sistema, también se pueden crear tablas e insertar datos.



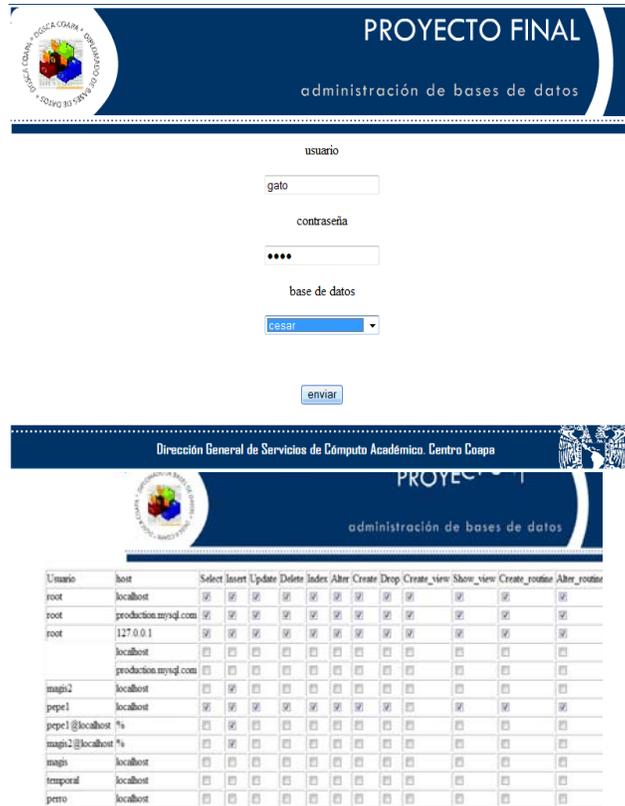
### 2.1.3 Menú administración de bases de datos

En el menú de creación de tablas figura 2.1.4 pide el nombre de la tabla a crear y los campos y tipos de dato que contendrá dicha tabla.



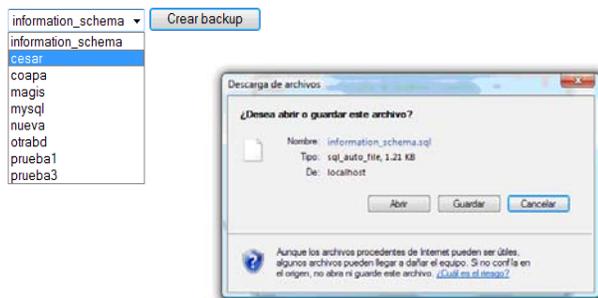
### 2.1.4 creación de tablas

En el menú de creación y administración de cuentas de usuario como se muestra en la figura 2.1.5. para poder crear un usuario se coloca el nombre del usuario, el password y la base de datos o tabla donde tiene permisos para poder hacer una función específica.



### 2.1.5 administración de cuentas de usuario

En la figura 2.1.5 está el menú de la creación de respaldos, esta aplicación guarda un archivo con extensión .sql con la opción de elegir la ruta donde deseamos guardar nuestro respaldo.



### 2.1.5 creación de respaldos

## 2.2 Propuesta de un proyecto

El anterior proyecto es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas Web. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y modificar tablas, borrar, editar y añadir campos, ejecutar sentencias SQL, administrar claves en campos, administrar privilegios de usuario, y crear respaldos.

Se puede crear cualquier tipo de base de datos que se desee o lo que el usuario requiera, en este sentido esta limitado a la imaginación de quienes utilicen la plataforma para administrar las bases de datos. Por ejemplo se puede crear base de datos de un catálogo de autos, de materiales de pisos y azulejos, de muebles, productos enlatados, en una papelería, etc. Teniendo así, un mejor control en cuanto a existencias y precios de los productos.

En este caso la propuesta consiste en la administración de una escuela de cómputo. En la cual, se crea una base de datos para que contenga los antecedentes de todos lo alumnos, por ejemplo:

- Nombre.
- Apellidos.
- Fecha de nacimiento.
- Edad.
- Sexo.
- Domicilio.
- Código postal.
- Teléfono
- Teléfono Celular.
- Correo electrónico.
- En caso de menores: nombre del tutor que autoriza la inclusión de los datos del tutelado en el fichero.
- Cursos a los que se inscribe.

Una vez teniendo los antecedentes personales, puede contener un historial con los datos de los cursos que ha tomado o los cursos y en los que está inscrito:

- Número de cursos.
- Nombre del curso.
- Calificación obtenida.
- Clave del curso.
- Tipo de pago.
- Periodo.
- Horario.
- Días de clase.
- Duración en horas.
- Profesor.
- Programa.
- Aula.
- Estatus (Impartido/Cancelado).

Se debe tener una tabla donde se den de alta los datos de los cursos que se impartirán:

- Clave del curso.
- Nombre del curso.
- Folio del curso.
- Periodo en que se impartirá.
- Horario.
- Días del curso.
- Profesor.

Y por otra parte debemos tener una tabla en la cual podamos introducir los datos para poder generar un contrato al profesor que imparte algún curso

- No contrato.
- Folio grupo.
- Nombre del curso.
- Tipo de curso.
- Status.
- Profesor.
- Periodo del curso.
- Número de horas.
- Honorarios por hora.
- Horario.

Creando la base de datos y las tablas correspondientes se estarían vinculando con su información contenida para así poder obtener datos entendibles.

## **Conclusiones**

Al concluir este trabajo sobre el informe del Diplomado Administración de bases de datos fueron muchos los esfuerzos y conocimientos adquiridos durante su elaboración.

En cada uno de los módulos que constan el diplomado fui adquiriendo conocimiento de las bases de datos por su definición, requerimiento, ventajas y características y de aquellas bases de datos que son la colección de datos o información usada para dar servicios a muchas aplicaciones al mismo tiempo.

En cuanto al requerimiento de la aplicación realizada se puede decir que cumple las mismas tareas de análisis del software y tiene como característica el relacionar la información para su organización y asociación, donde la base de datos tiene una ventaja que es utilizar la plataforma para su uso y desarrollo del sistema de aplicación.

También los conocimientos que fui consiguiendo durante la duración del diplomado no solo me servirán en el ámbito académico sino también en el laboral ya que el objetivo es aprender y estar preparados para la solución de problemas. Por otro lado uno de los principales objetivos fue aprender a trabajar en equipo para dar una mejor solución de los problemas.

## **Bibliografía**

- Introducción a los sistemas de bases de datos, Date, C.J. *Editorial Alhambra Mexicana, S.A.2001.*
- Notas de los módulos 1-12 del diplomado Administración de bases de datos.
- Desarrollo Web con PHP y MYSQL, Welling Luke, Thomson Laura, *Editorial Anaya Multimedia, 2005.*
- PHP 5, Cabezas Granado, Luis Miguel, *Editorial Anaya Multimedia, 2004.*