

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**Facultad de Estudios Superiores Aragón**



**“INFORME DEL PROYECTO DE UN SISTEMA  
PARA GENERAR Y ADMINISTRAR  
BASES DE DATOS”**

**QUE PARA OPTAR POR EL GRADO DE:  
INGENIERO EN COMPUTACIÓN**

**PRESENTA:**

**JUAN SALVADOR GARCÍA CONTRERAS**



**Bosques de Aragón, Edo. De México**

**marzo 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.

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

Facultad de Estudios Superiores Aragón



**“INFORME DEL PROYECTO DE UN SISTEMA  
PARA GENERAR Y ADMINISTRAR  
BASES DE DATOS”**

QUE PARA OPTAR POR EL GRADO DE:

**INGENIERO EN COMPUTACIÓN**

PRESENTA:

**JUAN SALVADOR GARCÍA CONTRERAS**

Bosques de Aragón, Edo. De México

marzo 2009



## Contenido

INTRODUCCIÓN .....	3
INFORME GENERAL DEL DIPLOMADO DE “ADMINISTRACIÓN DE BASES DE DATOS” .....	6
1.1 Sistemas de Información y el modelo de datos relacional.....	7
1.1.1 Conceptos generales de las Bases de Datos.....	7
1.1.2 Características básicas de las Bases de Datos relacionales.....	8
1.1.3 Estructura básica de una base de datos Relacional. ....	9
1.1.4 Modelado Relacional.....	10
1.1.5 Proyecto del módulo: Socios de Oracle. ....	16
1.2 Sistemas Manejadores de Bases de Datos Relacionales.....	20
1.2.1 El RDBMS y el ANSI.....	20
1.2.2 Las reglas de Codd.....	21
1.2.3 Principales RDBMS libres y comerciales.....	22
1.3 SQL (Structured Query Language).....	24
1.3.1 Definición de Datos. ....	24
1.3.2 Manipulación de Datos. ....	25
1.3.3 Estructuras de control y procedimientos almacenados.....	25
1. 4 Acceso a datos a través de la Programación de clientes.....	25
1.4.1 Funcionamiento de una aplicación web.....	26
1.4.2 PHP y la conexión a Bases de Datos . ....	26
1.4.3 Proyecto del módulo: Sistema que obtiene información de MySQL .....	29
1.5 Fundamentos de Sistemas Operativos.....	32
1.5.1 Linux .....	32
1.5.2 Windows Server .....	34
1.6 Habilidades directivas para administradores.....	35
1.6.1 Estilos de comunicación. ....	36
1.6.2 Liderazgo. ....	36
1.6.3 Presentaciones y reuniones efectivas.....	37
1.7 Administración de Bases de Datos.....	37
1.7.1 Tareas del administrador. ....	38
1.7.2 Sybase.....	38

1.7.3	PostgreSQL	41
1.7.4	MySQL	41
1.8	Buenas prácticas en la función de la Administración	42
1.8.1	Auditoría informática (ISACA)	42
1.8.2	Objetivos de Control (COBIT)	43
1.8.3	Mejores prácticas (ITIL)	44
1.9	Seguridad en Bases de Datos	45
1.9.1	Seguridad de la información	45
1.9.2	Seguridad en Bases de datos	46
1.10	Performance and Tunning	46
1.11	Modelado orientado a objetos	47
1.11.1	UML (Lenguaje de Modelado Unificado)	48
1.11.2	Proyecto del módulo: Modelado UML del proyecto del Diplomado	49
1.12	Tópicos avanzados de Bases de Datos	55
<b>PROYECTO DE UN SISTEMA QUE GENERA Y ADMINISTRA BASES DE DATOS</b>		<b>56</b>
2.1	Características generales	57
2.2	Características adicionales	57
2.3	Funciones del Sistema	58
<b>CONCLUSIONES</b>		<b>78</b>

# INTRODUCCIÓN

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

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

La Universidad Nacional Autónoma de México, a través de la Dirección General de Servicios de Cómputo Académico, imparte el Diplomado de Administración de Base de Datos, con el objetivo de formar profesionales que apliquen técnicas efectivas en el manejo de los RDBMS, desde su modelado, hasta su implementación, optimización, administración y mantenimiento en un servidor; asimismo, se comprende la trascendencia de las normas ANSI SQL en cada una de las versiones disponibles, y se imparten las principales tareas del administrador, para llevarlas a la práctica.

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

**a) Sistemas de Información y el modelo de datos relacional.**

Se enseñan los conceptos fundamentales de las bases de datos y del modelo relacional, con el cual se trabaja durante todo el diplomado. Se lleva a cabo el proceso de análisis y diseño de bases de datos. Se hace un proyecto de diseño de base de datos en base a cierta información.

**b) Sistemas manejadores de bases de datos relacionales (RDBMS).**

Se conocen las características básicas de los manejadores de bases de datos relacionales más importantes, tales como sus requerimientos en hardware y software. Se asigna un proyecto en el que se diseña una base de datos y se elige en qué lenguaje programarla, en qué RDBMS montarla y qué hardware y sistema operativo debe tener el servidor.

**c) SQL (Structured Query Language).**

Se desarrollan prácticas en las que se aplican las distintas sentencias y cláusulas SQL (Lenguaje de Consulta Estructurado). Se tratan de peticiones de reportes básicos y otras en las que se programan triggers y procedimientos almacenados.

**d) Acceso a datos a través de la programación de clientes.**

Se aprende el lenguaje PHP y JavaScript mediante ejercicios, prácticas y proyectos, cuya aplicación es la conexión y obtención de información de Bases de Datos.

**e) Fundamentos de sistemas operativos.**

Se administran servidores con sistemas operativos Linux y Windows. Se aprenden los comandos y herramientas fundamentales para la administración de usuarios, procesos, archivos y todo lo relacionado con el sistema que hospedará la base de datos.

**f) Habilidades directivas para administradores.**

Se realizan dinámicas para conocer el estilo de comunicación de cada persona y aprender a relacionarse con ellas. Se aprenden técnicas para organizar y desarrollar una reunión y presentación efectivas con diapositivas.

**g) Mejores prácticas en las funciones de un DBA.**

Se conocen los fundamentos básicos de la auditoría informática y las organizaciones dedicadas a ellas, como lo es ISACA (Asociación de Auditoría y Control de Sistemas), las normas que establece, los objetivos de control de COBIT (Objetivos de Control para la Información y la Tecnología Relacionada) y las mejores prácticas de ITIL (Biblioteca de Infraestructura de Tecnología de la Información) relacionadas con la administración de Bases de Datos.

**h) Seguridad en bases de datos.**

Se ponen al descubierto las posibles vulnerabilidades de los equipos a través de la red y se ejecutan técnicas de prevención y corrección ante éstas.

**i) Administración de bases de datos.**

Se instalan, configuran y se hacen tareas de administración y mantenimiento en algunos RDBMS. Tareas como administración de usuarios y respaldos de bases de datos.

**j) Performance and tuning.**

Se realizan tareas de optimización en memoria, en disco, en restricciones a usuarios y modificación de parámetros en el archivo de configuración.

**k) Modelado orientado a objetos.**

Se conocen los conceptos básicos del modelo orientado a objetos. Se trabaja con herramientas UML para realizar los diagramas concernientes a este modelo.

**l) Tópicos avanzados de bases de datos.**

Se conocen los conceptos de Minería de Datos, Dataware Housing y OLAP.

Así, este Diplomado y sus módulos constituyen una solución integral al problema que enfrentan las organizaciones en el manejo y administración de sus Bases de Datos.



# I

## **INFORME GENERAL DEL DIPLOMADO DE “ADMINISTRACIÓN DE BASES DE DATOS”**

## 1.1 Sistemas de Información y el modelo de datos relacional.

El objetivo de este módulo es identificar todos elementos que conforman una base de datos y la diferencia del manejo de archivos de datos.

Es un tema que incluye los tres tipos de conocimiento: factual, conceptual y procedimental. Durante este módulo, se presentaron técnicas, estrategias y las mejores prácticas del diseño de bases de datos relacionales.

Una gran parte de las fallas de los sistemas se debe al mal diseño de las bases de datos. Como administradores de bases de datos, no es una función propia el hacer ese diseño. Sin embargo, sí corresponde solucionar las fallas de un sistema y si éstas se deben al diseño, como ingenieros es un deber rediseñar para optimizar.

### 1.1.1 Conceptos generales de las Bases de Datos.

La tabla 1.1.1 muestra los conceptos fundamentales de las bases de datos. Es necesario tener este tipo de conocimiento para que todos estemos a la par en este aspecto.

<b>Dato</b>	Unidad mínima de información. Hecho aislado que no representa algo. En una BD sería un <i>campo</i> .
<b>Información</b>	Conjunto de datos relacionados. En una BD sería un <i>registro</i> .
<b>Sistema de Información</b>	Elementos que tienen algo en común, con un objetivo para la toma de decisiones. En una BD sería un archivo.
<b>Sistema de Bases de Datos</b>	Estructura de la Base de datos, que engloba: hardware, software, usuarios y datos.
<b>Base de Datos</b>	Conjunto de información organizada y almacenada en un dispositivo físico.
<b>Modelo</b>	Representación / abstracción de la realidad.
<b>Tabla</b>	Conjunto de campos y registros.
<b>Persistencia de datos</b>	En cualquier momento puedo acceder a la BD y obtener de ella información.

Tabla 1.1. 1 Conceptos generales de Bases de Datos.

Desde hace más de 50 años, las bases de datos han ido evolucionando paulatinamente. Hasta hoy, se han establecido cinco modelos de ellas: en los 60's las bases de datos jerárquicas desplazaron al sistema de archivos, que había nacido en los 50's. En los 70's surgieron las bases de datos de red. Pero éstas perdieron terreno en los 80's ante las bases de datos relacionales, que hasta hoy se han consolidado como las más eficientes. En los 90's aparecieron las bases de datos orientadas a objetos, y aunque han ido tomando fuerza, no ha sido la suficiente para colocarse como las primeras.

## 1.1.2 Características básicas de las Bases de Datos relacionales.

Entre las características básicas de las Bases de Datos Relacionales se puede mencionar:

### a) Independencia lógica y física de los datos.

Independencia lógica significa que algún cambio en la aplicación que los maneja no afecta directamente a la Base. Independencia física quiere decir que los datos pueden estar repartidos en varios dispositivos físicos o pueden cambiarse constantemente sin que afecte la estructura de la base de datos.

### b) Redundancia mínima.

No hay datos repetidos innecesarios. Esto se puede eliminar desde el modelado.

### c) Acceso concurrente por parte de múltiples usuarios.

Muchos usuarios pueden acceder al mismo tiempo a la base de datos y obtener, si quisieran, la misma información.

### d) Integridad de los datos.

La integridad, en términos generales, existe si un dato es válido y completo, de acuerdo a las condiciones o restricciones que da el dueño de la base. Estas restricciones se conocen como reglas del negocio. Con las bases relacionales es sencillo mantener la integridad de los datos, gracias al modelado y al manejador de bases relacionales, llamado formalmente Sistema Manejador de Bases de Datos Relacionales.

### e) Consultas complejas.

Gracias al lenguaje de consulta de las bases relacionales (SQL) se pueden hacer consultas que requiera de información de muchas tablas y de una gran cantidad de datos.

### f) Respaldo y recuperación.

La sencillez de la estructura de una base de datos relacional, permite que su respaldo y restauración se ejecuten sin problemas.

### g) Seguridad.

Se refiere a qué y quiénes pueden tener acceso o hacer algo sobre la base de datos. Los manejadores relacionales permiten configurar altas medidas de seguridad.

### h) Acceso a través de lenguajes de programación.

Existen lenguajes de programación que permiten crear aplicaciones que puedan conectarse a las bases de datos de manera que cualquier usuario pueda realizar operaciones con éstas. Por ejemplo, cualquier persona podría darse de alta en una base de datos tan sólo con poner su nombre y oprimir un botón.



PAIS		
id_pais	nombre_pais	area_pais

Imagen 1.1.2 Tabla con campos sin registros.

Como se ve en la imagen 1.1.3, la tabla PAIS tiene dos registros. Desde el momento en que se le asignó el valor '1' al campo 'id\_pais', ya se tenía un registro.

PAIS		
id_pais	nombre_pais	area_pais
1	Mexico	Norteamerica
2	Brasil	Sudamerica

Imagen 1.1.3 Tabla con campos y registros.

## Tipos de relación entre tablas.

Existen tres tipos de relaciones, que se explican mediante estos ejemplos sencillos:

**1:1 (uno a uno)** Para una asesoría extraclase. Un alumno sólo puede inscribirse con un profesor, y un profesor sólo puede tener asignado un alumno.

**1:M (uno a muchos)** Para una liga de fútbol. Un equipo tiene muchos jugadores y un jugador sólo puede pertenecer a un solo equipo.

**M:M (muchos a muchos)** Para una escuela. Un profesor tiene muchos alumnos y un alumno tiene muchos profesores.

De hecho, una de las ventajas de las bases de datos relacionales sobre las de red y las jerárquicas es que permite representar los tres tipos de relación. En otras notaciones, en vez de M se escribe una N o un símbolo  $\infty$ .

### 1.1.4 Modelado Relacional.

El modelado relacional tiene su base en el diagrama Entidad-Relación. Sin embargo, el buen diseño de una Base de Datos Relacional requiere de un proceso de modelado que involucra:

- Modelo Conceptual.
- Modelo lógico.
- Normalización.
- Modelo Físico.

## Modelo Conceptual.

El modelado conceptual se compone de tres pasos:

### a) Identificar las entidades.

Son los actores principales. Identificar una entidad puede en algunos casos no ser una tarea fácil. Básicamente una entidad tendrá atributos que la definan y tendrá más de un registro. La práctica en el modelado relacional provee de experiencia para identificar las entidades. Durante este Diplomado, realizamos prácticas suficientes y eficientes para desarrollar esta habilidad.

### b) Realizar una matriz de relaciones.

Ayuda a observar de una manera organizada la relación entre cada entidad.

### c) Dibujar un diagrama Entidad-Relación.

Se compone de Entidades, Vínculos y Propiedades de las entidades. La imagen 1.1.4 es un ejemplo sencillo de un diagrama Entidad-Relación que representa la siguiente oración:

*“Un matrimonio puede tener varios hijos,  
pero un hijo pertenece sólo a un matrimonio”.*

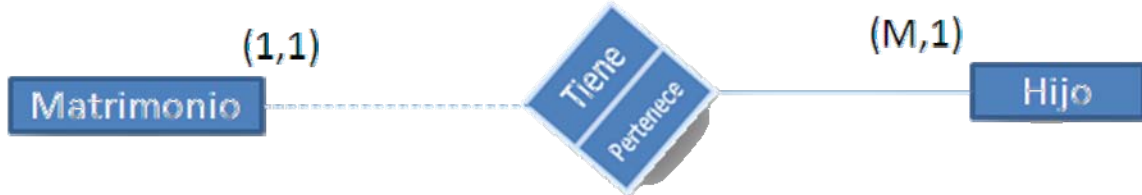


Imagen 1.1.4 Diagrama Entidad-Relación en su forma básica (sin propiedades)

Del enunciado, se identifica que las entidades son 'Matrimonio' e 'Hijo' (aún y cuando se hable de ellas en plural, como regla en el diagrama se escriben en singular y en recuadros). Se han resaltado ciertas partes del enunciado subrayándolas y dos palabras en negrita. Para el primer renglón, el 'puede' se representa con una línea punteada. Para el segundo renglón, la obligatoriedad 'sólo' se representa con una línea continua. En el rombo se escribe el verbo que define la relación entre ellas: Matrimonio tiene Hijo e Hijo pertenece a Matrimonio. De las palabras resaltadas en negritas se concluye que el tipo de relación es 1:M. Pero en el diagrama se representa como se muestra en la imagen 1.1.5.

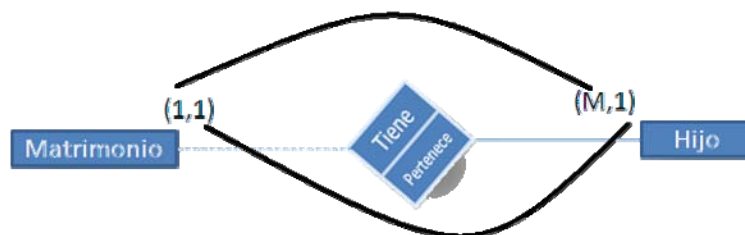


Imagen 1.1.5 Representación del tipo de relación en un Diagrama Entidad-Relación

Es de observarse que el 1 de la izquierda se relaciona con la M, que también está de lado izquierdo. Y el 1 de la derecha se relaciona con el otro 1 que también está a la derecha. Esto se lee de la siguiente forma:

**Un** Matrimonio tiene **Muchos** Hijos y **Un** Hijo pertenece a **un** Matrimonio.

La imagen 1.1.6 es la forma completa del diagrama Entidad-Relación, ya que de las Entidades se desprenden propiedades.

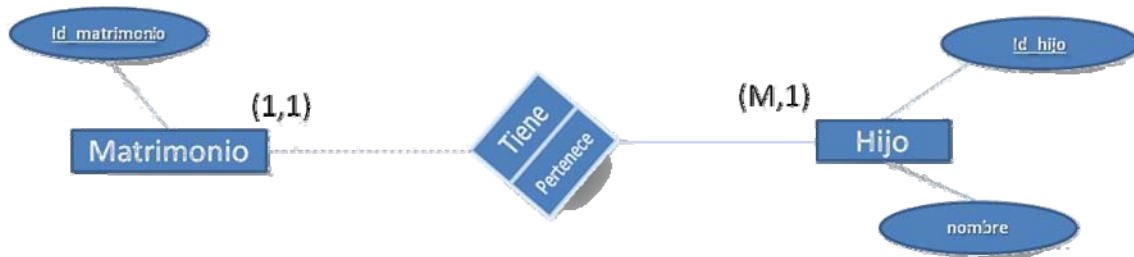


Imagen 1.1.6 Diagrama Entidad-Relación con Propiedades.

La tabla 1.1.2 muestra los tipos de propiedades y su representación en este diagrama.

Propiedad	Ejemplo	Gráfico
<b>Monovaluada</b>	Por ejemplo, la propiedad multivaluada "teléfono" de un Cliente puede tener distintos valores al mismo tiempo. El teléfono de casa, el celular y el de oficina.	
<b>Multivaluada</b>		
<b>Base</b>	Las derivadas son por lo general, el resultado de hacer cálculos de otras propiedades tipo base. Por ejemplo, la propiedad "Promedio" es resultado de hacer un cálculo con la propiedad base "Calificación"	
<b>Derivada</b>		
<b>Simple</b>	Por ejemplo, la propiedad compuesta "Dirección" puede estar formada por las propiedades simples "País", "Estado", "Ciudad", "Calle_numero".	
<b>Compuesta</b>		
<b>Clave</b>	Sólo una propiedad puede ser clave. Esto se refiere a que es la llave primaria.	
<b>No clave</b>		

Tabla 1.1.2 Tipos de propiedades y su representación gráfica en un diagrama Entidad-Relación

## Modelo lógico.

Para comenzar este modelado, deben hacerse tres sencillos cambios con respecto al conceptual:

### a) Diagramar Relaciones.

Las Entidades se dibujan como Relaciones, es decir, como tablas (Imagen 1.1.7).



Imagen 1.1.7 Cambio en la representación gráfica de las Entidades. Del modelo conceptual al lógico.

### b) Establecer atributos.

Las propiedades pasan a ser atributos en las Relaciones. Cada tipo de propiedad tiene una regla para convertirse en atributo:

- **Monovaluada, Base, Simple, No clave** → Pasan igual
- **Derivada** → Se elimina
- **Compuesta** → Sólo pasan sus Simples
- **Clave** → Pasa como llave primaria (PK)
- **Multivaluada** → Pasa como otra Relación

### c) Diagramar el tipo de relación.

Los tipos de relación se dibujan de la siguiente forma:



Cuando dos entidades tienen una relación muchos a muchos (M:M) se crea otra tabla llamada 'transitiva', la cual, como buena práctica, debe tener una combinación de los nombres de las tablas de las cuales deriva y además contener como atributos las llaves primarias de éstas.

Para entender mejor el proceso de diseño con el modelado relacional, en la sección 1.1.5 se muestra el proceso completo con el caso práctico que me fue asignado como proyecto en este módulo del Diplomado.



## Normalización.

El proceso de Normalización no tiene un orden definido. Es decir, que se puede hacer antes o después del modelado lógico, o antes o después del modelado físico. Sin embargo, como buena práctica en el diseño, es conveniente hacerlo después del lógico. Como ya se mencionó antes, haber hecho correctamente el modelado conceptual y el lógico, representa en gran medida no tener que hacer las tres formas de la Normalización, porque en el diagrama Entidad-Relación si se identifican correctamente las propiedades multivaluadas y compuestas, al pasarlas al diseño lógico, prácticamente se habrán dejado las tablas en 1ª forma normal.

La Normalización se conoce como el 'proceso de descomposición sin pérdida'. Esto significa que se pueden quitar muchas partes del diseño para evitar repetición de datos o atributos o relaciones que no deban existir, sin que con ello se pierda información.

La Normalización consta de etapas conocidas como 'formas normales'. Dependiendo del diseño, se necesitará avanzar a la siguiente forma normal. Algunas veces no se llega a la tercera forma. La tabla 1.1.3 contiene una explicación básica de las tres formas normales.

Forma	Regla
<b>1FN</b>	Todos los atributos deben guardar valores atómicos. <i>Ejemplo:</i> CLIENTE {id_cliente, nombre, telefono} Las reglas del negocio dicen que se harán búsquedas por apellidos. Por lo tanto el atributo 'nombre' que incluye el nombre de pila y los apellidos debe ser dividido. Quedaría así: CLIENTE {id_cliente, nombre, apellido_paterno, apellido_materno, telefono} Ahora todos los atributos guardan valores atómicos según las reglas del negocio.
<b>2FN</b>	Se debe estar en 1FN y que todos los atributos no clave dependan de la llave primaria. <i>Ejemplo:</i> FACTURA {id_factura, descripcion_factura, total_factura, nombre_cliente} Aunque en una factura sí aparece el nombre del cliente, debería quitarse y quedar así: FACTURA {id_factura, descripcion_factura, total_factura} CLIENTE {id_cliente, nombre_cliente}
<b>3FN</b>	Se debe estar en 2FN y cada atributo sólo depende de la llave primaria y no de algún otro atributo. <i>Ejemplo:</i> SOCIO {id_socio, pais_socio, ciudad_socio, nombre_socio} El atributo no clave 'pais' depende de la clave 'id_pais' a través del atributo no clave 'ciudad'. En 3FN debería ser así: SOCIO {id_socio, nombre_socio} CIUDAD {id_ciudad, nombre_ciudad} PAIS {id_pais, nombre_pais}

Tabla 1.1.3 Las tres formas de la normalización de bases de datos.

Antes de comenzar a normalizar, es necesario comprender uno de los conceptos más usados en el diseño de bases de datos relacionales: 'atomicidad'. Se puede decir que un atributo tiene atomicidad o que un atributo tiene un valor atómico, o simplemente que un atributo es atómico.

Se llama valor atómico porque hace referencia a la teoría que decía que el átomo era el elemento más pequeño de la materia y que éste era indivisible. De la misma manera, hay atributos que no pueden dividirse.

La Normalización más allá de ser un proceso complejo, puede ser, más bien, difícil de explicar. Requiere de mucha práctica para entender una diferencia clara entre la 2FN y la 3FN. Como ya se ha dicho, el diseño no es una actividad propia del administrador de bases de datos, es por ello que sólo se ha visto este tema de manera general y no se profundizara en él. Las prácticas efectivas del Diplomado dan un enfoque más claro del proceso de Normalización.

### Modelo físico.

La diferencia entre éste y el conceptual es que ahora adoptamos dos nuevos conceptos: llave foránea y tipo de dato.

La llave foránea (FK) es un atributo en la entidad que a su vez, es la llave primaria (PK) de la entidad con la que se relaciona.

Cada atributo debe tener definido un tipo de dato. Básicamente un tipo de dato define si el valor de un atributo será una cadena de caracteres o será numérico. Al establecer un tipo de dato para un atributo también se establece la longitud de ese valor. Más adelante se verán los tipos que se pueden utilizar según el RDBMS que gestione la base de datos.

La imagen 1.1.8 muestra un ejemplo del modelo físico y así es como finalmente quedaría la última fase del modelado relacional antes de usar SQL para crearla con un sistema manejador de bases de datos relacionales (RDBMS).

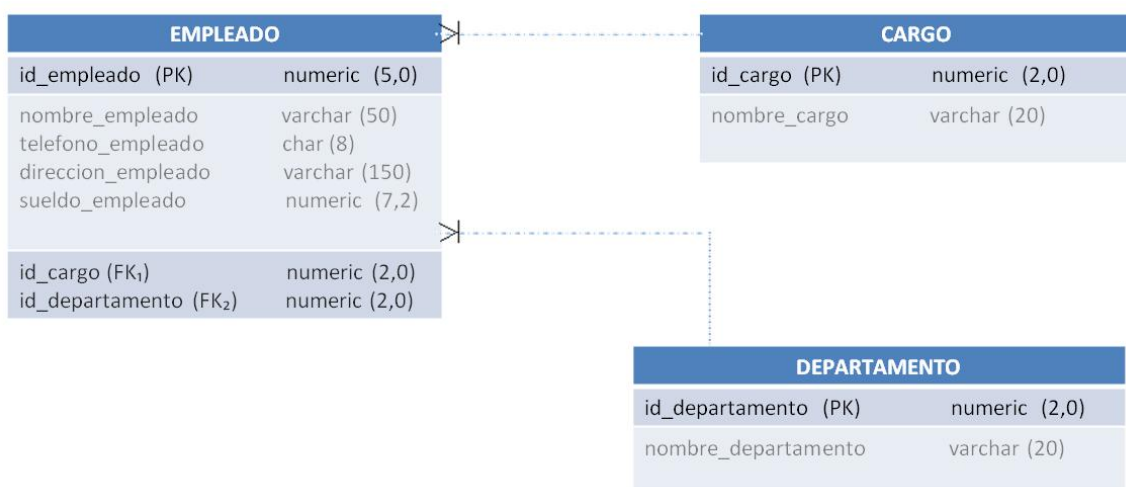


Imagen 1.1.8 Ejemplo del modelo físico. Incluye tipos de datos y llaves foráneas.

### 1.1.5 Proyecto del módulo: Socios de Oracle.

#### Problema.

A continuación se describe el problema que se modelará:

Nuestro Grupo de Usuarios de Oracle regional ha aumentado a más de 200 socios. Nosotros somos una organización voluntaria, y nuestros registros están revueltos. Necesitamos un sistema de información que nos ayude a registrar todas las actividades a las que estamos a cargo. Definitivamente necesitamos automatizar el registro de nuestros socios. Por cada socio necesitamos conocer su nombre, puesto, dirección para la correspondencia, número de teléfono de su oficina, tipo de membrecía (individual o corporativa), y si el socio esta o no al corriente en sus cuotas. Nosotros recolectamos cuotas en base a una anualidad, y la fecha de vencimiento de pago de ésta es en Enero para todos los socios. A nosotros también nos gustaría saber en qué compañía trabaja un socio, pero mantener esta información actualizada es un verdadero lío porque nuestros socios siempre están cambiando de compañía. Para cada compañía registramos el nombre de la compañía, dirección y el tipo de negocio. Tenemos claves estandarizadas para identificar cada tipo de negocio. Nosotros nada más registraríamos la dirección de la oficina matriz de cada empresa. Quisiéramos registrar la fecha de cada evento, número de asistentes, donde se lleva a cabo, cuánto dinero se gasto en él, y cualquier comentario sobre el evento. Tratamos todos los comentarios como si fueran anónimos. Este conjunto de comentarios es sólo una forma de texto gratuita de cualquier longitud. Numeramos cada conjunto de comentarios, y así tenemos múltiples conjuntos de comentarios por evento. También necesitamos registrar que tipo de plataformas de computadora utilizan nuestros miembros. Tenemos un código de identificación que es único y de tres dígitos para cada tipo de plataforma. Por ejemplo, 001 es para IBM/MVS; 002 es para IBM/MV; 003 es para VAX/VMS; 020 es para OS/2; 030 es para PC/DOS; 050 es para Sun UNIX y 080 es para otras plataformas en UNIX.

#### Modelo conceptual.

Se identifican las entidades. En este caso son las siguientes;

SOCIO  
COMPAÑÍA  
EVENTO

La tabla 1.1.4 muestra la relación que hay entre cada entidad:

Un socio trabaja en una compañía.

Un socio puede participar en muchos eventos.

Una compañía tiene muchos socios.

Un evento tiene muchos socios.

	SOCIO	COMPAÑÍA	EVENTO
SOCIO		trabaja	Participa
COMPAÑÍA	Tiene		-
EVENTO	tiene	-	

Tabla 1.1 . 4 Matriz de relaciones.

La Imagen 1.1.9 es el Diagrama Entidad-Relación construido a partir del problema y la matriz de relaciones.

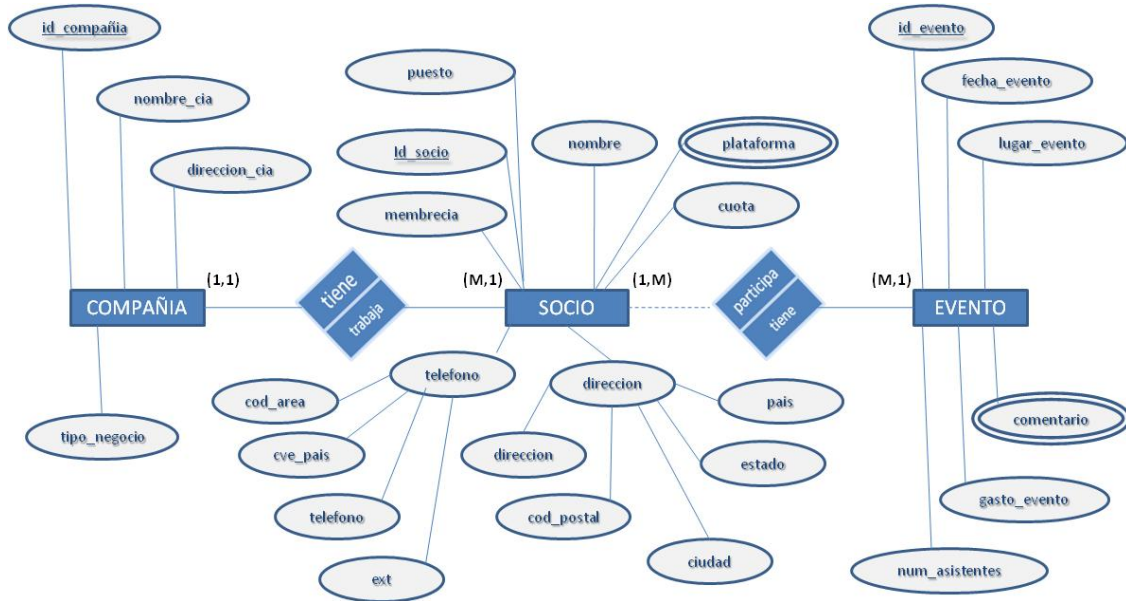


Imagen 1. 1 . 9 Diagrama Entidad-Relación.

## Modelo lógico.

Conviene tener un apoyo como la tabla 1.1.5 para organizar el cambio de un modelo a otro.

Tipo de propiedad	Atributos	Transición
<b>monovaluadas base simple no clave</b>	nombre_cia, direccion_cia, tipo_negocio, membrecia, puesto, cuota, fecha_evento, lugar_evento, gasto_evento, num_asistentes, nombre, cod_area, cve_pais, telefono, ext, direccion, pais, ciudad, estado, cod_postal	Pasan igual
<b>derivada</b>	-	-
<b>compuesta</b>	direccion, telefono	Se eliminan
<b>clave</b>	Id_compañía, id_socio, id_evento	Pasan como PK de su entidad
<b>multivaluada</b>	plataforma, comentario	Se convierten en otra entidad

Uno de los objetivos de este primer módulo ha sido saber utilizar una herramienta Case para facilitar el diseño y modelado de las bases de datos. El diagrama mostrado en la imagen 1.1.10 está hecho en CA ERwin Process Modeler.

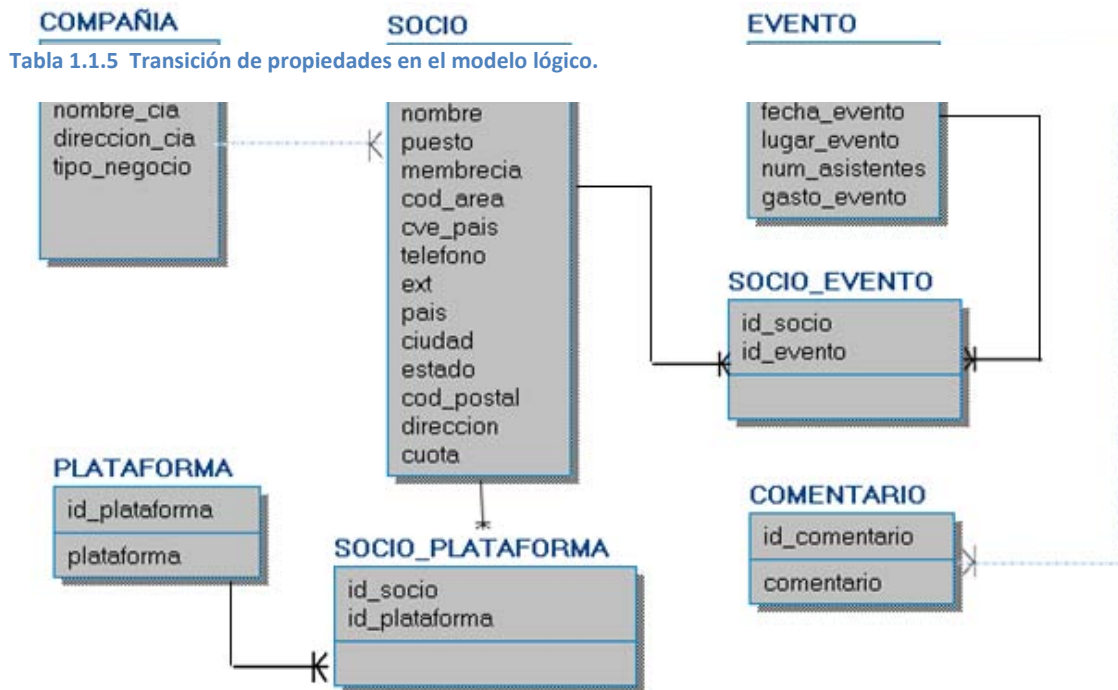


Imagen 1.1.10 Modelo Lógico.

Como puede verse en la imagen 1.1.10, las dos propiedades ‘plataforma’ y ‘comentario’, al ser multivaluadas, pasaron como entidades. Además se tienen otras dos entidades: SOCIO\_PLATAFORMA y SOCIO\_EVENTO. Estas son las llamadas tablas transitivas. Y surgieron porque la relación de SOCIO con EVENTO y de SOCIO con PLATAFORMA era de muchos a muchos en ambos casos:

Un socio puede tener muchas plataformas y una plataforma puede ser utilizada por muchos usuarios.

Un socio puede asistir a muchos eventos y a un evento pueden asistir muchos socios.

Así es como se deben romper las relaciones muchos a muchos. Las tablas transitivas llevan el nombre de las dos entidades de las que procede, así como sus atributos sólo son las llaves primarias de esas entidades; y por lo tanto, al ser PK de otras, aquí se convierten en FK. Aunque en realidad no es hasta el modelo físico en donde se ponen todas las FK’s de todas las entidades.

### Normalización.

Aunque existen estándares en cuanto a las notaciones gráficas, cada herramienta Case tiene la suya propia. Así, el Erwin tiene otra representación gráfica de los tipos de relación. El siguiente modelo (Imagen 1.1.11) es como ha quedado después de aplicarle la 2FN. La primera no fue necesaria, porque ya todos los valores eran atómicos y a la tercera no se llegó.

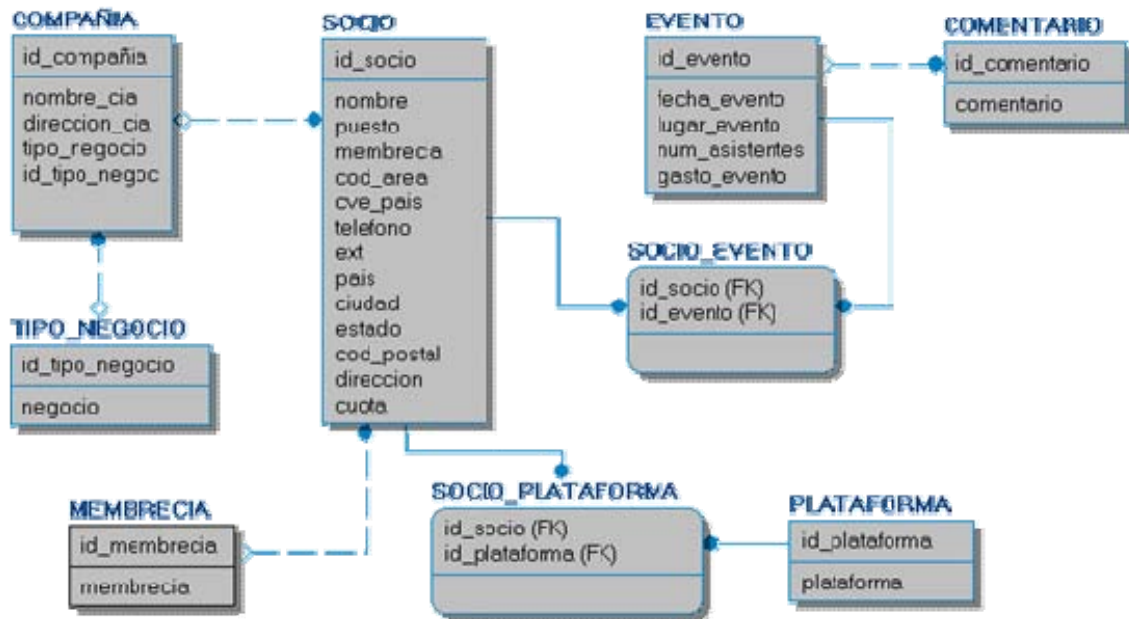


Imagen 1.1. 11 Modelo Lógico Normalizado.

El atributo 'tipo\_negocio' de COMPAÑÍA se saca como otra entidad porque éste no depende de la llave primaria. Membrecía sale como otra entidad porque no depende de 'id\_socio'. Es decir, el socio X no deja de ser el mismo socio por cambiar su tipo de membresía.

### Modelo físico.

Ahora sólo es cuestión de definir el tipo y la longitud de los datos que tendrá cada atributo. Además de establecer las llaves foráneas (FK) en donde corresponda. Por ejemplo, COMPAÑÍA tiene una relación 1:M con SOCIO, por ello a SOCIO se le agrega la FK 'id\_compañia'. TIPO\_NEGOCIO tiene una relación 1:M con COMPAÑÍA, por eso a ésta se le agrega la FK 'id\_tipo\_negocio', tal como se ve en la imagen 1.1.12.

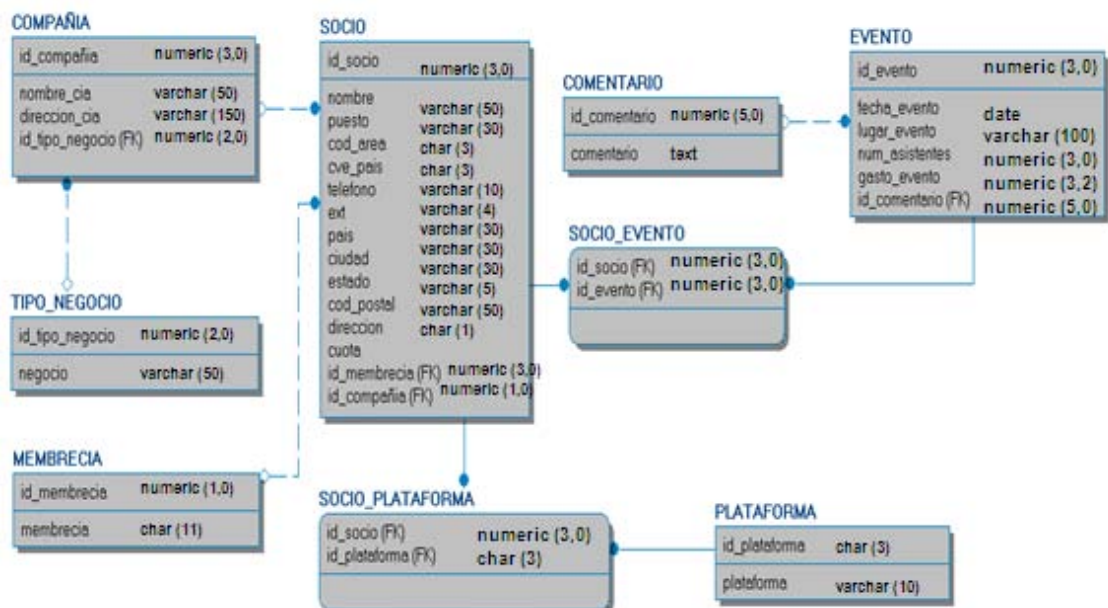


Imagen 1.1. 12 Modelo Físico.

## 1.2 Sistemas Manejadores de Bases de Datos Relacionales.

Como administradores de bases de datos se debe tener el criterio necesario para estructurar un ambiente adecuado para cada necesidad. Habrá bases de datos que necesiten menos requerimientos que otras. Necesitarán más memoria, más espacio en disco, un sistema operativo determinado, etc. Este módulo enseña cómo y qué elegir.

### 1.2.1 El RDBMS y el ANSI.

Un RDBMS es un software que se encarga de gestionar bases de datos. A través de él se pueden manipular (crearlas, eliminarlas, crear tablas, insertar datos, respaldarlas, etc.) y controlar su acceso.

La principal función de los Sistemas Manejadores de Bases de Datos Relacionales es relacionar la información entre sí. Y su principal objetivo es permitir que la información se comparta. Un RDBMS se encarga, además, de facilitar la integridad, la seguridad y el acceso de los datos.

La imagen 1.2.1 muestra la interacción entre los principales componentes de un Sistema de Bases de Datos.

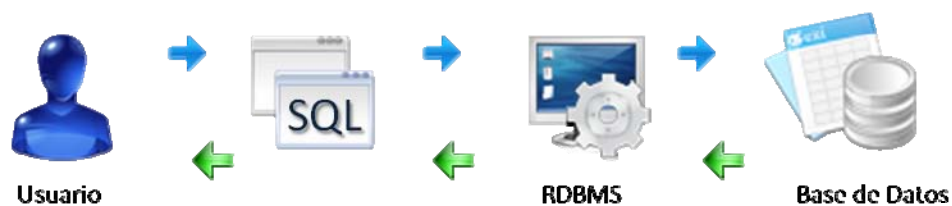


Imagen 1.2. 1 Diagrama de interacción entre componentes de un Sistema de Bases de Datos

La secuencia sería de la siguiente manera:

- El usuario hace una solicitud que se traduce a una petición SQL.
- El RDBMS interpreta y analiza la petición.
- El RDBMS verifica sintaxis del SQL.
- El RDBMS checa que exista la base de datos.
- El RDBMS verifica los permisos del usuario.
- El RDBMS verifica que existan las tablas y elementos.
- El RDBMS ejecuta las operaciones necesarias sobre la base de datos.
- El RDBMS devuelve resultados al usuario.

Todos los sistemas manejadores tienen características que los diferencian de los otros; sin embargo existe el ANSI (Instituto Americano de Estándares Nacionales) que revisa y establece

estándares del SQL que deben seguir todos y cada uno de los RDBMS. De igual manera, cada uno de éstos maneja diversos tipos de datos; y aunque algunos varían de uno a otro manejador, la mayoría debe de tener los tipos estándar del ANSI, para de esta forma asegurar la compatibilidad entre ellos. El objetivo es que si se tiene conocimiento de SQL, éste se puede utilizar en cualquier manejador.

Hasta hoy, la revisión que ANSI hizo en 1992 es la que se ha mantenido y es conocida como ANSI SQL 92.

La tabla 1.2.1 muestra los tipos de datos que se pueden utilizar según el RDBMS que gestione la base de datos.

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

Tabla 1.2. 1 Tipos de Datos de los RDBMS.

## 1.2.2 Las reglas de Codd.

Cuando surgieron las Bases de Datos Relacionales, también se desarrollaron muchos manejadores que se decían relacionales; sin embargo Edgar Codd, inventor del modelo relacional, al ver que muchos de estos DBMS realmente no eran relacionales, desarrolló 12 reglas que debía cumplir un DBMS para ser considerado un RDBMS. La tabla 1.2.2 muestra estas reglas y la definición de cada una de ellas.



Regla	Definición
<b>Regla de la información.</b>	La información se guarda en tablas a nivel lógico y todo dentro de la Base se ve como tal.
<b>Regla del acceso garantizado.</b>	Se necesita el nombre de la tabla, valor de la PK y el nombre de la columna para ingresar a algún dato.
<b>Regla del tratamiento sistemático de un valor nulo.</b>	Se debe manejar de manera correcta los valores desconocidos.
<b>Regla del catálogo dinámico basado en línea.</b>	Son tablas que guardan descripción de la estructura de la base de datos.
<b>Regla del sublenguaje de datos completo.</b>	El SQL no es un lenguaje procedural, no tiene estructuras de control; pero es un estándar en los RDBMS.
<b>Regla de la actualización de vistas.</b>	Una representación lógica de una tabla. Es una estructura que puede mostrar una sola parte o completamente los datos de una tabla.
<b>Regla de actualización, borrado e inserción de alto nivel.</b>	Se pueden actualizar varios registros al mismo tiempo en una sola instrucción.
<b>Regla de independencia física de datos.</b>	Los programas de aplicación y actividades permanecen inalterados si cambio la base de datos a otro dispositivo de almacenamiento.
<b>Regla de independencia lógica de datos.</b>	No importa si realizo una modificación a la base de datos, ésta no se ve alterada en forma general.
<b>Regla de independencia de integridad.</b>	Siempre puedo cambiar las reglas del negocio.
<b>Regla de independencia de distribución.</b>	El modelo de Base de datos Relacional es fácilmente distribuible y visto como una unidad.
<b>Regla de no subversión.</b>	No permite ataques de otro lenguaje de alto nivel.

Tabla 1.2. 2 Las 12 reglas de Codd.

### 1.2.3 Principales RDBMS libres y comerciales.

Actualmente existen varios RDBMS, algunos son gratuitos y otros requieren que se pague licencia por ellos. Algunos son gratuitos pero no de software libre. Al ser un RDBMS de software libre se puede ver y modificar su código fuente para adoptarlo a nuestras necesidades; tal es el caso de MySQL o PostgreSQL, dos manejadores muy potentes. SQL Server, por su parte, ofrece una versión express gratuita, sin embargo, no se tiene acceso al código fuente. La tabla 1.2.3 muestra las características fundamentales de los principales Sistemas Manejadores de Bases de Datos Relacionales.

RDBMS	Características Importantes	Sistema Operativo	Procesador (mínimo)	RAM (mínima)	Disco Duro	Licencia
Microsoft SQL Server	Permite realizar algunas tareas de mantenimiento y administración de la base de datos sin tener que darla de baja. Manejo de varias particiones físicas para almacenamientos de datos flexibles. Permite realizar acciones OLAP (Online Analyzing Processing)	Windows XP, Windows 2000, Windows 2003, Windows 2008	Pentium IV 1.4GHz o equivalente.	128 Mb	280 Mb	Comercial (Existe una versión Express libre)
Sybase	Diseñado para soportar aplicaciones OLTP (On Line transaction Processor, ambiente diseñado para insertar, actualizar y borrar datos en una base datos) Conectividad con clientes ODBC y JDBC. Soporte para BLOB's (Large Objects). Compresión de copias de respaldo.	HP-UX AIX IRIX Red Hat Windows	Pentium III o equivalente	128 Mb	240 Mb	Comercial
Oracle	Ofrece varias plataformas de desarrollo para Internet y aplicaciones tradicionales, tales como: XML, Enterprise Java Engine, SQL y PL/SQL, C, C++, entre otras. Conectividad con clientes ODBC y JDBC. Soporte para BLOB's. Ofrece escalabilidad y performance sin modificar las aplicaciones	Solaris HP-UX Compaq Tru64 AIX HP Alpha Linux Windows NT/2000/XP Professional	Pentium IV o equivalente.	1 Gb	400 Mb	Comercial
DB2 (IBM)	Soporta Bases de datos de más de 4 TB. Provee acceso a cualquier tipo de cliente. Permite manejo de base de datos distribuidas. Contiene plataformas de desarrollo con SPL: (Informix, Stored Procedure Language), C, Java, XML. Conectividad via ODBC, JDBC, OLE/DB	IBM AIX. SG IRIX. Sun Solaris. HP-UX. Compaq Tru64. Linux. Windows NT/2000/XP/2003.	Procesador de 600 MHz	512 Mb	120 Mb	Comercial
PostgreSQL	Velocidad. Confiabilidad. Flexibilidad. Bajo costo de operación. Conformación a estándares ANSI Soporta replicación de bases de datos. Interfases nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, XML.	Corre bajo cualquier plataforma UNIX	Pentium III o equivalente	128 Mb	140 Mb	Libre
MySQL	Contiene esquemas de almacenamiento independiente que se pueden seleccionar de acuerdo a las necesidades. InnoDB para transacciones y bloqueo de registros. MyISAM sin transacciones Soporte para SSL. Queryz con manejo de cache que puede incrementar el performance de la base de datos en un 200%. Permite manejo de replicación de bases de datos.	Linux Windows. FreeBSD Sun Solaris. IBM-AIX. MacOS X. HP-UX.	Procesador de 200 MHz	128 Mb	200Mb	Libre

Tabla 1.2. 3 Cuadro comparativo de los RDBMS más importantes.

## 1.3 SQL (Structured Query Language).

Un lenguaje de programación se necesita para crear aplicaciones con las cuales las personas pueden manipular y acceder a bases de datos. Sin embargo, no es el lenguaje de programación el que lo hace directamente; éste sólo se conecta a la Base y obtiene información gracias a que en el código incluye el Lenguaje de Consulta Estructurado (SQL). Al ser administradores de bases de datos, se puede utilizar directamente SQL sobre el RDBMS y hacer peticiones, tal como se vio en la sección anterior. Estas peticiones se conocen como 'queries'.

SQL se compone de tres tipos de sublenguajes, los cuales se conforman de cierto tipo de sentencia comunes, como se aprecia en la imagen 1.3.1.

Componente	Descripción	Sentencias
DDL	Lenguaje de Definición de Datos. Se crean, modifican o eliminan objetos (BD, tablas, vistas)	CREATE, ALTER, DROP
DML	Lenguaje de Manipulación de Datos. Se seleccionan o insertan datos; se borra el contenido y se actualizan objetos.	SELECT, INSERT, DELETE, UPDATE
DCL	Lenguaje de Control de Datos. Se otorgan o revocan privilegios para el acceso a los datos de la Base.	GRANT, REVOKE

Tabla 1.3.1 Sublenguajes de SQL y sus sentencias.

Cada clase del módulo realizábamos una práctica, misma que consistía en promedio de veinte peticiones de reportes que teníamos que mostrar ejecutando sentencias SQL. Las prácticas fueron estructuradas de manera que como alumnos adquiriéramos experiencia en generar reportes para situaciones comunes en las empresas. Las prácticas fueron hechas en MySQL, PostgreSQL, SQL Server y Oracle, con las mismas 'queries', demostrando que SQL es un lenguaje estándar que funciona en cualquier RDBMS.

### 1.3.1 Definición de Datos.

Las bases de datos tienen distintos objetos, tales como las tablas, las vistas, procedimientos almacenados, usuarios, reglas, entre otros. Las sentencias DDL permiten definir la estructura de estos objetos, ya sea para crearlos, modificarlos o eliminarlos. Sin embargo, todo gira en torno a las tablas; éstas son el elemento fundamental que compone a una base de datos relacional, ya que ellas son las estructuras que contienen nuestra información.

Al crear una tabla deben tomarse en cuenta consideraciones como el nombre que se dará. Cada RDBMS tiene sus propias reglas y palabras reservadas que no pueden utilizarse para este fin. Al utilizar la definición de los datos, se pueden crear reglas, asignar valores predeterminados o por 'default', asignar llaves ya sean primarias y/o foráneas. Si en la creación de una tabla no se definió alguno de estos elementos o se quiere hacer algún cambio o adición a una ya creada, siempre se puede utilizar ALTER para hacerlo. De la misma forma, si

se quiere eliminar por completo las tablas o cualquier otro objeto de la base de datos, incluso esta misma, se utiliza la clausula DROP.

### **1.3.2 Manipulación de Datos.**

La manipulación de datos comprende la mayor parte de las clausulas SQL. Mientras las de definición permiten crear, alterar o eliminar la estructura de los objetos, las de manipulación permiten trabajar directamente con los registros. Con el DML se puede obtener, eliminar, proveer y actualizar información de las tablas. Para ello se puede establecer condiciones y así sólo manipular los valores que se necesiten. Al utilizar las clausulas necesarias para obtener información de las tablas, entre todo lo que se puede hacer, destaca: definir el orden en que se quiere que los registros aparezcan en las queries, definir un límite de registros, realizar operaciones aritméticas entre campos, obtener el promedio, el valor mínimo o el máximo, o contar registros.

### **1.3.3 Estructuras de control y procedimientos almacenados.**

Aunque SQL no es propiamente un lenguaje de programación, también cuenta con estructuras de control de flujo:

- If then – end if
- if then – else – end if
- for loop
- while loop

Estas estructuras son de gran utilidad cuando se programan 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.

## **1. 4 Acceso a datos a través de la Programación de clientes.**

Un cliente es una aplicación cuyo principal objetivo es facilitarle al usuario la manipulación, la definición y el control de acceso a las Bases de Datos y sus objetos, a través de interfaces que se suponen de sencillo uso. Los clientes que con más frecuencia se programan son

aplicaciones web, ya que uno de los objetivos de las Bases de Datos es que la información esté accesible para varios usuarios al mismo tiempo desde donde sea y en el momento que sea. El objetivo de este módulo, es utilizar los distintos elementos que brinda el lenguaje de programación PHP, para la consulta y manipulación de información en las bases de datos. PHP es un lenguaje óptimo para el desarrollo de muchos sistemas y junto con ActionScript y JavaScript se logran crear aplicaciones bastante eficaces.

### 1.4.1 Funcionamiento de una aplicación web.

Para programar una aplicación web se necesita conocer cómo es que ésta funciona y comunica con el servidor. La imagen 1.4.1 muestra en resumen, con un ejemplo, el proceso que hay entre el cliente y el servidor. En este módulo del Diplomado se aprende a crear aplicaciones web dinámicas con PHP, con conexión a bases de datos, principalmente de MySQL.

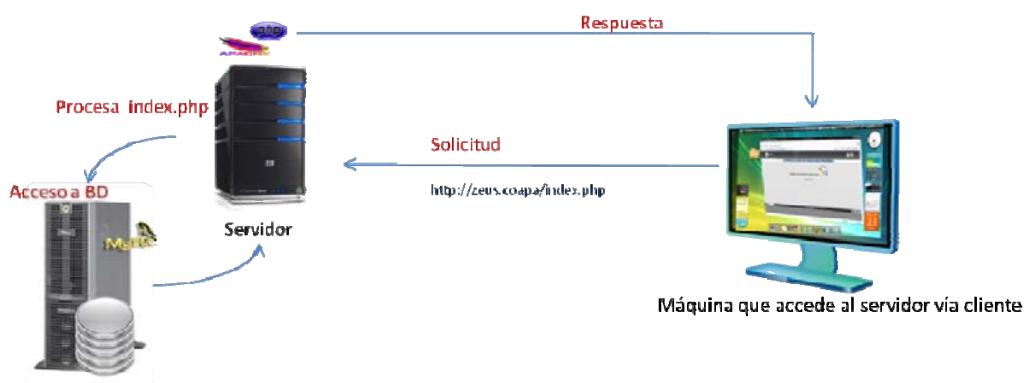


Imagen 1.4.1 Funcionamiento de una aplicación web como cliente.

Cuando se teclea la dirección electrónica de un sitio o en su defecto su IP, se muestra en el navegador de internet la aplicación como una página web. Una de las virtudes de programar aplicaciones web es que muchos usuarios pueden tener acceso a la información desde sus máquinas sin que en cada una de ellas esté instalada una aplicación. Todas las operaciones y por lo tanto toda la ocupación de recursos como RAM y CPU que requiere la aplicación se realizan en el equipo Servidor. Sólo se ve el resultado del proceso que se hace en el servidor y en la base de datos, cuando se hace una petición o se envía información desde la máquina.

### 1.4.2 PHP y la conexión a Bases de Datos .

PHP es uno de los lenguajes de software libre más potentes y entre sus características más destacadas están las siguientes:

- Fácil de aprender.
- Sintaxis similar a C.
- Multiplataforma.
- Librerías para acceder a BD.
- Soporta P.O.B.
- Funciones de criptografía.
- Instrucciones para manejo de concurrencia.
- Tiene muchos huecos de seguridad.

Aunque una aplicación web no se crea sólo con PHP, éste es el que le da el toque de dinamismo. Básicamente, con HTML se envía información mientras que PHP la procesa y ejecuta operaciones necesarias, tales como las aritméticas, el manejo de cadenas o el uso de estructuras de control. La imagen 1.4.2 muestra de manera sencilla cómo se utilizan estos dos lenguajes. Existen dos métodos de envío de datos: GET y POST. La diferencia básica es que al utilizar el primero se observan en la barra de direcciones los datos que se mandan, mientras que con el segundo esto no sucede.

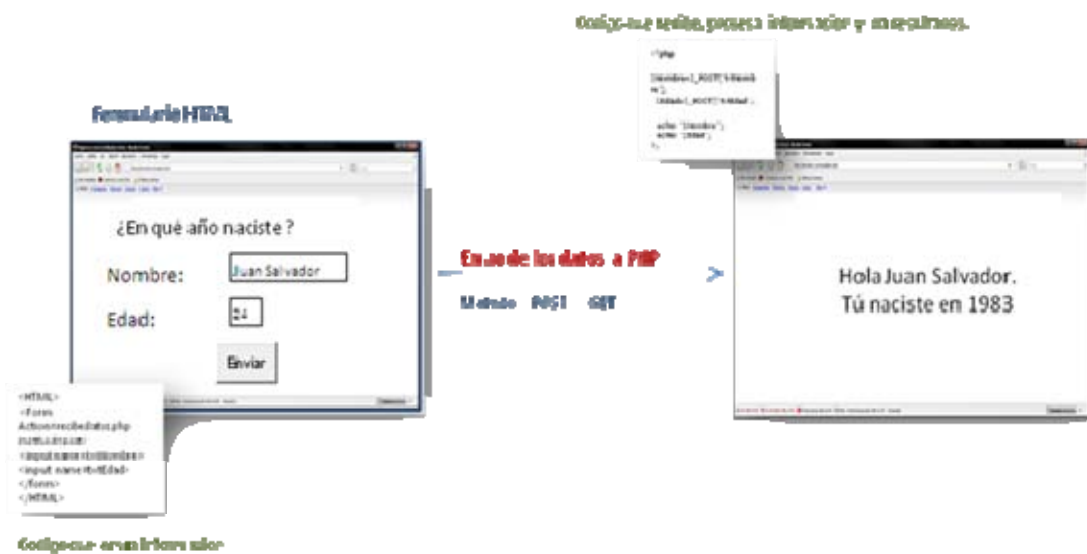


Imagen 1.4. 2 HTML envía, PHP recibe y genera un resultado.

PHP tiene funciones para conectarse a servidores de base de datos con RDBMS como Sybase, PostgreSQL y MySQL, entre otros, y obtener información de éstos. La tabla 1.4.1 contiene algunas de las funciones de estos tres manejadores. Además de los ejercicios prácticos del módulo, que fueron desde la sintaxis básica hasta el envío de datos a través de formularios, se desarrollaron dos aplicaciones completas, una para administración de usuarios (alta, baja y cambios) y otra que fue el proyecto final. Las dos incluyeron la conexión a bases de datos MySQL y el manejo de consultas SQL.

Funciones PHP para acceso y control de Bases de Datos.		
MySQL	SyBase	PostgreSQL
<u>mysql_connect</u> ___ Abre una conexión a un servidor MySQL	<u>sybase_connect</u> ___ abre una conexión con un servidor Sybase	<u>pg_connect</u> ___ Abre una conexión
<u>mysql_close</u> ___ cierra el enlace con MySQL	<u>sybase_close</u> ___ cierra una conexión Sybase	<u>pg_close</u> ___ Cierra una conexión PostgreSQL
<u>mysql_affected_rows</u> ___ Devuelve el número de filas afectadas de la última operación MySQL	<u>sybase_affected_rows</u> ___ obtiene el número de filas afectadas por la última consulta	<u>pg_affected_rows</u> ___ Devuelve el número de filas afectadas.
<u>mysql_create_db</u> ___ Crea una base MySQL	<u>sybase_get_last_message</u> ___ Returns the last message from the server	<u>pg_fetch_array</u> ___ obtiene una fila en la forma de un array
<u>mysql_query</u> ___ Envía una sentencia SQL a MySQL	<u>sybase_num_fields</u> ___ obtiene el número de campos de un resultado	<u>pg_fetch_row</u> ___ obtiene la fila como un array enumerado
<u>mysql_drop_db</u> ___ Borra una base de datos MySQL	<u>sybase_num_rows</u> ___ obtiene el número de filas de un resultado	<u>pg_host</u> ___ Devuelve el nombre del host
<u>mysql_error</u> ___ Devuelve el texto del mensaje de error de la última operación MySQL	<u>sybase_query</u> ___ envía una consulta a Sybase	<u>pg_last_error</u> ___ Devuelve el texto del mensaje de error de la última operación MySQL
<u>mysql_result</u> ___ Devuelve datos de un resultado	<u>sybase_result</u> ___ obtiene datos de un resultado	<u>pg_copy_to</u> ___ Copia el contenido de una tabla dentro de un arreglo.
<u>mysql_select_db</u> ___ Selecciona un base de datos MySQL	<u>sybase_select_db</u> ___ selecciona una base de datos Sybase	<u>pg_query</u> ___ Ejecuta una consulta SQL
<u>mysql_stat</u> ___ Obtener el status actual del sistema	<u>sybase_fetch_field</u> ___ obtiene la información del campo	<u>pg_num_fields</u> ___ Devuelve el número de campos de un resultado.
<u>mysql_list_tables</u> ___ Lista las tablas en una base de datos MySQL	<u>sybase_fetch_row</u> ___ obtiene una fila como un array enumerado	<u>pg_num_rows</u> ___ Devuelve el número de filas de un resultado.

Tabla 1.4.1 Funciones PHP para MySQL, SyBase y PostgreSQL.

### 1.4.3 Proyecto del módulo: Sistema que obtiene información de MySQL .

Las características y funciones de este proyecto son:

Formulario en el que el usuario ingresa sus datos de acceso, elige el servidor y opcionalmente, si se quiere conectar directamente a una Base. Además, debe escribir el código de seguridad que ve en una imagen “human check” a fin de prevenir el ataque por software. (Imagen 1.4.3)

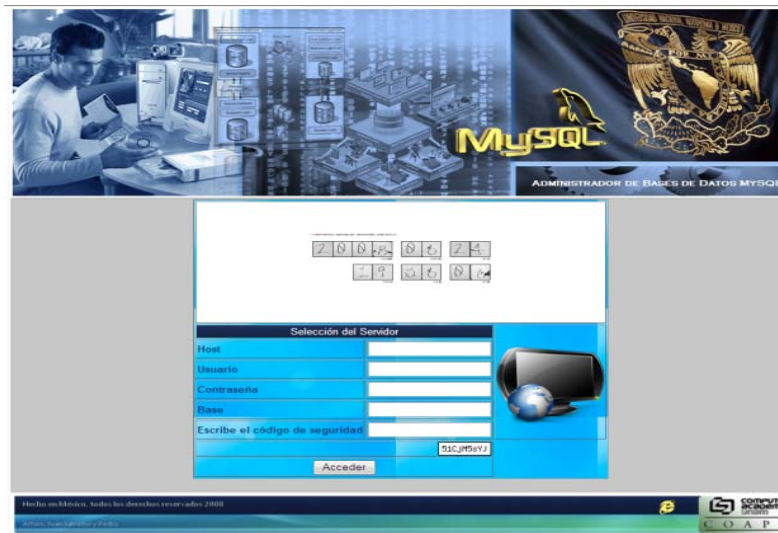


Imagen 1.4.3 Formulario de acceso.

Se envían los datos y se validan con la función `mysql_connect('servidor', 'usuario', 'contraseña')`. Si los datos son correctos se conecta al servidor, de lo contrario con `mysql_error()` se obtiene la causa por la que no hubo conexión.

Si en el formulario inicial, el usuario no elige una Base, se enlistan las que estén disponibles en el servidor haciendo uso de php y SQL: `mysql_query("SHOW DATABASES")`. Se muestra información del servidor usando `mysql_stat()`. Y se crea otro formulario HTML para que al listarse las Bases, el usuario pueda elegir una para acceder a ella (Imagen 1.4.4).

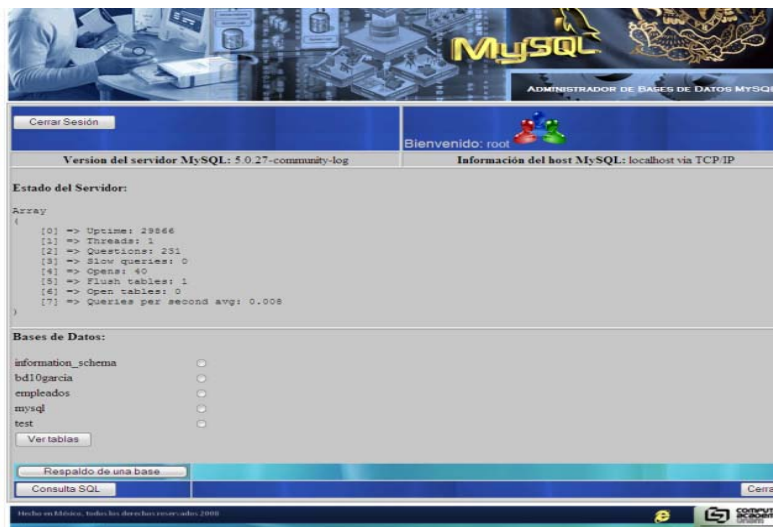


Imagen 1.4.4 Lista de Bases de Datos del servidor.



Pantalla que enlista las tablas de una Base elegida, ya sea desde donde se enlistaron o desde el formulario de acceso inicial. Desde cualquiera que haya sido, se ha enviado a otro código php el nombre de esta base y se hace la conexión a ésta con la función `mysql_select_db('base')`. Ahora php sabe en qué Base se está. Se usa una función y SQL para mostrar sus tablas: `mysql_query("SHOW TABLES")`. Las tablas se listan dentro de un formulario HTML para que el usuario elija una y vea los campos que la forman (Imagen 1.4.5).

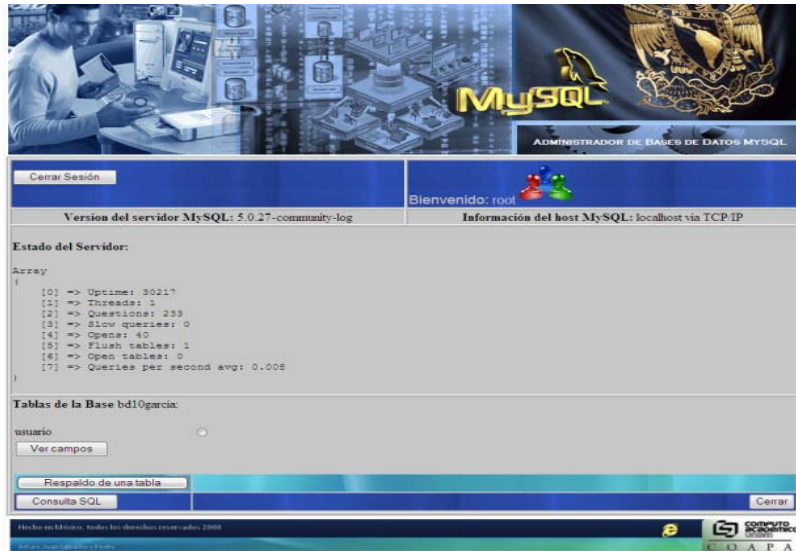


Imagen 1.4.5 Lista de tablas de una base de datos.

Pantalla que muestra la descripción de los campos de una tabla elegida. Muestra de cada campo su nombre (`mysql_field_name`), su tipo de dato (`mysql_field_type`), su longitud (`mysql_field_len`) y si es llave primaria y/o si puede contener valores nulos (`mysql_field_flags`) (Imagen 1.4.6).

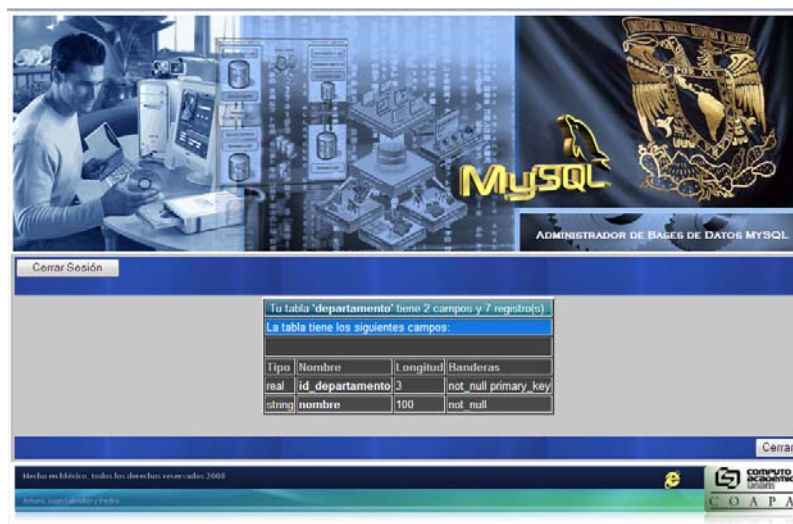


Imagen 1.4.6 Descripción de los campos de una Tabla.

Formulario que permite al usuario ingresar en una caja de texto una consulta SQL y ver el resultado en la misma página. Los datos en el área de texto son enviados como argumento a la función `mysql_query` (Imagen 1.4.7).

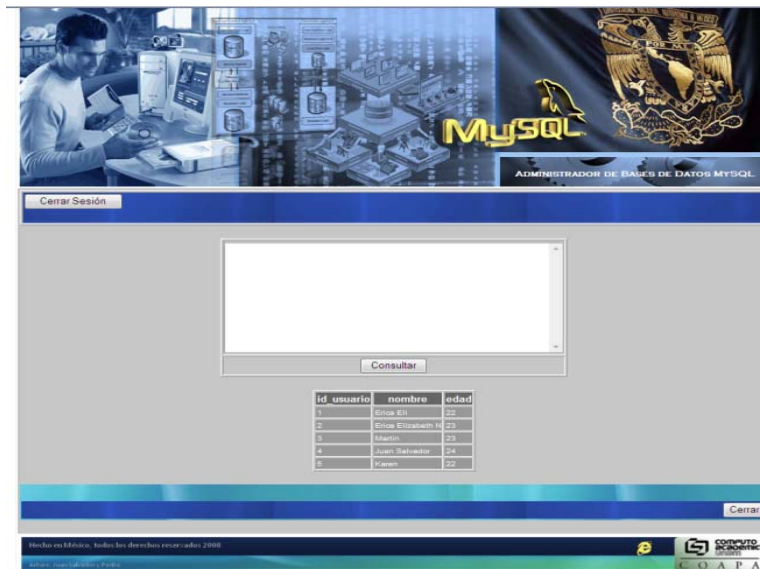


Imagen 1.4.7 Interfaz de consultas SQL.

Interfaz que permite respaldar una base de datos (Imagen 1.4.8). El usuario elige una Base y da el nombre del respaldo. Al oprimir el botón 'Hacer respaldo' se ejecuta con los datos enviados el comando propio de MySQL: `mysqldump`.

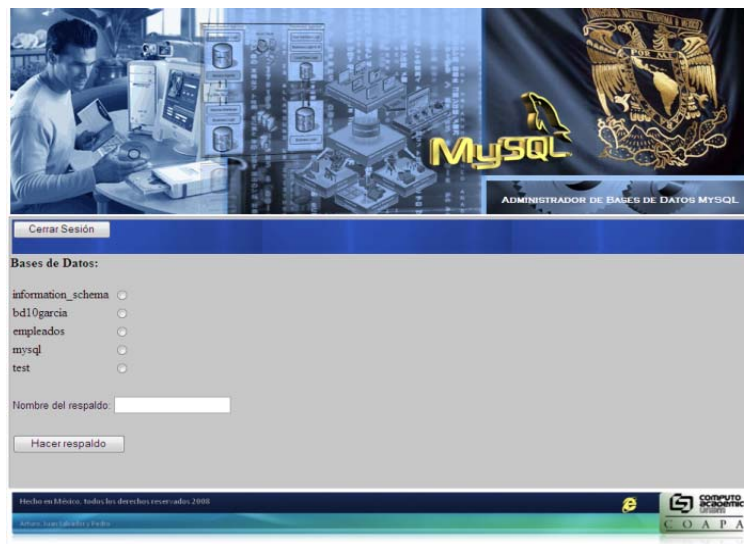


Imagen 1.4.8 Interfaz de Respaldo de una Base de Datos.

Interfaz en la que el usuario elige y respalda una tabla (1.4.9). A diferencia del respaldo de Base, en este caso se utiliza la función php para consultas y SQL: `mysql_query("SELECT * INTO OUTFILE <archivo_respaldo> FROM <tabla>")`

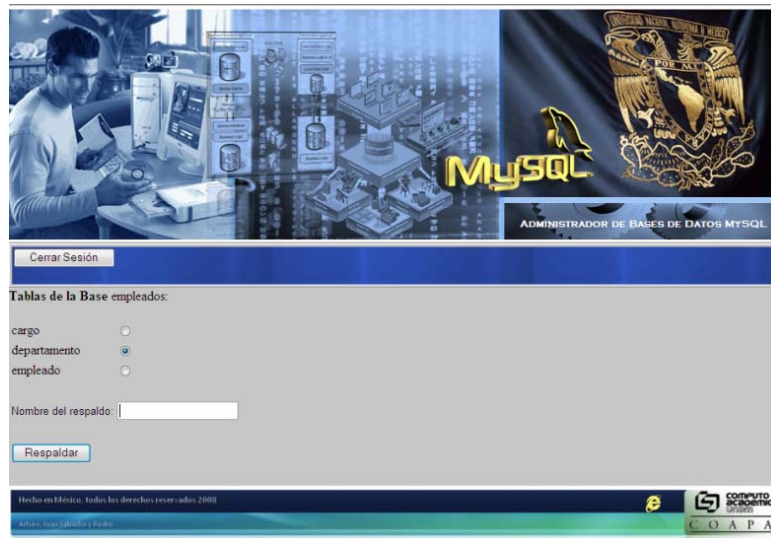


Imagen 1.4.9 Interfaz de respaldo de una Tabla.

## 1.5 Fundamentos de Sistemas Operativos.

El sistema operativo es el software que interpreta las instrucciones del usuario o de las aplicaciones y dice a la computadora qué hacer. Se comunica con periféricos como el monitor, disco duro, impresoras, módems, etc. y gestiona los recursos como la memoria RAM.

Un mismo RDBMS tiene versiones diferentes según el sistema operativo en el que se vaya a instalar. Es por ello que como administradores es fundamental conocer las dos plataformas más importantes: Windows y Linux. Justamente el objetivo de este módulo es conocer y aprender los comandos y herramientas de administración que intervienen en la configuración de un servidor en el que se implante una base de datos.

### 1.5.1 Linux

El sistema operativo Linux tiene tres componentes principales:

- El Kernel, que administra todos los recursos de la computadora (sistemas de archivos, dispositivos, procesos y memoria).
- El Shell, que es la interfaz que se presenta, sea o no gráfica. El Shell recibe las instrucciones, las interpreta y las ejecuta.

- El **árbol de directorios**, que es el conjunto de todos los directorios donde se almacena la información del sistema.

Las tareas de administrador aprendidas en el Diplomado son: la administración de usuarios y grupos, administración de archivos, respaldos, monitoreo de procesos, administración de espacio en disco, monitoreo de red y administración de las bitácoras del sistema.

La creación de usuarios y grupos permite controlar el acceso al sistema, pudiéndoles otorgar distintos permisos.

Cada usuario tiene un identificador único y pertenece al menos a un grupo, el cual al tener ciertos permisos los comparte con todos sus integrantes. Sin embargo, en la creación no se dice sobre qué recursos tienen permisos, sino más bien al recurso se le dice quién y de qué forma puede acceder a él.

La tabla 1.5.1 muestra los comandos para crear, modificar y eliminar grupos y usuarios.

<b>Comandos Linux para cuentas de usuarios y grupos.</b>	
useradd	Crea una nueva cuenta de usuario
usermod	Modifica los parámetros de una cuenta
userdel	Borra una cuenta de usuario
passwd	Modifica la contraseña de una cuenta
groupadd	Crea un nuevo grupo de usuarios
groupmod	Modifica los parámetros de un grupo
groupdel	Elimina un grupo

**Tabla 1.5.1 Comandos básicos de administración de usuarios en Linux**

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

- /etc/passwd
- /etc/shadow
- /etc/group

La tabla 1.5.2 muestra cómo y en qué archivos se registran los datos de las cuentas de usuario. Es de suma importancia conocerlos, ya que aunque no se recomienda hacerlo siempre, la administración de usuarios se puede hacer modificando las entradas a estos archivos. Aunque lo mejor, por seguridad, es usar los comandos de la tabla 1.5.1.

## Archivos de sistema para definición de usuarios.

Cada cuenta de usuario tiene una línea en el archivo `/etc/passwd` que contiene siete campos separados por dos puntos.

*ejemplo:*

```
garciajs : X : 102 : 10 : Juan_Salvador : /home/garciajs : /bin/ksh
```

1. Nombre de usuario.
2. No usado (antiguamente clave).
3. Identificador de usuario (UID).
4. Identificador del grupo primario (GID).
5. Descripción o nombre completo de la persona que representa dicho usuario.
6. Directorio personal.
7. Intérprete de mandatos.

Todas las contraseñas son cifradas y mantenidas en un archivo separado llamado `/etc/shadow`

*ejemplo:*

```
garciajs : $56fTy54$iihGl : 15994 : 0 : 90 : 7 : 15 : :
```

1. Nombre de usuario.
2. Clave codificada.
3. Días transcurridos desde 01-01-1970 hasta el último cambio de contraseña.
4. Días hasta que la clave pueda ser cambiada.
5. Días para pedir otro cambio de clave.
6. Días para avisar del cambio de la clave.
7. Días para deshabilitar la cuenta tras su caducidad.
8. Fecha de caducidad. Expresado en días después de 01-01-1970.
9. Reservado (normalmente ignorado).

Tabla 1.5.2 Archivos de Linux donde son guardados los datos de las cuentas de usuarios.

En Linux hay usuarios y grupos predefinidos, de los cuales el más importante es el usuario root, que pertenece al grupo del mismo nombre. Este usuario no tiene restricciones y puede hacer lo que quiera. Por ello, también por seguridad, no es buena práctica entrar como root ya que él puede eliminar o desconfigurar cualquier cosa por accidente. Sólo hay que utilizarlo cuando sea estrictamente necesario.

Una buena práctica de la administración es dar seguridad controlando el acceso a recursos como los archivos, y dado que en Linux todo es visto como archivos, esta práctica se facilita demasiado. Todos los archivos y los directorios que los alojan tienen un dueño y estos, además del root, pueden decidir quién puede leer, escribir y/o ejecutarlos.

Estando sobre un directorio, al ejecutar el comando `ls -l` se enlistan los archivos que hay, mostrando entre otras cosas los permisos que tienen. Con el comando `chmod` se modifican los permisos de acceso de cada archivo.

### 1.5.2 Windows Server

Una de las cualidades del sistema operativo Windows Server es que, al ser todo gráfico, su administración es muy sencilla y no se requiere usar tantos comandos como en Linux. El Active Directory es un software con el cual se puede llevar a cabo el control y la administración centralizada de todos los usuarios del dominio de la red. Durante este módulo se aprende

cómo configurar un grupo de trabajo y un dominio. Sin embargo, el dominio presenta más ventajas empresariales; comenzando por la escalabilidad, es decir, se puede ir agregando muchas más computadoras que las 10 que permite un grupo de trabajo. Además no sólo se centraliza la administración de usuarios, sino también de archivos o impresoras, entre otros. Es cuestión de que en el servidor se den de alta a los usuarios y entonces éstos podrían acceder desde cualquier computadora de la red o incluso reemplazar su equipo por uno nuevo sin mayores problemas.

Active Directory proporciona métodos para almacenar los datos como nombres, contraseñas y teléfonos del directorio y ponerlos a disposición de los administradores y los usuarios autorizados de la misma red. Se puede crear cuentas de usuarios y de grupos. Esto funciona de la misma manera que como se había visto en Linux; resulta sencillo otorgar permisos a un grupo y agregar miembros a éste para que los compartan. Un usuario puede también pertenecer a varios grupos. Con los sistemas Windows para servidores se utilizan las herramientas administrativas mediante las cuales se pueden otorgar políticas de grupo.

Las políticas de grupo son restricciones que limitan el acceso a los recursos de los equipos y de la red, y se otorgan tanto a usuarios como a computadoras. Es decir que se puede decidir que un usuario x puede acceder a cualquier máquina y tener los mismos privilegios, o decidir que en un grupo de máquinas se puede tener acceso a recursos que en otras no. La administración centralizada que proporciona configurar un dominio es notablemente rápida y sencilla.

No es uno de los objetivos de este módulo presentar una comparación de estos dos sistemas como competencia; sino conocer el tipo de administración que tiene cada uno, ya que en las diversas organizaciones puede existir uno u otro o los dos, y ello dependerá de las necesidades de cada empresa.

## **1.6 Habilidades directivas para administradores.**

Un buen administrador no sólo conoce su equipo material de trabajo, como las computadoras con su software o hardware, sino también, o mejor aún, a su gente. Aquellas personas con las que logra conformar no un grupo sino un equipo.

El objetivo de este módulo es conocer los principios básicos para la administración de la actividad laboral de equipos de trabajo, así como la planeación del tiempo y actividades.

Este módulo enseña la parte humana que se debe tener como administradores, conocer y comprender a las personas que trabajan con nosotros, para descubrir y obtener lo mejor de ellas logrando un mejor funcionamiento grupal.

### 1.6.1 Estilos de comunicación.

Todas las personas tienen un estilo de comunicación dominante. Aprender cuál es el nuestro y saber identificarlos en las personas con las que nos relacionamos, llevarán a una comunicación efectiva. Estos son los estilos de comunicación:

**Visual:** Piensas con imágenes. Representas las ideas, los recuerdos y la imaginación mediante imágenes mentales

**Auditivo:** Piensas con sonidos. Los sonidos pueden ser voces o ruidos.

**Cinestésico:** Representas los pensamientos como sentimientos internos o sensaciones físicas. El sabor y el olor se incluyen en esta categoría sensorial.

Cuando se sabe identificar cuál es el estilo de una persona, se puede de cierta forma imitarla; ya que las personas se sienten más en confianza con otras con las que encuentran detalles en común.

### 1.6.2 Liderazgo.

Más que administrador se debe ser líder. Las técnicas y actividades del módulo hacen entender que nuestro trabajo forma parte de un sistema que integra recursos técnicos, materiales y humanos destinado a lograr objetivos comunes. La tabla 1.6.1 presenta las diferencias que hay entre un administrador y un líder.

ADMINISTRADOR	LÍDER
administra	innova
copia	crea
mantiene	desarrolla
manda	convence
acepta la realidad	investiga la realidad
enfoque a sistemas y estructuras	enfoque en la gente
control	confianza
cómo y cuándo	qué y por qué
hace las cosas bien	hace las cosas correctas

Tabla 1.6.1 Diferencias entre un administrador y un líder.

El liderazgo es el proceso de dirigir a los miembros del grupo hacia el éxito en la consecución de metas y objetivos organizacionales, mediante:

- Inspiración de confianza a su gente.
- Centrarse en las personas como tal y no como cosas.
- Desarrollar un equipo.
- Generar un clima motivador.

### 1.6.3 Presentaciones y reuniones efectivas.

Se dice que cuando hacemos una presentación en una reunión, la audiencia genera una opinión de nosotros basándose en tres características principales:

- Contenido verbal= 7%
- Interés del discurso oral= 38%
- Lenguaje corporal= 55%

Así que antes de comenzar una presentación se deben cuidar aspectos como la apariencia, el volumen y el tono de la voz, la pronunciación, saber quién será la audiencia y saber que todos los movimientos les dicen algo a quienes les hablamos. Algunos consejos básicos para esto son con respecto a las manos:

- No cerrarlas delante de uno mismo.
- No colocarlas tras la espalda.
- No cerrar los puños a menos que se quiera generar un efecto de énfasis.
- No cruzar los brazos.
- No jugar con objetos.

El manejo de las diapositivas es muy importante y se debe entender que hay que cuidar todos los detalles como el contraste de los colores o las imágenes de fondo, el tamaño y color de la letra, el número de imágenes o la cantidad de palabras, el número y tipos de efectos.

Es muy importante además preparar con tiempo esa reunión, ya que ello permitirá dar una buena imagen y se lograrán mejores resultados. Hay que empezar puntual y terminar de la misma manera, no empezar muy tarde ni excederse del tiempo programado. Existen formatos adecuados para organizar una buena planificación de los tiempos.

## 1.7 Administración de Bases de Datos.

El objetivo del módulo es analizar las principales tareas que debe cubrir el perfil de un administrador de bases de datos, así como la manera de llevarlas a cabo dentro de un RDBMS. Durante este módulo se aprenden comandos y técnicas, y se adquieren habilidades prácticas de administración de bases de datos en el sistema operativo Linux instalado dentro de una máquina virtual que corre a su vez en una plataforma Windows.



### 1.7.1 Tareas del administrador.

Las funciones básicas del administrador de Bases de datos (DBA) son:

- Instalación del RDBMS.
- Creación de las Bases de Datos.
- Definir el tamaño de la base de datos.
- Administración de usuarios.
- Auditoría.
- Respaldo.
- Resguardo de la integridad.

Las prácticas y ejercicios de administración fueron hechos en tres manejadores de Bases de Datos libres: Sybase, Postgresql y MySQL. Antes de administrar una Base de datos es importante conocer la administración del sistema operativo, ya que en ocasiones se deben de crear usuarios a este nivel o verificar si los servicios o procesos están levantados correctamente.

### 1.7.2 Sybase.

#### **Instalación y creación de la base de datos.**

A diferencia de las instalaciones de los RDBMS en los sistemas Windows, en Linux se requieren ejecutar comandos para cada acción, lo que significa sacrificar la sencillez por la seguridad, ya que cuando uno instala en Linux se da cuenta de lo que se está realmente haciendo y de esa manera se puede llevar un mejor control. Después de descargar, instalar Sybase y conocer los comandos para levantar y detener, así como los comandos para acceder a este servidor, es necesario crear dispositivos que alojarán a las Bases de Datos. Para ello se utiliza el comando "disk init" especificando entre otros parámetros el nombre del dispositivo, la ruta en disco y su tamaño expresado en páginas. 1024 Kb (1MB) equivalen a 512 páginas.

El hecho de crear los dispositivos da la oportunidad de controlar dónde se crea la Base. Además, una de las claras ventajas que esto otorga a los administradores es poder separar la base de datos, del log de transacciones ( que es el registro de los INSERT, DELETE y UPDATE que se realizan a la Base), en distintos dispositivos. Además siempre se puede incrementar el tamaño de la Base; no siendo así si se quisiera decrementar, para lo cual se tendría que repetir el proceso de creación.

## Cómo calcular el tamaño de la base de datos.

Saber administrar el espacio que ocupa la base de datos es una práctica que no puede faltar en un buen administrador de Bases de Datos. En las prácticas de este módulo se realiza el proceso del cálculo del tamaño que podría tener nuestra Base. Para ello se debe calcular el tamaño de cada tabla y al final hacer la suma de todas. Para ello se siguen los siguientes pasos:

- Tener al menos un registro completo y saber el tamaño en bytes de éste.
- Tener una estimación de cuántos registros habrá en 5 años.
- Multiplicar los valores anteriores y aumentarle un 25% recomendado como buena práctica, por los índices que pueda tener la tabla.
- Aumentar otro 25% por controles y espaciado.
- Si es una Base altamente transaccional, aumentar 30%, si no lo es, aumentar 15%.
- Redondear.
- Convertir el valor obtenido a MB y multiplicarlo por 512 para obtener el valor en páginas que se necesitan.

Con dicho valor se puede ahora modificar con una sencilla sentencia ALTER DATABASE el tamaño al que se incrementará la base de datos.

Este es el proceso que se debe conocer; sin embargo, con el procedimiento almacenado **sp\_estspace** <nombre\_tabla>, <No\_registros\_estimados> se obtendrá el mismo valor.

## Administración de usuarios.

La administración de usuarios es muy sencilla con los procedimientos almacenados de Sybase. Se pueden agregar usuarios (**sp\_addlogin**), eliminarlos (**sp\_droplogin**), cambiar su contraseña (**sp\_password**), asignarles roles (**sp\_role**), bloquearlos (**sp\_blocklogin**), darles alias (**sp\_addalias**), agregarlos a un grupo (**sp\_changegroup**), ver su perfil (**sp\_displaylogin**), saber los permisos que tiene sobre una tabla u objeto (**sp\_helprotect**), cambiar el dueño de una Base (**sp\_changedbowner**), mostrar todos los usuarios y grupos (**sp\_helpuser** , **sp\_helpgroup**).

Recapitulando, existen consultas SQL con sentencias GRANT para otorgar permisos a los usuarios. Pero con **sp\_role** se puede hacer esta tarea más simple, ya que se le asigna a un usuario permisos predeterminados según el tipo de rol. La tabla 1.7.1 tiene los roles de Sybase y los privilegios que poseen.

Roles de la Base de Datos	
sa (system administrator)	Maneja los discos; crea dispositivos; bloquea y desbloquea logins y los modifica; puede crear, eliminar y modificar BD; modifica la configuración del servidor; mata procesos y hace shutdown al servidor; cambiar dueños de BD.
ss0 (system security officer)	Crea logins; cambia passwords; crea, otorga y revoca roles; maneja el sistema de auditorías; bloquea y desbloquea logins.
oper	Crea y recupera respaldos.

Tabla 1.7.2 Roles de la Base de Datos de Sybase.

## Rendimiento.

Además de la administración de usuarios y todo lo que ello implica, en el módulo se realizan prácticas en las que se modifican parámetros de las bases de datos con el comando **sp\_dboption**, con el cual se adecuan a las necesidades.

Otro elemento importante que interviene en el rendimiento de la base de datos es su log de transacciones, entre más contenga, más lentas serán las respuestas que se obtengan de ella. Por eso es necesario limpiar eventualmente el log con el comando **dumptran**.

Es una tarea fundamental verificar la consistencia de las bases de datos, para ello se realizan distintas pruebas con los comandos **dbcc** que junto con otros comandos revisan que no estén corruptas e incluso puede, en dado caso, borrarlas y al mismo tiempo recuperar los datos respaldados.

## Respaldos.

Sybase tiene métodos muy simples para crear y recuperar respaldos, aún y cuando el proceso de recuperación requiera de más pasos.

El comando **dump database** <nombre\_base> **to** <ruta\_respaldo> respalda tanto la base como su log de transacciones.

El comando **dump tran** <nombre\_base> **to** <ruta\_respaldo> respalda sólo el log de transacciones y lo limpia.

La restauración de un respaldo se hace en caso de que el sistema sufra algún fallo y se dañe la base de datos, siguiendo este procedimiento:

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

### 1.7.3 PostgreSQL.

La siguiente lista muestra las prácticas, ejercicios y pruebas que se hicieron en la administración de las Bases de Datos bajo este RDBMS:

- Instalar el servidor de PostgreSQL.
- Configurar las variables de ambiente.
- Levantar el servidor.
- Listar las bases de datos.
- Crear, modificar y eliminar usuarios.
- Ingresar a otro servidor de la red.
- Crear grupos y agregarles o quitarles usuarios.
- Asignar o revocar privilegios sobre los objetos de la base de datos.
- Creación de bases de datos local y remotamente.
- Tareas de mantenimiento con el archivo postgresql.conf.
- Respaldo con el comando Linux **tar -cvf** <nombre\_respaldo> <base>.
- Respaldo con el comando PostgreSQL **pg\_dump** <base> -f <nombre\_respaldo>.
- Respaldo de todas las tablas con **pg\_dump all** > <nombre\_respaldo>.
- Restauración con **psql** <base\_a\_restaurar> <archivo\_respaldo>.
- Monitorear servidor con el comando Linux **ps -aux|grep ^postgres**.

### 1.7.4 MySQL.

MySQL es si no el más, uno de los más sencillos de manejar; sin embargo fue creado para bases de datos pequeñas, es decir con pocos registros y cuya administración no requiera gran seguridad o no sean altamente transaccionales.

Es por ello que las prácticas sobre este manejador fueron suficientes y efectivas para asegurar una buena administración. Algunas de las actividades hechas fueron:

- Instalar del servidor MySQL.
- Inicio del servidor (requerido por única vez).
- Levantar y dar de baja el servidor (service mysqld start/stop)
- Monitoreo del servidor (mysqld status).
- Administrar usuarios.

- Configuración con el archivo my.cnf.
- Respaldos y restauración con el comando mysqldump.

## **1.8 Buenas prácticas en la función de la Administración.**

Existen organizaciones internacionales que regulan y establecen normas sobre las buenas prácticas de la Tecnología de la Información (TI). El objetivo del módulo es conocer los fundamentos de la auditoría informática y controles, así como las mejores prácticas en el ámbito de la administración de Bases de Datos.

### **1.8.1 Auditoría informática (ISACA).**

Una de estas organizaciones es la Asociación de Auditoría y Control de Sistemas (ISACA™), líder en Auditoría de Sistemas con más de 50,000 miembros en más de 140 países. ISACA tiene un programa de certificación para auditores llamada Certified Information Systems Auditor™ (Auditor Certificado de Sistemas de Información), el cual en su manual de revisión establece las responsabilidades que debe cumplir un Administrador de Bases de Datos:

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

Es imprescindible que la administración se lleve a cabo conforme a los objetivos de control y apoyándose en las mejores prácticas reconocidas mundialmente. Además de someterla a una auditoría, que más allá de ser una cuestión para temer debe considerarse como un proceso en el que se evaluará conforme a estas normas y brindará soluciones adecuadas al entorno de la organización. Las bases de datos deben estar sometidas a controles, ya que las computadoras y los centros de procesamiento de datos son blancos apetecibles para el espionaje, la delincuencia y el terrorismo.

### **1.8.2 Objetivos de Control (COBIT).**

Un objetivo de control es un resultado que se desea alcanzar mediante el control de una actividad. Los controles son medidas que se toman para prevenir, detectar y/o corregir un evento no deseado.

Estos controles se derivan o son presentados en documentos creados por ISACA. COBIT es el documento por excelencia que presenta las mejores prácticas de los objetivos de control. Los Objetivos de Control para la Información y la Tecnología relacionada (COBIT®) brindan buenas prácticas a través de un marco de trabajo de dominios y procesos, y presenta las actividades en una estructura lógica y manejable. Las buenas prácticas de COBIT representan el consenso de los expertos. Están enfocadas fuertemente en el control y menos en la ejecución. Estas prácticas ayudarán a optimizar las inversiones facilitadas por la TI, asegurarán la entrega del servicio y brindarán una medida contra la cual juzgar cuando las cosas no vayan bien. La tabla 1.8.1 muestra que COBIT divide la TI en 34 procesos que se integran en 4 dominios y proporciona un objetivo de control de alto nivel para cada uno. Los objetivos de control que resaltan en negritas son los relacionados con la administración de Bases de Datos.

Los Objetivos de Control para la Información y la Tecnología relacionada	
Dominio (Objetivo de control de alto nivel)	Objetivos de Control
Planear y Organizar	P01 Definir un plan estratégico de TI P02 Definir la arquitectura de la información P03 Determinar la dirección tecnológica P04 Definir los procesos, organización y relaciones de TI P05 Administrar la inversión en TI P06 Comunicar las aspiraciones y la dirección de la gerencia P07 Administrar recursos humanos de TI P08 Administrar la calidad P09 Evaluar y administrar los riesgos de TI P010 Administrar proyectos
Adquirir e Implantar	AI1 Identificar soluciones automatizadas AI2 Adquirir y mantener software aplicativo AI3 Adquirir y mantener infraestructura tecnológica AI4 Facilitar la operación y el uso AI5 Adquirir recursos de TI AI6 Administrar cambios AI7 Instalar y acreditar soluciones y cambios
Entregar y dar Soporte	DS1 Definir y administrar los niveles de servicio DS2 Administrar los servicios de terceros DS3 Administrar el desempeño y la capacidad DS4 Garantizar la continuidad del servicio DS5 Garantizar la seguridad de los sistemas DS6 Identificar y asignar costos DS7 Educar y entrenar a los usuarios DS8 Administrar la mesa de servicio y los incidentes DS9 Administrar la configuración DS10 Administrar los problemas DS11 Administrar los datos DS12 Administrar el ambiente físico DS13 Administrar las operaciones
Monitorear y Evaluar	ME1 Monitorear y evaluar el desempeño de TI ME2 Monitorear y evaluar el control interno ME3 Garantizar el cumplimiento regulatorio ME4 Proporcionar gobierno de TI

Tabla 1.8.1 Objetivos de Control de COBIT

### 1.8.3 Mejores prácticas (ITIL).

ITIL (Biblioteca de Infraestructura de Tecnología de la Información) es un conjunto de documentos que recopilan las mejores prácticas de los mejores organismos del mundo en servicios de TI, tanto públicos como privados.

Las mejores prácticas relacionadas con la Administración de Bases de Datos se enfocan en el diseño, el acceso, la administración, las interfaces y la portabilidad. Y en resumen tratan lo siguiente:

**Diseño:** Deben realizarse el modelado conceptual, el lógico y el físico

**Acceso:** Deben de analizarse los procedimientos almacenados, triggers e índices para minimizar el tiempo de acceso a la Base.

**Administración:** Deben de analizarse y verificarse los niveles de seguridad, los permisos de acceso y los procedimientos de recuperación de desastres y respaldos.

**Interfaces:** Los procedimientos de importación y exportación de la información deben ser verificados con otros sistemas.

**Portabilidad:** Siempre que sea posible debe utilizarse SQL

## 1.9 Seguridad en Bases de Datos.

Al igual que en los módulos anteriores como posteriores, el objetivo de este módulo se cumplió en su totalidad, mismo que consistía en distinguir las principales vulnerabilidades en seguridad en nuestro entorno de trabajo, lo cual se refuerza tomando las medidas necesarias para su corrección y el total aprovechamiento de los elementos y bondades ofrecidos por los RDBMS.

### 1.9.1 Seguridad de la información.

La seguridad es un mecanismo que impide el funcionamiento indeseado. Se dice que un sistema es seguro si éste y su software se comportan de la manera deseada y la información permanece inalterable y accesible. Los objetivos de la seguridad son:

- Confidencialidad
- Integridad
- Autenticidad
- Disponibilidad

Existen cuatro tipos de amenazas de seguridad, de las cuáles se puede ver su descripción en la tabla 1.9.1.

Tipos de amenazas en los Sistemas	
Interrupción	Un activo del sistema se pierde o se vuelve inutilizable.
Intercepción	Alguna parte no autorizada logra acceso a un activo del sistema.
Modificación	Cuando una parte no autorizada accede al sistema y manipula el activo.
Fabricación	Una parte no autorizada puede fabricar objetos falsos en un sistema.

Tabla 1.9.1 Tipos de amenazas en los Sistemas de Información.



Durante este módulo se probaron diferentes vulnerabilidades en el sistema operativo y en la base de datos y se usaron métodos y herramientas para prevenir y corregir dichas vulnerabilidades. Una vulnerabilidad es cualquier debilidad que puede explotarse para causar daño o pérdida al sistema. Y mediante buenas prácticas se aprendió a dotar de seguridad al servidor de Bases de Datos. Una base de datos con un bajo nivel de seguridad compromete no solamente a la base de datos misma, sino también al sistema operativo y a otros sistemas relacionados.

### **1.9.2 Seguridad en Bases de datos.**

Las medidas de seguridad hechas durante el módulo y que deben tomarse en cualquier servidor comprenden la prevención de:

- Liberación inapropiada de información (pérdida de secrecía o confidencialidad).
- Modificación inapropiada de datos (pérdida de integridad).
- Negación de servicio (pérdida de disponibilidad).
- Robo o fraude.
- Accidentes, por ejemplo errores de captura o de programación.
- Uso malicioso de la base de datos.
- Fallas de hardware o software que corrompen datos.

Los requerimientos mínimos necesarios que se deben configurar y hacer cumplir en cualquier RDBM son:

- Autenticación de usuarios
- Protección de acceso impropio
- Integridad de la base de datos
- Administración y protección de datos sensibles
- Registro de eventos y auditoría

### **1.10 Performance and Tunning.**

El Performance es el número de transacciones en un lapso determinado de tiempo que la base de datos puede ejecutar. El Tunning es el ajuste y la modificación de parámetros para optimizar el performance y el rendimiento del servidor.

Para lograr el rendimiento y la optimización deseada en un servidor, se deben configurar como aspectos básicos: la memoria, el disco y el límite de recursos a los usuarios.

Los RDBMS tienen archivos de configuración o en su defecto procedimientos almacenados que permiten elegir entre diversas opciones a las que se les pueden dar valores o cambiar de estado con el fin de realizar la tarea de Tuning.

Cuando se modifica algún valor de esas opciones, en consecuencia cambia la cantidad de memoria que se necesita tener por 'default' para el óptimo trabajo del servidor. Es decir, si se aumenta el valor para alguna opción, se disminuye la cantidad de memoria de este valor. Así que una de las tareas será mantener este valor sin cambios.

Otra de las prácticas desarrolladas durante el módulo es la creación de cachés de memoria exclusivos para tablas, ya que si se tienen muchas tablas y se ejecutan consultas al mismo tiempo y se llena el caché, todas las consultas se estarían peleando por la memoria; por ello se deben apartar ciertas localidades de la memoria para las tablas más importantes, por ejemplo los catálogos, porque tienen información fija y casi todas las consultas las ocupan.

Cuando se hace una consulta, el RDBMS va directamente al disco duro y hace cierta cantidad de lecturas físicas; después las sube al caché y de ahí hace el mismo número de lecturas lógicas. Con la creación de cachés y la asignación de éstos a tablas, esto sólo lo hará la primera vez, y las siguientes sólo va a leer directamente desde el caché. En conclusión, al asociar un caché a una tabla se ahorra tiempo y lecturas.

Cuando hay bases de datos que han sido creadas en dispositivos con capacidad insuficiente corresponde usar técnicas para corregir esta falla. Entre ellas están las de expandirlas a otro dispositivo u otra muy importante, la de creación de segmentos en el disco para la creación de tablas. Los segmentos son espacios de discos exclusivos para el alojamiento de una Tabla. Las Tablas altamente transaccionales son candidatas para ocupar un segmento.

En ocasiones cuando el servidor se vuelve lento (y por ende la respuesta que da al usuario que ejecuta consultas sobre éste) se debe a que muchos más usuarios están accediendo y consultando a la Base. Además de poder definir el número máximo de conexiones simultáneas, se pueden ejecutar procedimientos para restringir ciertos recursos.

## **1.11 Modelado orientado a objetos.**

Se cree que cuando a las Bases de Datos Orientadas a Objetos (BDOO) se les creen estándares que las unifiquen, alcanzarán la fuerza necesaria para reemplazar a las Bases de Datos Relacionales. Se dice que una de las ventajas importantes de las BDOO sobre las BDR es que éstas últimas fueron creadas para el manejo de datos sencillos (números, texto, fechas o booleanos) pero las primeras pueden manejar tipos de datos complejos, definidos por el usuario y no convencionales como las imágenes, los videos o los datos espaciales.

Por ahora no es posible tener acceso tan sencillo a la creación de BDOO o a sistemas manejadores de estas bases. Es raro encontrar organizaciones que las tengan implementadas. Por lo que hay que esperar que estén un poco más popularizadas.

Si hay interés en prepararse para el desarrollo de las BDOO, se puede comenzar profundizando en la programación orientada a objetos y en el modelado orientado a objetos. Uno de los objetivos del módulo es conocer las características de la metodología orientada a objetos, tal como lo es su modelado.

En la actualidad, cualquier compañía dedicada al desarrollo de sistemas exige que se conozca el Lenguaje de Modelado Unificado. Es por ello que los ejercicios y prácticas del módulo fueron dedicados a UML, que en base a un caso práctico y con la herramienta Rational Rose se definieron los casos de uso y se hicieron los diagramas que integran a este lenguaje.

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

UML es un estándar en la industria de construcción de software, que permite modelar y documentar la arquitectura de una aplicación. Este modelado comprende la realización de los siguientes diagramas:

#### **a) Diagrama de casos de uso.**

Los casos de uso son las interacciones que tiene el sistema con un usuario u otro sistema. Los diagramas de caso de uso permiten especificar los requerimientos funcionales, modelan las posibles formas en que un sistema puede ser usado.

#### **b) Diagrama de actividad.**

Describen el comportamiento de un sistema y procesos en paralelo. Es la forma en que el sistema actúa y reacciona a su entorno.

#### **c) Diagrama de estado.**

Los diagramas de estado se usan para mostrar la historia de vida de una clase dada, los eventos que causan una transición de un estado a otro, y las acciones que resultan de un cambio de estado.

#### **d) Diagrama de componentes.**

Un componente es una unidad de código fuente que sirve como bloque constructor para la estructura física de un sistema, tales como los .exe o las .dll.

**e) Diagrama de distribución.**

Los diagramas de distribución son creados para mostrar los diferentes nodos (procesadores y dispositivos) en el sistema. Los nodos modelan la topología del hardware sobre el que se ejecuta el sistema

**f) Diagramas de clases.**

Representan las características estructurales, es decir, los requisitos funcionales de un sistema, los servicios que el sistema debe proporcionar a sus usuarios finales.

**g) Diagrama de secuencia.**

Las interacciones de objetos están ordenadas de acuerdo a la secuencia de tiempo de los mensajes intercambiados.

**h) Diagrama de colaboración.**

Muestran el flujo de datos entre objetos.

### **1.11.2 Proyecto del módulo: Modelado UML del proyecto del Diplomado.**

Es requisito para acreditar el Diplomado de Administración de Bases de Datos, desarrollar un proyecto final, cuyas características se detallaran en el capítulo II. Como proyecto de este módulo se pidió lo siguiente:

- Diagrama de Casos de Uso
- Descripción del caso de uso 'Crear base de datos'
- Diagrama de Clases
- Diagrama de Secuencia de los casos de uso "Autenticarse" y "Crear base de datos"
- Diagrama de Distribución

El diagrama de Casos de Uso es mostrado en la imagen 1.11.1



Imagen 1.11. 1 Diagrama de Casos de Uso.

La tabla 1.11.2 es el formato de la descripción del caso de uso ‘Crear base de datos’.

CASO DE USO	CREAR BASE DE DATOS	
ID. DEL CU	2	ESTADO En elaboración
ACTORES PARTICIPANTES	Usuario o Administrador – Cualquier usuario o el administrador van a poder crear Base de Datos en el servidor MySQL.	
BREVE DESCRIPCIÓN	Permite crear Base de Datos en el servidor respectivo al que se encuentre conectado.	
PRE-CONDICIONES	<p><b>Del proceso:</b></p> <ul style="list-style-type: none"> <li>Los usuarios previamente ya deben de estar dados de alta en el servidor MySQL conectado.</li> </ul> <p><b>Del sistema:</b></p> <ul style="list-style-type: none"> <li>El sistema valida que ya exista el usuario.</li> </ul>	
FLUJO PRINCIPAL	Este caso de uso inicia cuando el Usuario realiza la siguiente acción: <b>Crear Base de Datos</b> – Permite crear Base de Datos por medio de la interfaz del sistema.	
FLUJOS ALTERNOS	<p><b>Crear Base de Datos</b></p> <p>El sistema solicita que se ingrese el nombre con que se va a crear la Base de Datos. El usuario debe ingresar:</p> <p><b>Nombre.</b> Es el nombre de la Base de Datos a crear. Es un dato requerido y el sistema valida que no este duplicado dicho nombre. Ejemplo: Biblioteca</p> <p>Una vez ingresado el dato requerido y al hacer clic en el botón de crear. El sistema mostrara uno de los siguientes mensajes:</p> <ol style="list-style-type: none"> <li>La Base de Datos se ha creado con éxito</li> <li>La Base de Datos ya existe, favor de verificar</li> </ol>	
FLUJOS DE EXCEPCIÓN	<p><b>El nombre es obligatorio</b></p> <p>El sistema notifica: <i>“El nombre es obligatorio. Por favor ingréselo”</i>. El sistema permite ingresarlo.</p>	
POST-CONDICIONES	<ul style="list-style-type: none"> <li>El sistema crea la Base de Datos en el servidor respectivo en donde se encuentre conectado.</li> </ul>	

Tabla 1.11. 1 Descripción de Casos de Uso.

La imagen 1.11.2 es el Diagrama de Clases.

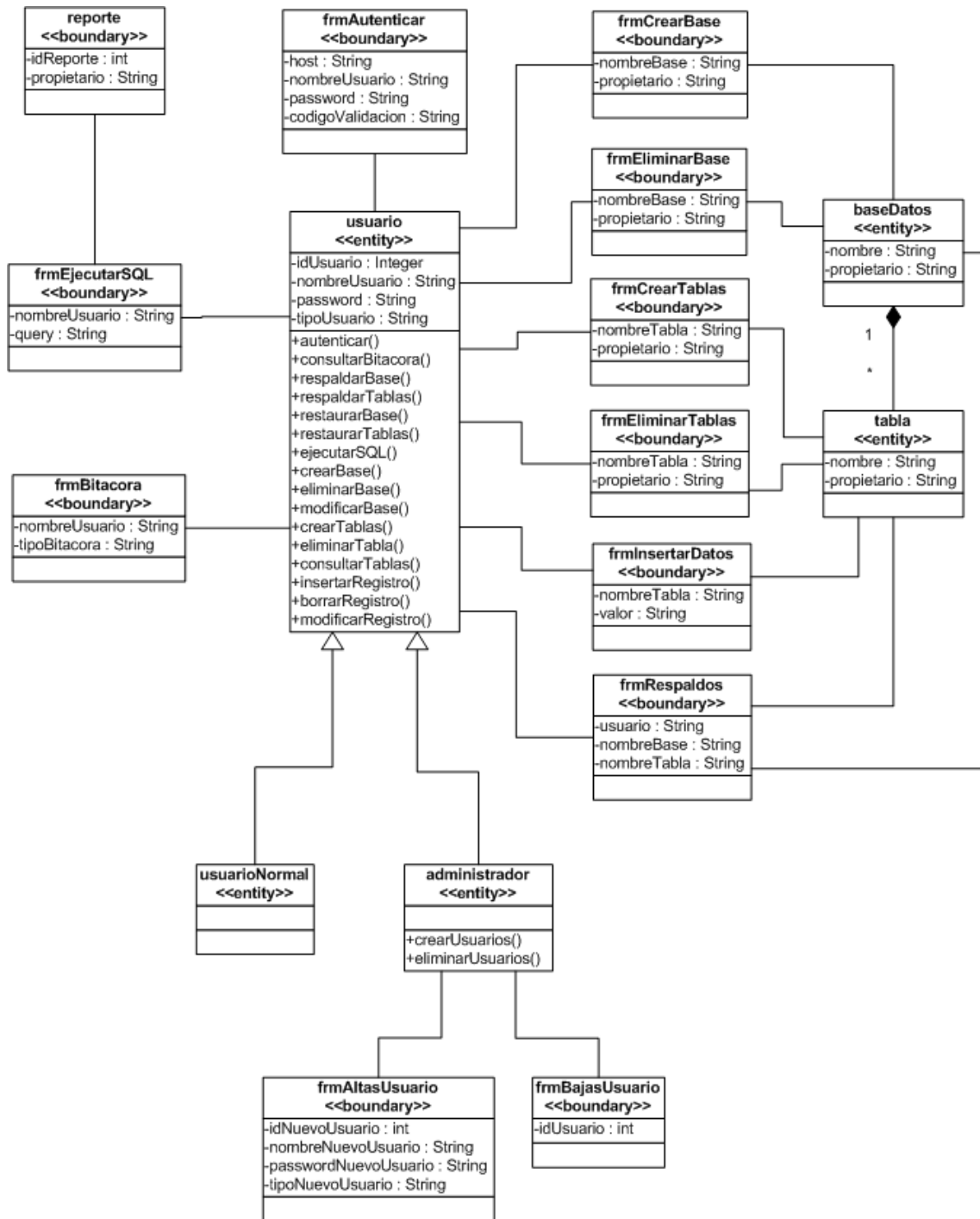


Imagen 1.11. 2 Diagrama de Clases.

Los diagrama de Secuencia de los casos de uso “Autenticarse” y “Crear base de datos” son representados en las imágenes 1.11.3 y 1.11.4, respectivamente.

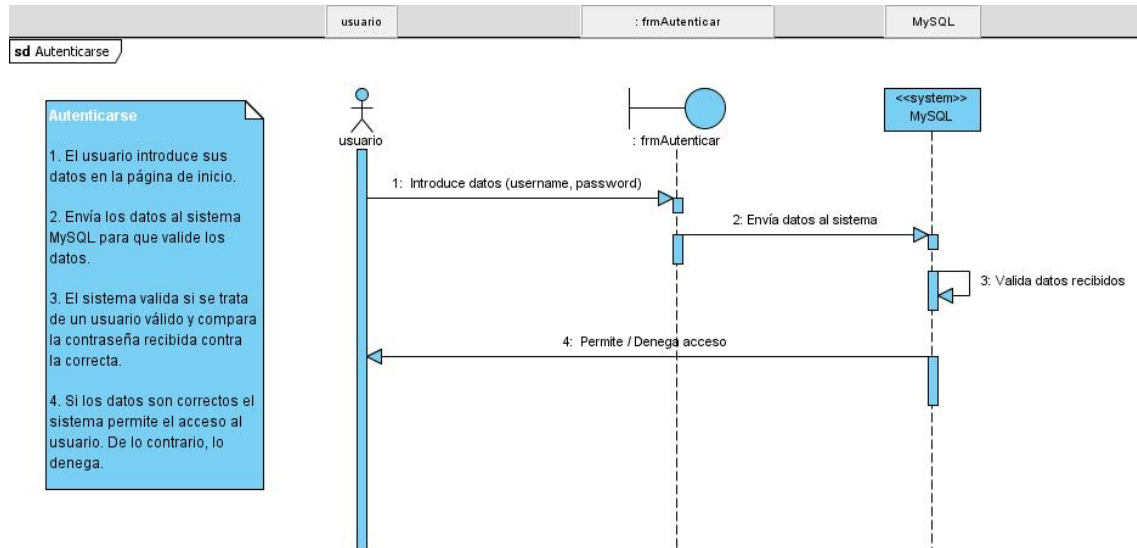


Imagen 1.11. 3 Diagrama de secuencia del caso de uso “Autenticarse”.

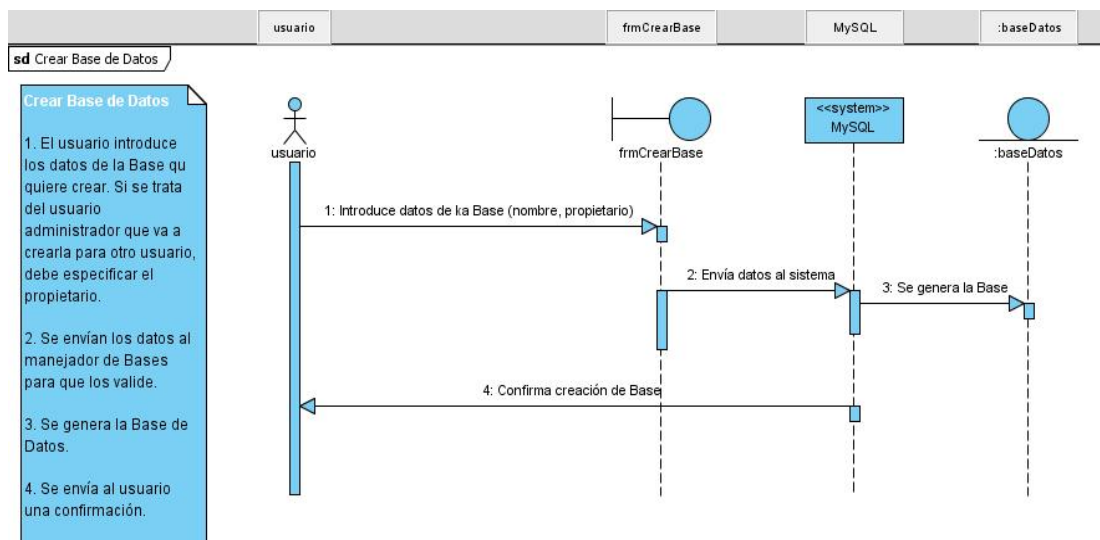


Imagen 1.11. 4 Diagrama de secuencia del caso de uso “Crear Base de Datos”.



En la imagen 1.11.5 se puede ver el diagrama de Distribución.

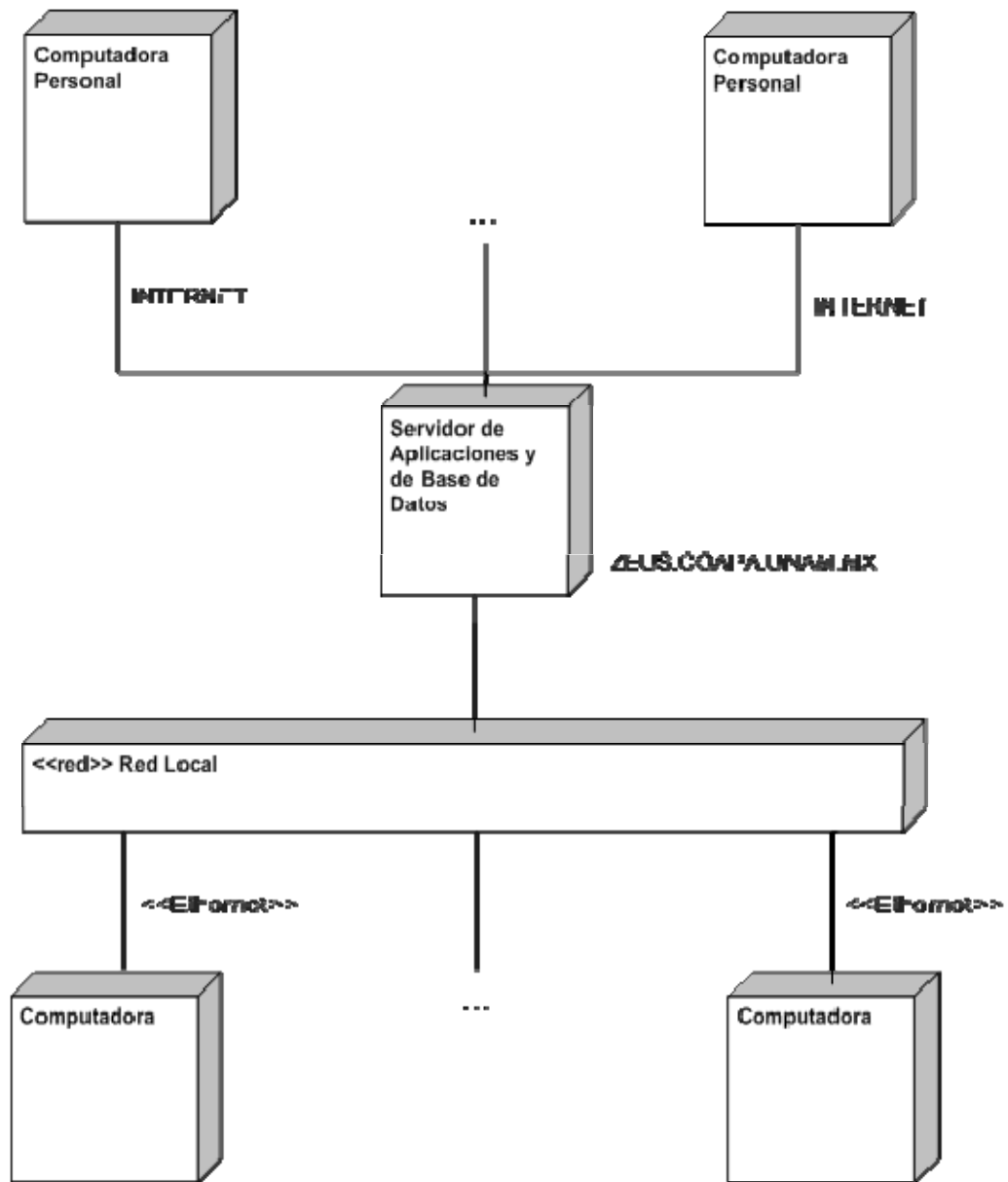


Imagen 1.11. 5 Diagrama de Distribución.

## 1.12 Tópicos avanzados de Bases de Datos.

El objetivo de este último módulo es conocer algunas de las tendencias actuales en el manejo de la información. Estas tendencias son conocidas como Minería de Datos, Data Warehousing y Bases de Datos multidimensionales. Temas bastante interesantes.

El Data Warehousing o Almacén de Datos, permite conjuntar como una sola base de datos toda la información completa de una organización, es decir, puede juntar la información de Access, SQL Server, MySQL, Excel o cualquier otra base de datos con fines comunes para facilitar la toma de decisiones.

Lo que hace la Minería de Datos es aprovechar esta unión de información para no sólo ver detalles de qué está pasando sino en base a eso obtiene conocimiento y crea tendencias, patrones para descubrir y comprender mejor los datos. Se puede decir que con el sólo almacén de datos, se puede analizar la información, mientras que la minería lo hace automáticamente.

Otra forma de explotar el Data Warehousing es el llamado OLAP (Procesamiento Analítico en Línea). Los Almacenes de Datos y las herramientas OLAP integran, transforman y combinan datos para que un usuario o sistema pueda fácilmente analizar información. Esto lo hace con los llamados cubos OLAP, o lo que es lo mismo, Bases de Datos Multidimensionales, las cuales ayudan a facilitar el análisis de los datos porque producen diferentes vistas. Sin embargo, en contraste con la Minería de Datos, OLAP es sólo para dar informes sobre una sola solución.

Con la herramienta OLAPcube trabajada en el módulo se realizan prácticas obteniendo información de Bases de Datos y haciendo todo tipo de consultas y reportes de una manera realmente sencilla y veloz. Si se hubiera querido obtener la misma información con SQL se habría tardado bastante tiempo más.

# II

## PROYECTO DE UN SISTEMA QUE GENERA Y ADMINISTRA BASES DE DATOS.

## 2.1 Características generales.

Uno de los requisitos para acreditar el Diplomado es presentar ante el Comité del mismo, un proyecto final que representa gran parte de lo aprendido en el curso. Se realiza en equipo de tres integrantes y como un detalle extra, el mejor proyecto es elegido para ser presentado el día de la clausura ante autoridades universitarias, compañeros e invitados. Se trata de un sistema que permite generar y administrar bases de datos. Las características básicas del proyecto dadas a esta generación por el Comité, se muestran a continuación en la imagen 2.1.1:

Características del proyecto final Generación 10. (Elección libre del lenguaje de programación y el RDBMS)	
Administrar Bases	<ul style="list-style-type: none"><li>• crear la base</li><li>• borrar la base</li><li>• crear tablas (con los tipos de datos permitidos dependiendo del servidor)</li><li>• permitir insertar datos sobre estas tablas</li></ul>
Administrar usuarios	<ul style="list-style-type: none"><li>• altas</li><li>• bajas</li><li>• bitácoras de acceso por medio de la aplicación</li></ul>
RespalDOS	<ul style="list-style-type: none"><li>• creación</li><li>• recuperación</li></ul>

Tabla 2. 1.1 Características generales del proyecto final.

## 2.2 Características adicionales.

Este proyecto cumple con estos requerimientos y agrega otros detalles propios que lo particularizan. Estas características exclusivas las he dividido en cuatro secciones en las que las detallo y justifico:

### a) Servidor.

Como se mostró en la tabla 2.1.1, el lenguaje de programación y el manejador son de libre elección. Este sistema fue programado en PHP con conexión a servidores de bases de datos de MySQL, ya que la experiencia en estos software es común entre los integrantes del equipo y la funcionalidad que ofrecen juntos es más que comprobable, además de que las características del sistema son adecuadas a este servidor.

## b) Diseño.

Se buscó un equilibrio entre la sencillez, la elegancia y la amigabilidad. Los colores de fondo nos cumplen las primeras dos características, mientras que los íconos coloridos y brillantes nos dan el toque de la tercera. El menú nos da acceso a todas las funciones de la aplicación y es de navegación sencilla.

## c) Seguridad.

Como buena práctica, a todo sistema debe dotársele de seguridad. En este sistema está implantado el llamado HumanCheck, lo cual genera una imagen compuesta con letras y números con la finalidad de comprobar que se trata de un humano y no de un ataque por robots. En la pantalla de acceso, el usuario debe de escribir el código que se genera en la imagen HumanCheck.

## d) Reportes.

El sistema está creado para personas cuya experiencia en las bases de datos no sea necesariamente amplia, sin embargo, se cuenta con una interfaz para aquellas que sí la tengan y conozcan de SQL, con lo cual podrán generar todo tipo de consultas y especialmente reportes. Tanto los reportes generados en esta parte del sistema como los generados en la parte de bitácoras, pueden ser exportados a Microsoft Excel.

## 2.3 Funciones del Sistema.

A continuación, se describen las funciones del sistema mediante pantallas y los mensajes de validación programados en JavaScript:

### a) Formulario de acceso (Imagen 2.3.1).

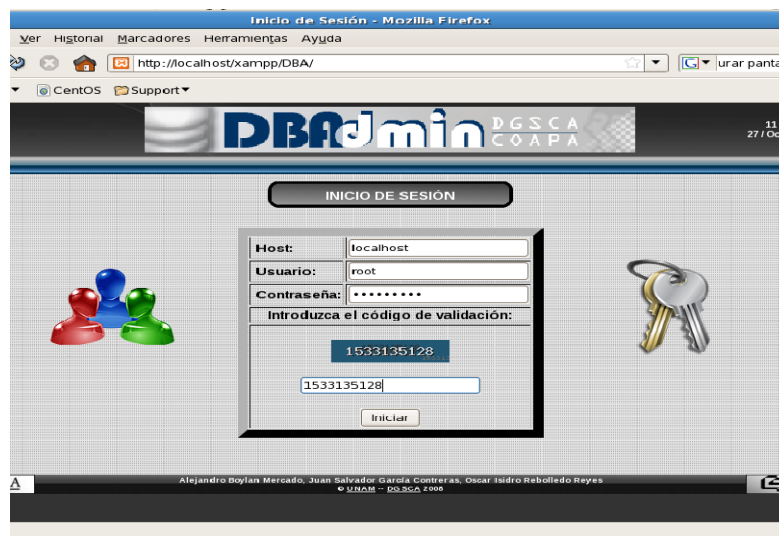


Imagen 2.3.1 Pantalla de acceso al servidor.

El sistema permite que el usuario se conecte no sólo a un servidor, sino a cualquier otro donde se tenga acceso. En la pantalla de inicio, el usuario debe ingresar los siguientes datos:

- Host (IP del servidor al que se quieren conectar)
- Usuario (Nombre de un usuario registrado)
- Contraseña
- Código de validación (Código que aparece en la imagen 'HumanCheck')

Cuando el usuario oprime el botón 'Iniciar' dejando alguno de estos campos vacíos, se le envía una notificación indicándoselo (La Imagen 2.3.2 muestra el caso en que no se coloque contraseña). Si ha escrito algún dato inválido se le envía a otra pantalla señalando los posibles errores por lo que no se pudo establecer la conexión al servidor.

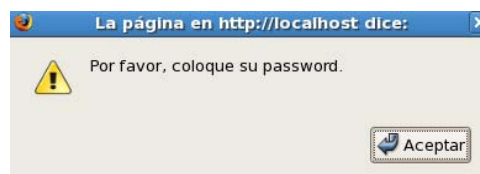


Imagen 2.3.2 Mensaje de contraseña faltante.

#### b) Home (Imagen 2.3.3).

Al establecer una conexión exitosa, el usuario tendrá acceso al menú y obtendrá información del servidor al que está conectado. Entre esa información destaca:

- Versión del servidor MySQL.
- Dirección IP del servidor.
- Dirección IP del cliente.

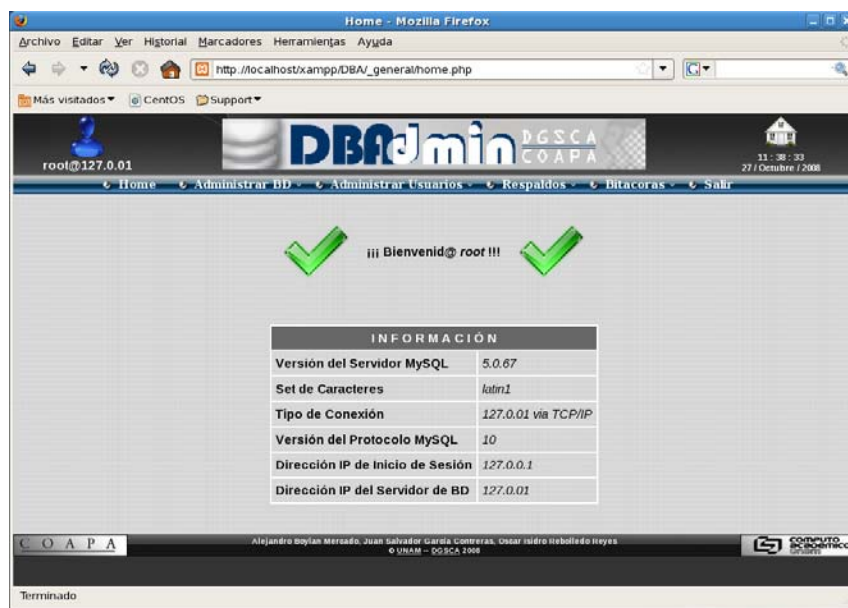


Imagen 2.3.3 Información mostrada en 'Home'

El menú quedará accesible durante la sesión, por lo que siempre será posible volver a Home desde cualquier otra página.

### c) Administrar BD

Como se puede apreciar en las imágenes 2.3.4-5, en 'Administrar BD' se reúnen las opciones para crear y eliminar bases de datos y tablas; así como la manipulación de éstas y la ejecución de consultas SQL.

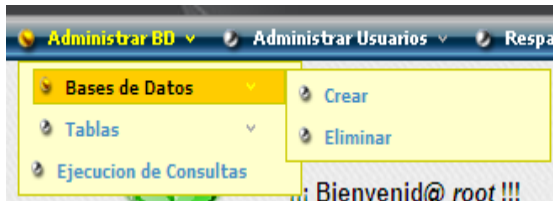


Imagen 2. 3. 4 Menú 'Administrar BD'



Imagen 2. 3. 5 Menú 'Administrar BD'

La imagen 2.3.6 muestra la pantalla en la que el usuario introduce el nombre de la base de datos que quiere crear,

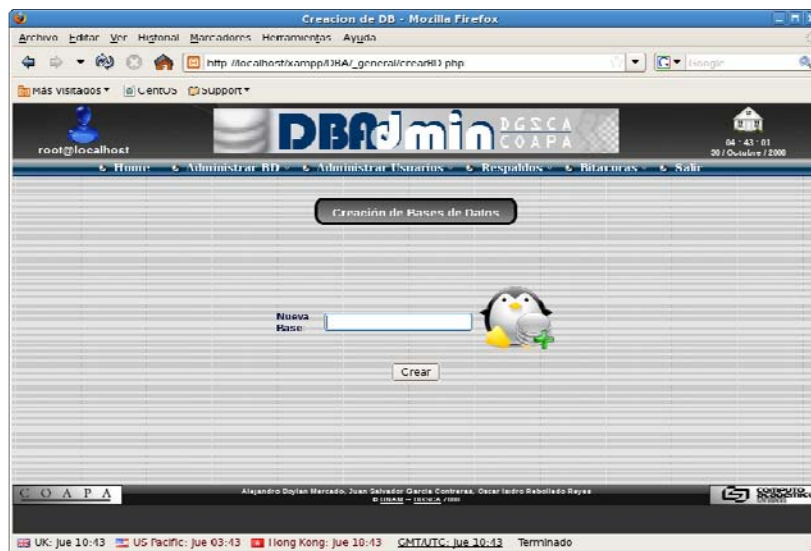


Imagen 2. 3. 6 Creación de Bases de Datos.

Mientras que la imagen 2.3.7 es el mensaje de éxito que aparece cuando ésta se ha creado con éxito, de esa forma se enlistan también todas las existentes, incluyendo la recién creada.



Imagen 2.3.7 Confirmación de Base creada.

Para eliminar una base de datos, se necesita ingresar a la opción correspondiente y elegirla de entre la lista desplegable, como se ve en la imagen 2.3.8



Imagen 2.3.8 Eliminación de Bases de Datos.

Al presionar el botón 'Eliminar', el usuario recibe un mensaje en el que se le pide confirmar (Imagen 2.3.9), si acepta se le notificará que ha sido eliminada y lo podrá constatar en la lista de bases de datos disponibles.



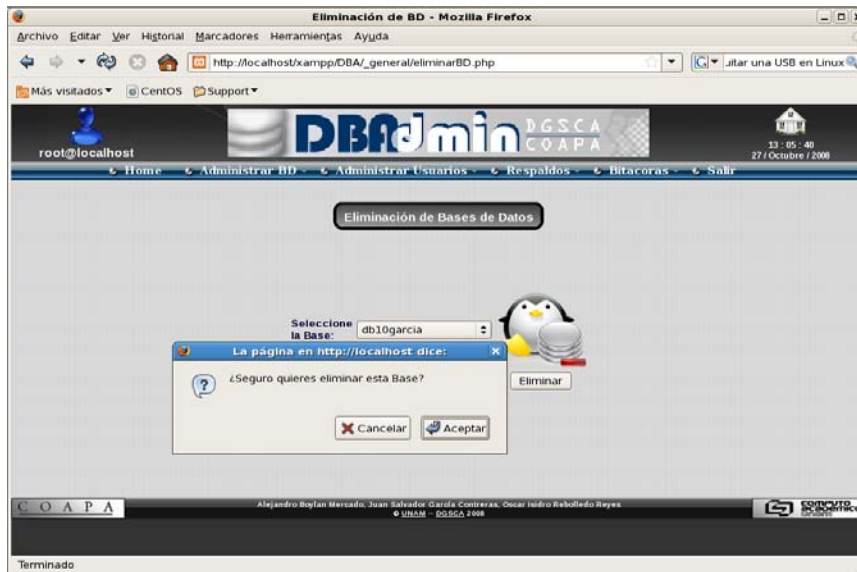


Imagen 2.3.9 Petición de confirmación de eliminación de una BD.

Sin embargo, existen tablas propias de MySQL que no se pueden eliminar porque el servidor se vendría abajo (Imagen 2.3.10).



Imagen 2.3.10 No se pueden eliminar bases de datos de Sistema.

Al contar con una base de datos, el sistema permite crear tablas de una forma muy sencilla y de igual forma eliminarlas. Para crear, debe ingresarse al menú correspondiente y elegir en qué base de datos se hará la acción (imagen 2.3.11).



Imagen 2.3.11 Creación de Tablas.

Se requiere el nombre de la nueva tabla y el número de campos que tendrá. En esta pantalla (como lo muestra la imagen 2.3.12), el usuario puede ver las tablas que ya existen en esa base de datos, en caso de que así sea.



Imagen 2.3.12 Nombre de Tabla y número de campos.

Si se deja uno de los dos campos en blanco o ha introducido el valor 0 u otro caracter no numérico en el número de campos, el usuario recibirá un mensaje indicándoselo (La imagen 2.3.13 muestra el caso en el que no se haya escrito el nombre de la tabla).

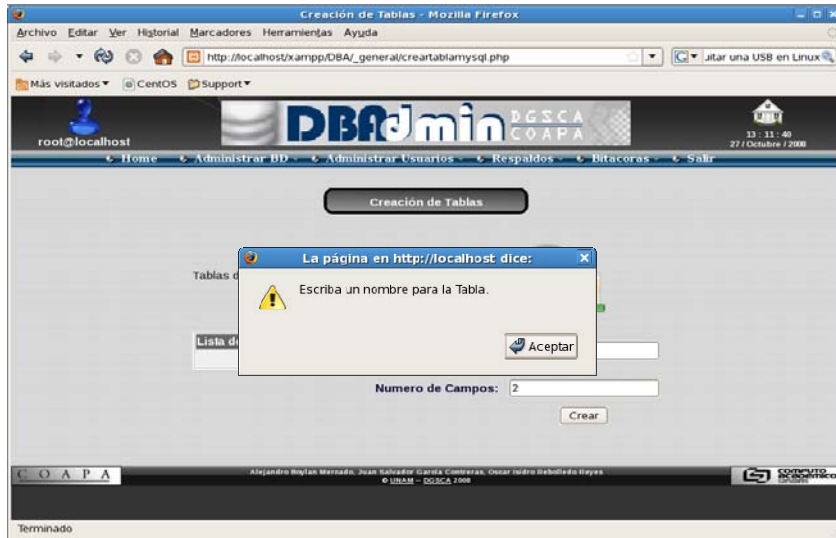


Imagen 2.3.13 Notificación de nombre de tabla faltante.

El siguiente paso es proporcionar el nombre de cada campo, elegirles un tipo de dato, darles una longitud (algunos tipo de dato no tienen longitud) y elegir si el campo puede contener valores nulos o no (Imagen 2.3.14).

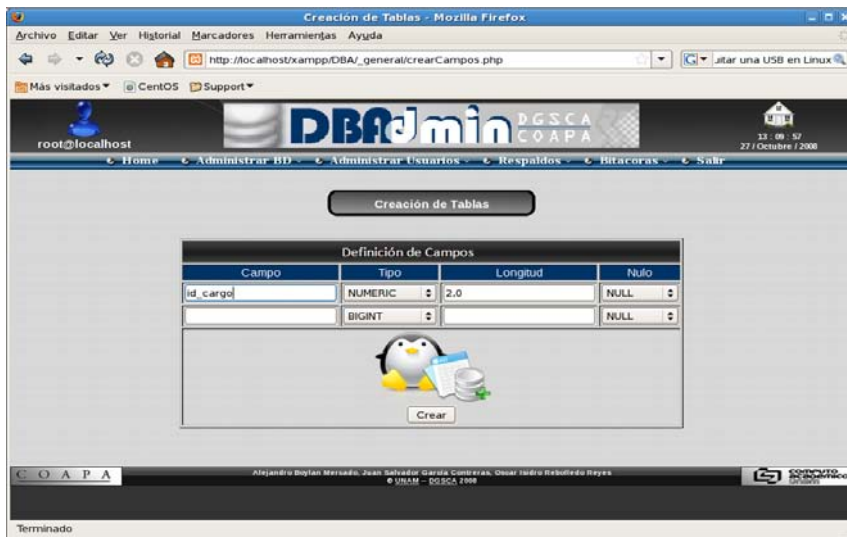


Imagen 2.3.14 Definición de campos.

Si todo ha salido bien, la siguiente pantalla (Imagen 2.3.15) muestra un mensaje de confirmación, la lista de tablas existentes y además la lista de los campos creados con la opción de crear una llave primaria con los campos que seleccione.



Imagen 2.3.15 Confirmación de tabla creada.

Al dar clic en el botón “Crear PK” el sistema verifica que se haya elegido al menos un campo, si no fue así se le notificará que debe hacerlo. Si eligió uno o más campos, la siguiente pantalla le mostrará un mensaje de verificación de llave primaria creada (Imagen 2.3.16).



Imagen 2.3.16 Confirmación de PK creada.

Eliminar una tabla requiere de menos pasos. El usuario elige la base de datos de donde quiere eliminar la tabla y lo puede hacer previa petición de confirmación, tal como se aprecia en la Imagen 2.3.17

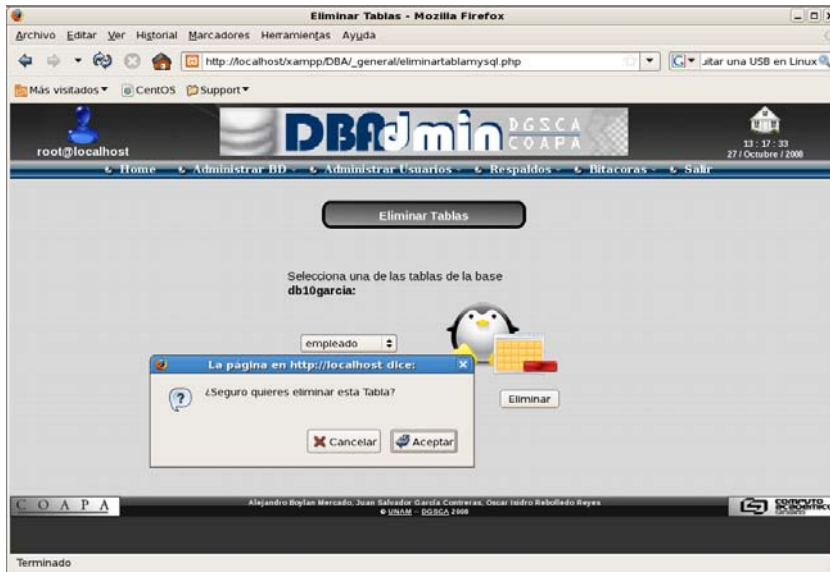


Imagen 2.3.17 Petición de confirmación de eliminación de una Tabla.

Si la operación fue exitosa recibe un mensaje indicándoselo (Imagen 2.3.18).



Imagen 2.3.18 Confirmación de tabla eliminada.

La imagen 2.3.19 muestra cómo si el usuario elige alguna base de datos de Sistema, no se le permitirá acceder.

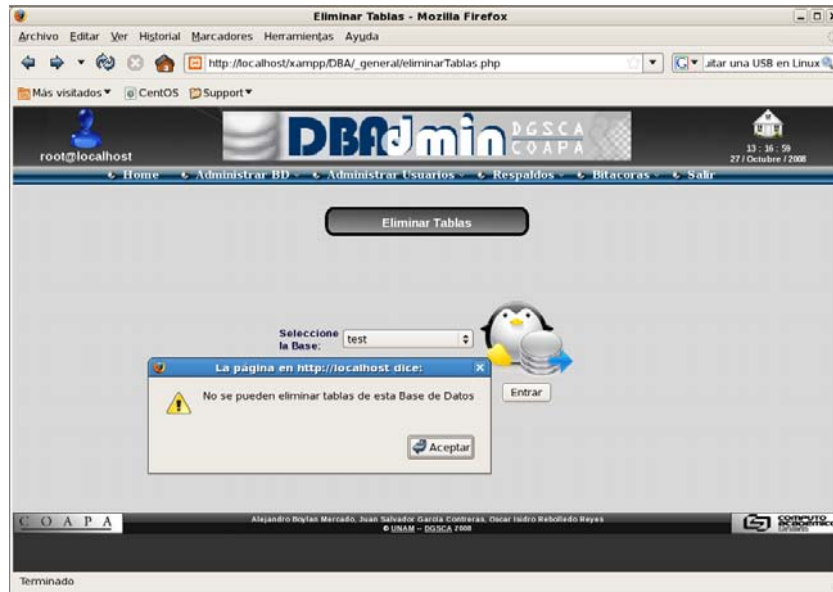


Imagen 2.3.19 No se pueden eliminar tablas de Sistema.

Una vez creadas la base de datos y la tabla, el usuario puede insertar datos accediendo a “Examinar o modificar” y eligiendo la base de datos donde está la tabla que quiere modificar (Imagen 2.3.20).



Imagen 2.3.20 Examinar o modificar tablas.

Al elegir una base de datos, al usuario se le muestran las tablas de dicha base y junto con éstas, íconos con los que puede examinarlas, insertarles o borrarles datos y eliminarlas, además que se pueden ver cuántos registros tiene cada una (Imagen 2.3.21).

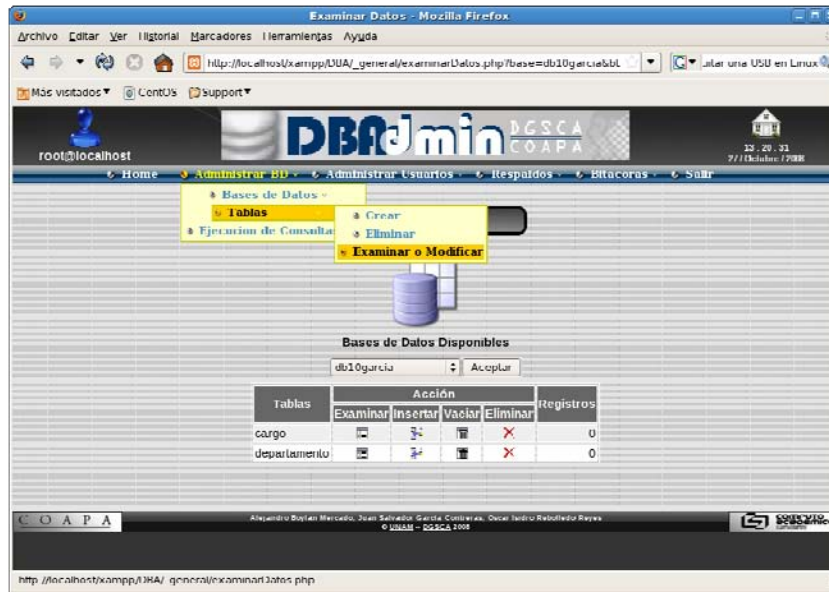


Imagen 2.3.21 Opciones de modificación de Tablas.

Al Insertar o modificar datos se presenta una pantalla como la mostrada en la imagen 2.3.22, en la que el usuario puede escribir los valores deseados en los campos correspondientes.

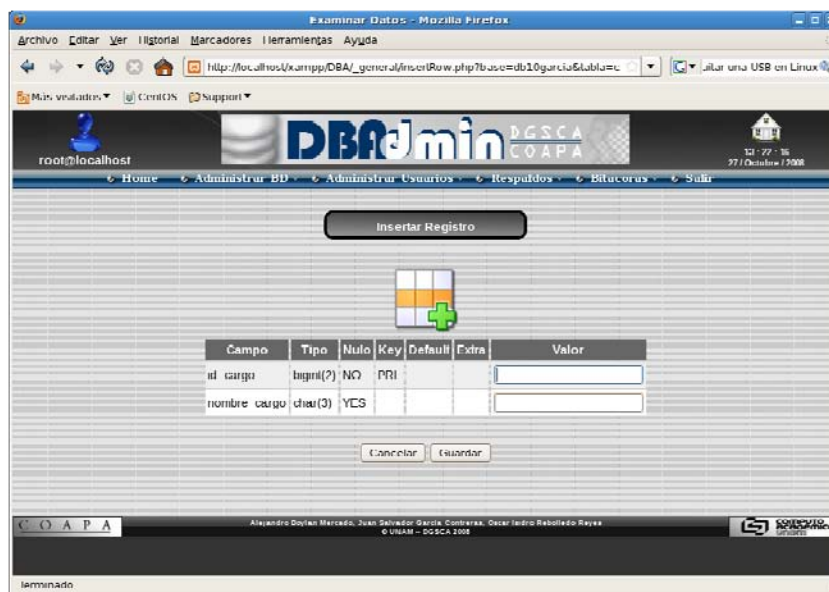


Imagen 2.3.22 Inserción de registros.

La opción de examinar muestra una pantalla como la que se muestra en la imagen 2.3.23. En ella se ven listados todos los registros y se pueden exportar a Microsoft Excel.

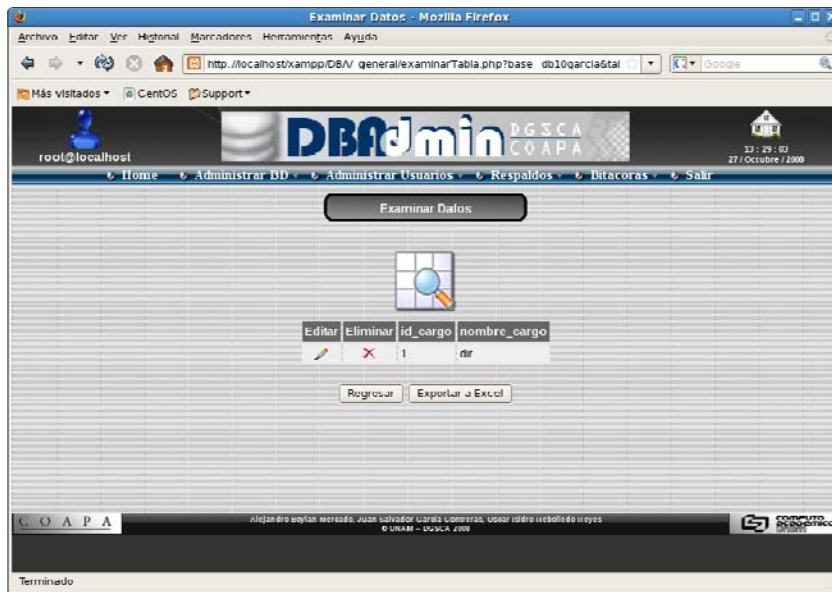


Imagen 2.3.23 Listar registros de una Tabla.

La última parte de este menú 'Administrar BD', incluye la interfaz en la que un usuario con conocimientos de SQL puede ejecutar consultas con el objetivo de obtener reportes y exportarlos a Excel para una manipulación más sencilla de esta información (Imagen 2.3.24). Para ello se debe elegir la base de datos en la cual se quiere ejecutar la consulta, como en anteriores procedimientos.

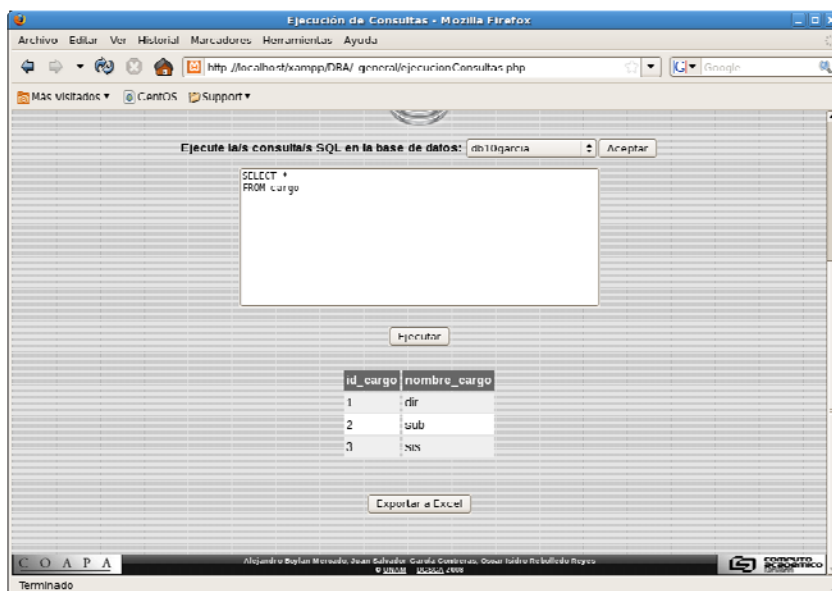


Imagen 2.3.24 Interfaz para ejecución de SQL.



#### d) Administrar Usuarios.

Como se muestra en la imagen 2.3.25, en este menú, el usuario root puede crear, eliminar, consultar y modificar cuentas de usuario.



Imagen 2.3.25 Menú 'Administrar Usuarios'.

Para crear una cuenta de usuario se ingresa el nombre de éste, la contraseña dos veces para confirmar y se indica si este usuario puede conectarse de cualquier servidor o sólo localmente (Imagen 2.3.26).

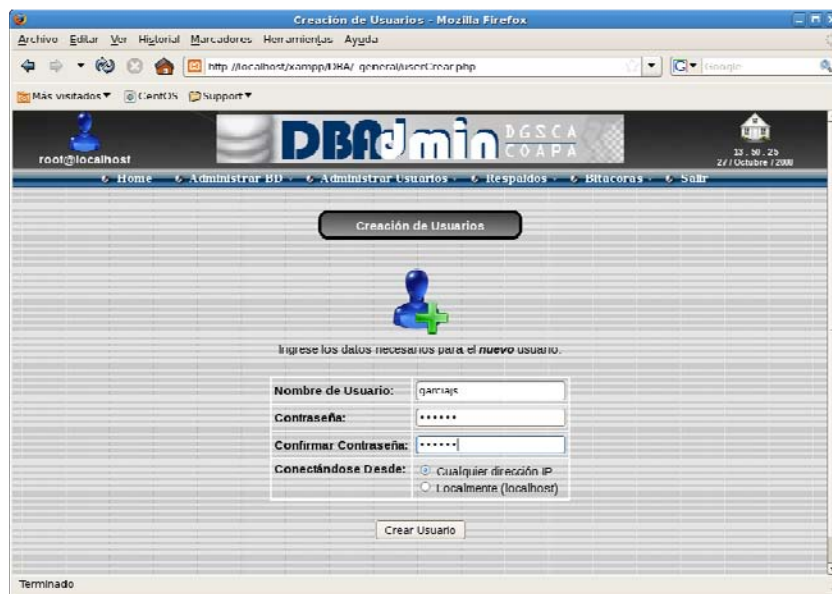


Imagen 2.3.26 Creación de usuarios.

Al crearlo, se podrá ver un mensaje de confirmación. Si se trata de eliminar una cuenta, es cuestión de elegirla de una lista de usuarios y aceptar la eliminación (Imagen 2.3.27), lo cual también mostrará un mensaje confirmando la acción (2.3.28).



Imagen 2.3.27 Eliminación de usuarios.



Imagen 2.3.28 Confirmación de eliminación de usuarios.

Para consultar y modificar la información se requiere también de elegirlo de una lista y realizar la acción correspondiente. La consulta muestra básicamente la misma información que se da al crear la cuenta (2.3.29).

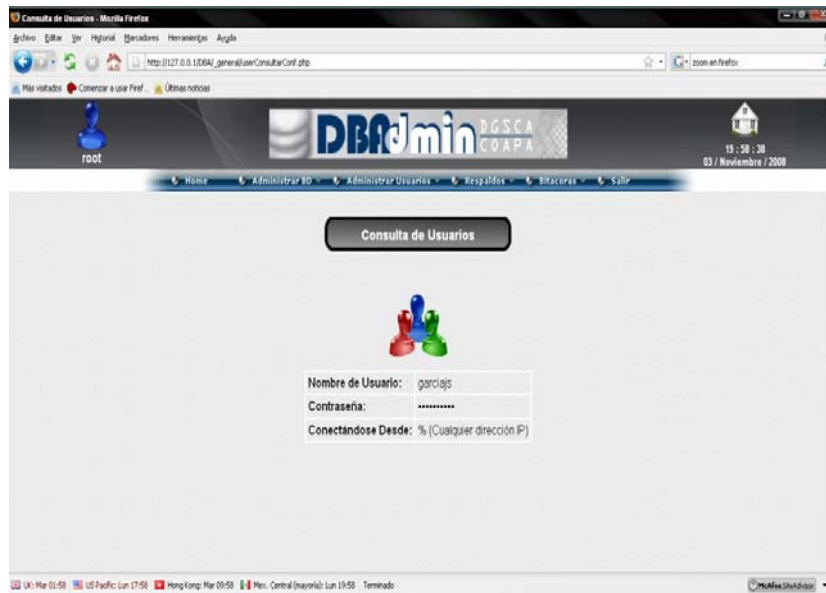


Imagen 2.3.29 Consulta de usuarios.

La modificación no permite cambiar el nombre de usuario, tan sólo la contraseña y si se conecta local o remotamente (2.3.30).

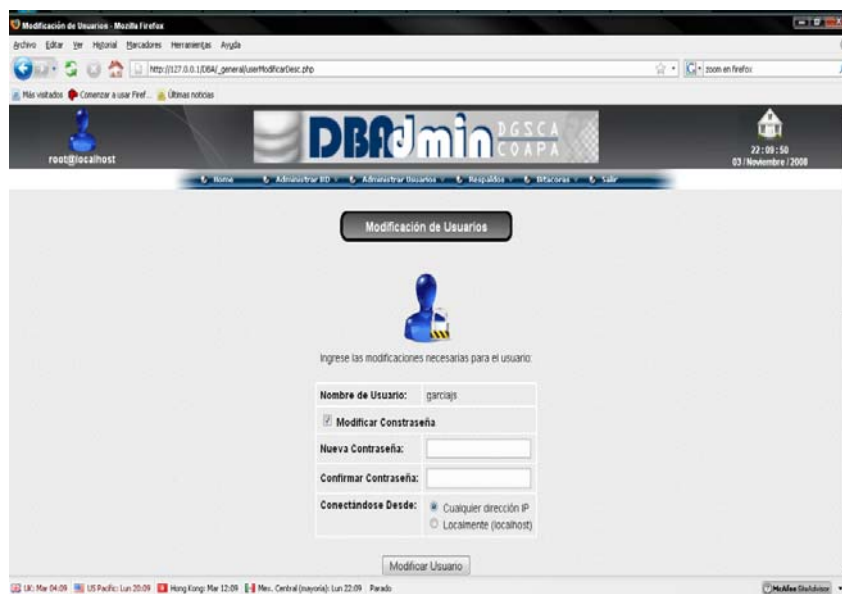


Imagen 2.3.30 Cambio de contraseña.

#### e) Respaldos.

La interfaz de respaldos permite al usuario elegir entre hacer un respaldo por cada base de datos o uno de todas (Imagen 2.3.31). Al elegir hacer un respaldo de todas las bases, se hace por consecuencia, el respaldo de todas las tablas de todas las bases de datos (Imagen 2.3.32).

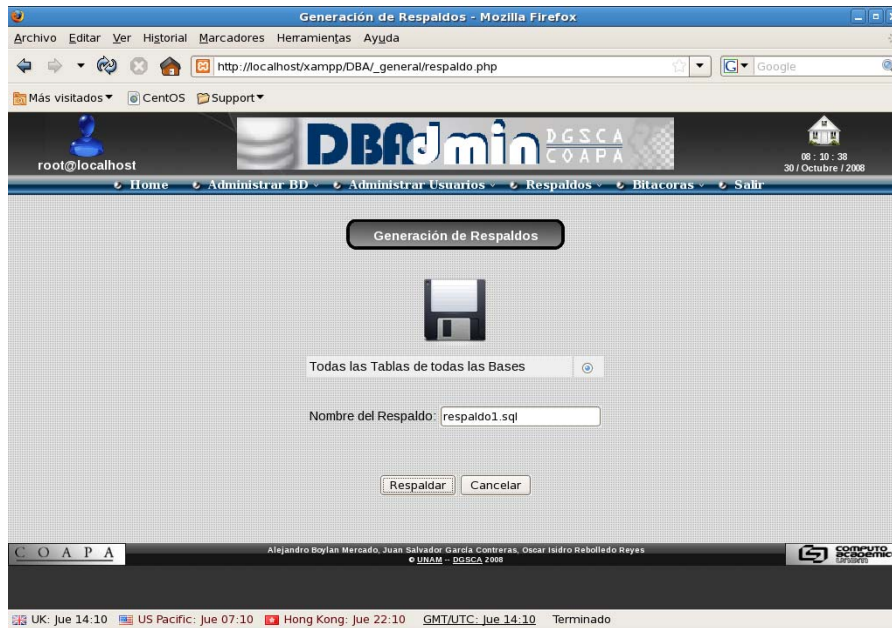


Imagen 2.3.31 Respaldos.

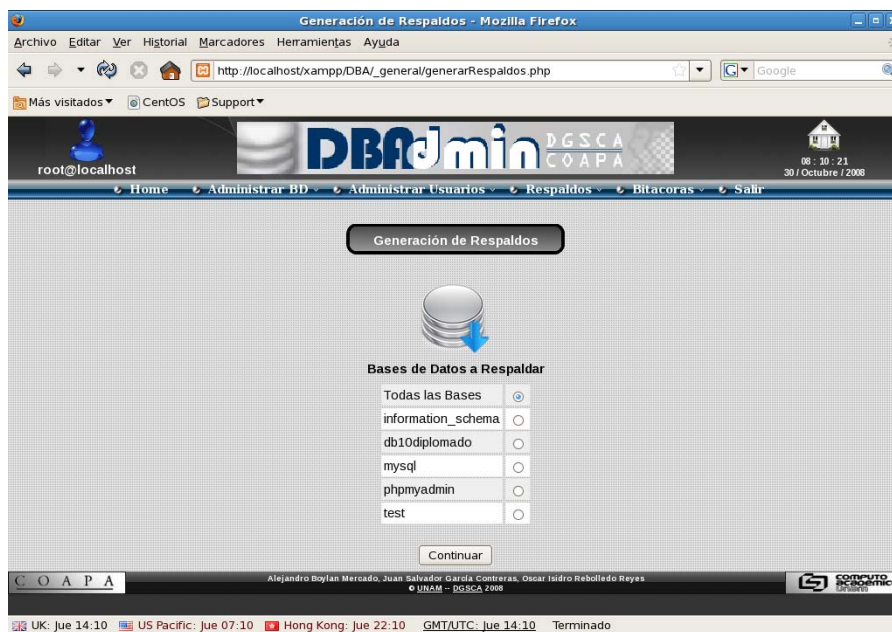


Imagen 2.3.32 Respaldo de todas las Tablas de todas las Bases.

Si se elige respaldar una sola , la siguiente pantalla mostrará la lista de sus tablas en las que se puede elegir entre respaldar todas las tablas o sólo una en específico, como se ve en la imagen 2.3.33.

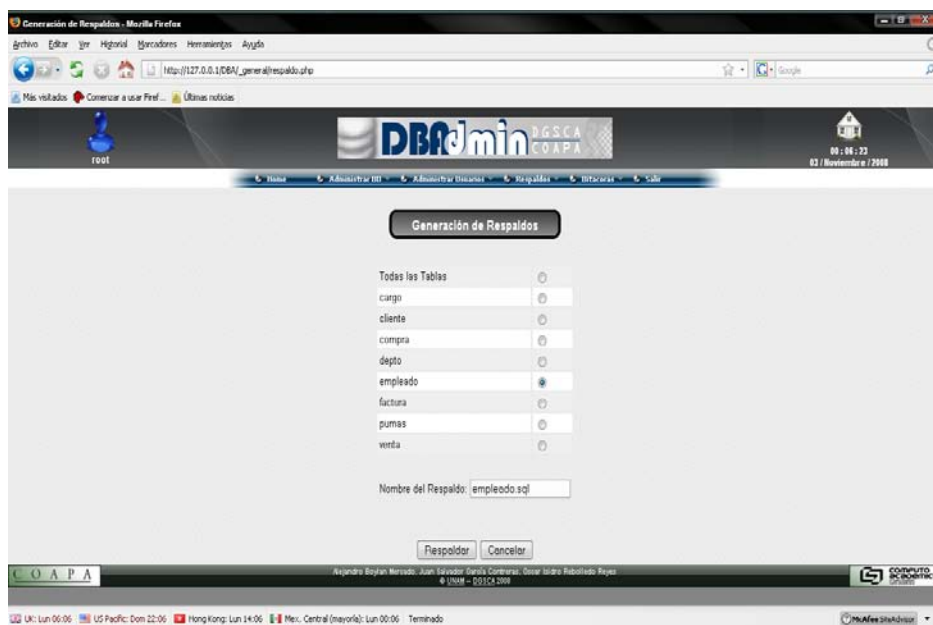


Imagen 2.3.33 Respaldos de Tablas.

Cualquiera que sea la opción elegida, el usuario da un nombre al respaldo y se le permite elegir un destino en donde guardar el archivo de respaldo (Imagen 2.3.34).

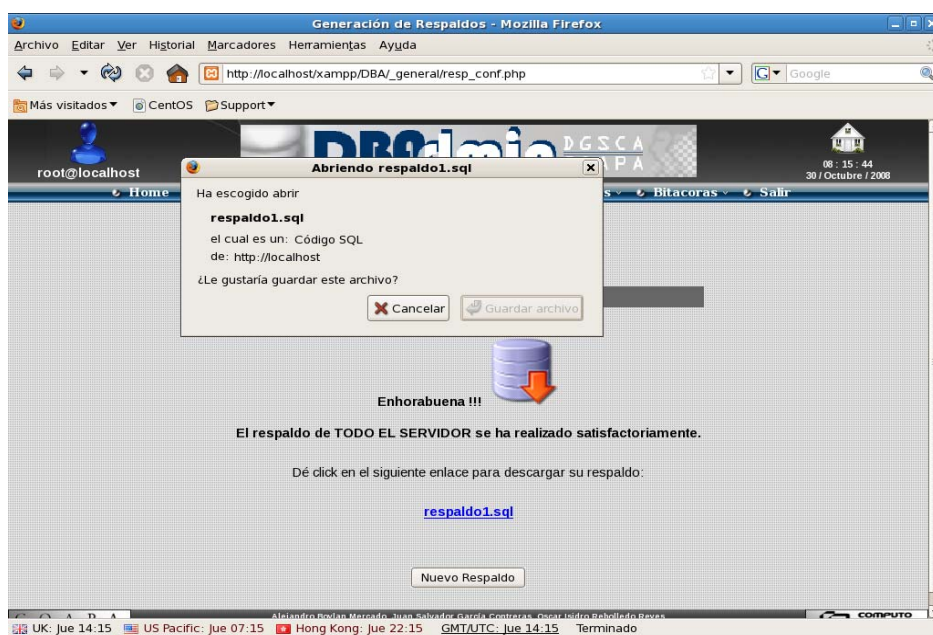


Imagen 2.3.34 Guardar archivo de respaldo.

Además de respaldar las bases de datos y las tablas, el usuario también puede hacerlo con las bitácoras de acceso. En la imagen 2.3.35 se ve cómo es la interfaz que permite elegir el nombre de usuario para respaldar la bitácora de éste.

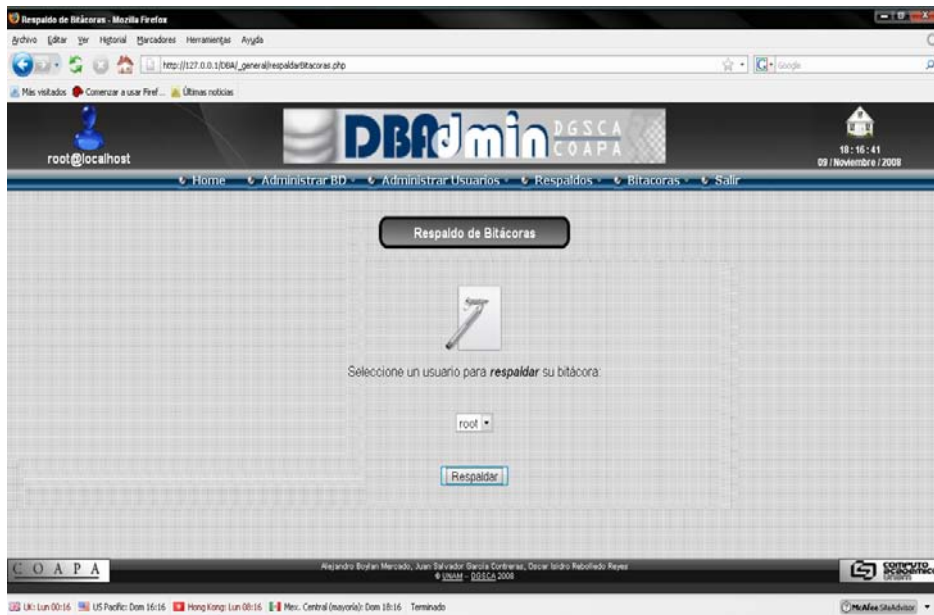


Imagen 2.3.35 Respaldo de Bitácoras.

Por supuesto que una buena práctica de respaldos va a acompañada de la restauración de los mismos. El sistema da la opción de restaurar todo el servidor o cargar un respaldo de una sola base o una tabla. Estas opciones las apreciamos en la imagen 2.3.36

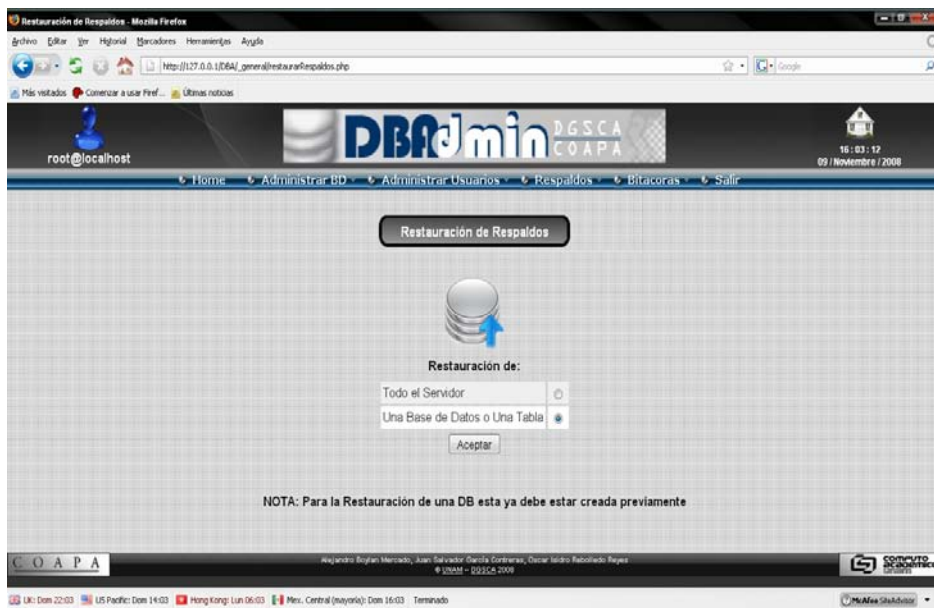


Imagen 2.3.36 Restauración de Respaldos.

Así como se permite elegir el lugar donde respaldar, la sencillez y amigabilidad del sistema da deja buscar el archivo de respaldo para cargarlo (Imagen 2.3.37).

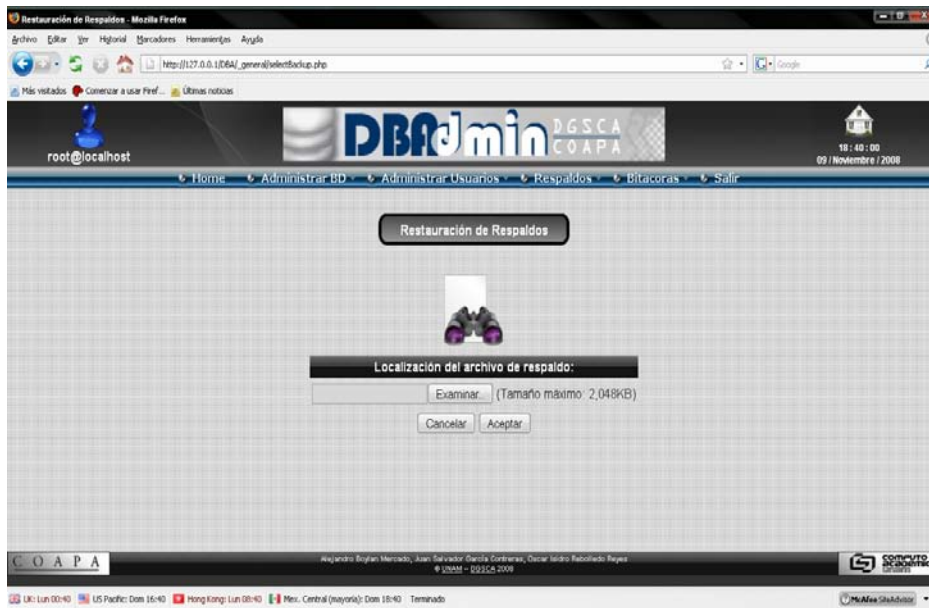


Imagen 2.3.37 Ubicación de archivo de respaldo.

#### f) Bitácoras.

Otra herramienta de administración propia del root, es la consulta de bitácoras de acceso de los usuarios que ingresan a través de esta aplicación a algún servidor. En este menú se cuenta con dos opciones, consultarlas y eliminarlas. La imagen 2.3.38 es una muestra de la interfaz de consulta. Puede elegir entre ver los accesos exitosos, no exitosos o ambos.

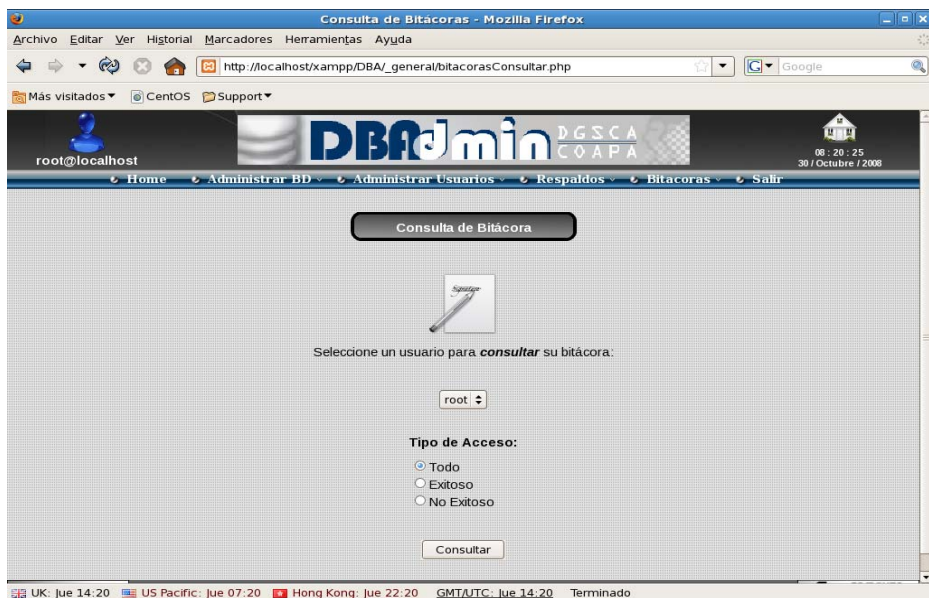


Imagen 2.3.38 Consulta de Bitácoras.

Este reporte también se puede exportar a Excel, como se ve en la imagen 2.3.39.

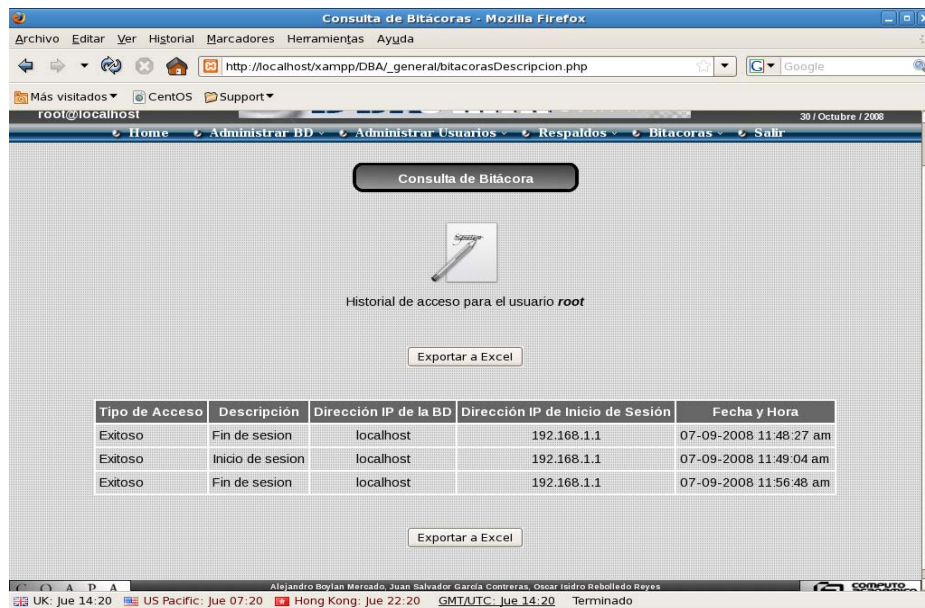


Imagen 2.3.39 Historial de acceso de usuarios.

Además de consultarlas, el usuario root puede eliminarlas (Imagen 2.3.40).

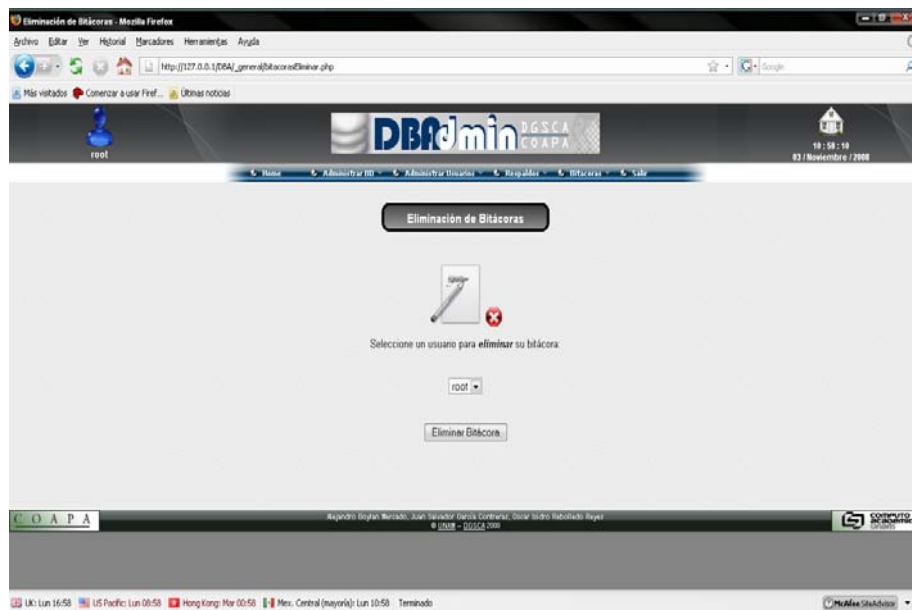


Imagen 2.3.40 Eliminación de bitácoras.



## **CONCLUSIONES**

La calidad de la enseñanza de la Universidad es innegable y la reafirma con cada clase, taller o curso en cualquiera de sus modalidades. Esta ocasión no podía ser la excepción y el Diplomado de Administración de Bases de Datos ha rebasado las expectativas y ha cumplido los objetivos planteados, de una forma extraordinaria.

Las técnicas, estrategias y habilidades desarrolladas y los conocimientos adquiridos hacen que cada uno de los participantes cumpla con lo suficiente y más de lo que las organizaciones requieren de cada profesionalista.

La administración de bases de datos requiere de mucha práctica, mucha concentración y de conocimientos sólidos en aspectos fundamentales, entre otros tantos aspectos. Como lo pudimos ver, existe un módulo con el mismo nombre del Diplomado; sin embargo, éste no sería nada sin los otros once. Un buen administrador no lo sería si no supiera cómo se diseña o cómo se abstrae la información necesaria para hacerlo. No se puede manejar un RDBMS a nivel administrador, cuando no se conoce a nivel usuario. No se puede ser un buen administrador si no se supieran automatizar ciertas tareas de administración a través de la programación de aplicaciones. Y ¿Cómo desarrollar estas aplicaciones sin conocer y practicar SQL? ¿Cómo poder decir que un administrador lo es, sino supiera los fundamentos de seguridad o de optimización del rendimiento? Un buen administrador de bases de datos, técnicamente estaría completo si tuviese las características mencionadas, pero no podemos olvidar la parte humana ni la parte normativa; aspectos sumamente importantes para desarrollarnos profesionalmente. La comunicación con el equipo de trabajo es la llave de muchas puertas de éxito en la organización y en lo personal. Conocer los estándares y las normas le dan credibilidad y confianza a nuestra labor. Estar actualizados y conocer las tendencias es una tarea que no debe tener fin; en cuestión de tecnología hay que mantenerse al tanto de sus avances.

Sin duda alguna, el Diplomado está estructurado y planificado en base a las necesidades críticas relacionadas con las bases de datos actuales.