



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

---

---

**FACULTAD DE INGENIERÍA**

**DISEÑO DE UN SISTEMA PARA LA  
ADMINISTRACIÓN DEL PERSONAL DE LA  
DIVISIÓN DE INGENIERÍA EN CIENCIAS DE  
LA TIERRA DICT**

**T E S I S**

PARA OBTENER EL GRADO DE  
INGENIERO EN COMPUTACIÓN  
PRESENTA:

**ROMERO DE LOS SANTOS VANESSA JAZMÍN**



**DIRECTOR DE TESIS: M. EN C. GABRIEL CASTILLO HERNÁNDEZ**

**CIUDAD UNIVERSITARIA**

**2011**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

*A MI MAMI!!!!!!*

GRAX!!! a todas las prsonas q directa o indirectamnt participaron en la realizaci3n d est  
proycto

# ÍNDICE

<i>INTRODUCCIÓN</i> -----	<b>1</b>
<i>CAPÍTULO 1</i> -----	<b>4</b>
1.1 Bases de Datos-----	<b>4</b>
1.1.1 Definiciones y conceptos de las bases de datos-----	4
1.1.2 Sistema manejador de bases de datos (DBMS)-----	6
1.1.3 <i>Modelo de datos</i> -----	12
1.2 Metodología para el desarrollo de sistemas-----	<b>24</b>
1.2.1 Definición de Metodología-----	24
1.2.2 Planeación-----	25
1.2.3 Análisis-----	26
1.2.4 Diseño-----	27
1.2.5 Desarrollo-----	28
1.3 Sistemas de Información vía Internet-----	<b>29</b>
1.3.1 Internet-----	30
1.3.2 Arquitectura Cliente Servidor-----	31
1.3.3 Servidores de Páginas Web-----	32
1.3.4 Protocolos de comunicación-----	33
1.3.5 World Wide Web-----	36
1.3.6 Lenguajes de Programación para web-----	38
1.4 Sistema Operativo Windows-----	<b>42</b>
1.3.7 Seguridad de Windows-----	43
1.3.8 Windows Server 2003.-----	43
<i>CAPÍTULO 2</i> -----	<b>49</b>
2.1 Análisis-----	<b>49</b>
2.1.1 Delimitación del Sistema-----	50
2.2 Diseño-----	<b>53</b>
2.2.1 Propuesta de Software-----	53
2.2.2 Diagrama jerárquico funcional del sistema-----	54
2.2.3 Diseño de la Base de Datos-----	56

<i>CAPÍTULO 3</i> -----	<b>65</b>
<b>3.1 Introducción</b> -----	<b>65</b>
3.1.1 Especificaciones del hardware-----	66
3.1.2 Seguridad del servidor-----	67
<b>3.2 Creación de la Base de Datos</b> -----	<b>67</b>
<b>3.3 Creación de la interfaz de usuario</b> -----	<b>69</b>
3.3.1 Tipos de Usuario-----	74
3.3.2 Tipos de Consulta-----	75
3.3.3 Alta y Baja del Personal-----	78
3.3.4 Inserción y Modificación del Personal-----	82
<b>3.4 Expectativas del sistema</b> -----	<b>86</b>
 <i>CAPÍTULO 4</i> -----	 <b>92</b>
 <i>REFERENCIAS Y BIBLIOGRAFÍA</i> -----	 <b>95</b>

# *INTRODUCCIÓN*



La Facultad de Ingeniería de la UNAM esta conformada por divisiones que engloban las diversas carreras que en ella se imparten. Una de las divisiones que mayor conflicto presenta, tanto para los alumnos como para los académicos y administrativos, en la localización y ubicación del personal docente, es la División de Ingeniería en Ciencias de la Tierra (DICT). Esto debido a que una gran parte de su personal docente no tiene cubículos dentro de la misma Facultad, o bien porque tienen otras actividades fuera de la misma.

Por otra parte, los sistemas de hoy en día requieren de niveles de seguridad más fortalecidos, esto se debe a que día a día el área de Tecnologías de la Información va creciendo, evolucionando y mejorando con el objetivo de garantizar que la información contenida dentro de las aplicaciones, se utilice correctamente.

Por tal motivo se ofreció a la DICT el diseño y desarrollo de un directorio de su personal docente el cual dentro de su sencillez ofrezca un nivel de seguridad alto.

Durante esta tesis se diseñó y desarrolló el Sistema Integral para la Gestión de Personal Docente, el cual tiene por objetivo principal la consulta y administración, vía Internet, de forma sencilla, segura y confiable de los datos del personal que integran la DICT. El sistema puede ser consultado vía web y la información contenida se alimenta por ese mismo medio. Además brinda la facilidad al usuario de ir actualizando este directorio de acuerdo al área a la que pertenezcan, garantizando la seguridad de la misma, ya que no tienen acceso a la manipulación de otras áreas.

En el capítulo 1, se presentan los fundamentos teóricos en computación que dan pauta al desarrollo de esta tesis y que permiten ofrecer la mejor solución a las necesidades de la DICT.

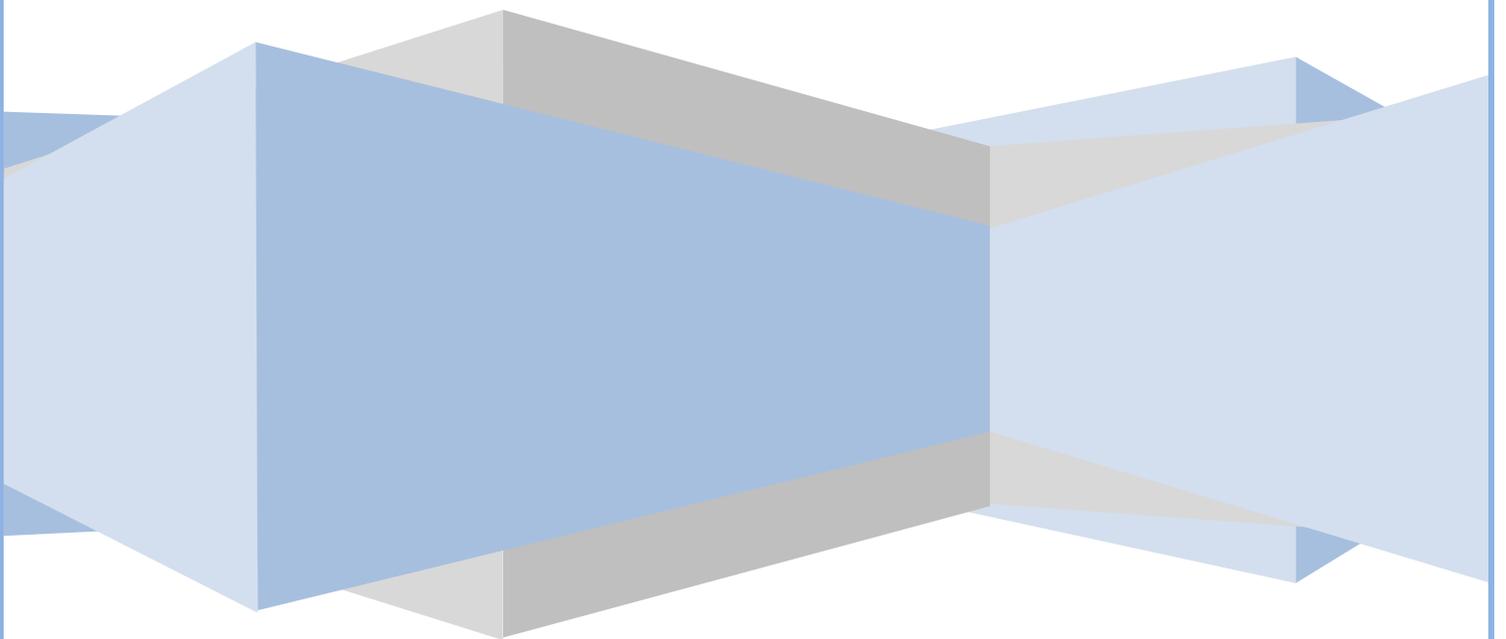
En el capítulo 2, se especifican todas aquellas características y requerimientos del sistema. Se explica también la etapa de diseño, donde se da a conocer la propuesta de software, un diagrama jerárquico funcional y la creación del modelo de la Base de Datos.

En el capítulo 3, se detalla la etapa de implementación, donde se lleva a cabo la elección de la plataforma operativa, la creación de la base de datos y la programación de la interfaz. También, se explica y ejemplifica el uso del sistema.

Finalmente en el capítulo 4, se exponen las conclusiones acerca del sistema y los logros obtenidos.

# Capítulo 1

## Fundamentos teóricos



# CAPÍTULO 1



## 1.1 BASES DE DATOS

### 1.1.1 DEFINICIONES Y CONCEPTOS DE LAS BASES DE DATOS

La historia de las bases de datos surge desde mediados de los años sesenta. En 1970 Codd<sup>1</sup> propuso el modelo relacional, este modelo es el que ha marcado la línea de investigación por muchos años, hoy en día se encuentran los modelos orientados a objetos.

Una Base de Datos es una colección de archivos interrelacionados creados con un Sistema Manejador de Bases de Datos (DBMS). El contenido de una base se obtiene combinando datos de todas las diferentes fuentes en una organización, de tal manera que los datos estén disponibles para todos los usuarios y los datos redundantes pueden eliminarse o por lo menos minimizarse

Las bases de datos son conjuntos de datos pertenecientes a un mismo contexto y almacenados de forma sistemática para ser usados después; este conjunto de datos tiene

---

<sup>1</sup> Edgar Frank Codd, Científico informático inglés que realizó teorías sobre modelado de datos y que definió las tres primeras Formas Normales que se utilizan en la normalización de sistemas de bases de datos.

que estar estructurado, no debe ser redundante, además su implementación debe ser accesible en tiempo real y la manipulación de los usuarios debe ser amigable.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que éstos explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas.

Una Base de Datos se conforma de cuatro componentes principales brevemente descritos a continuación:

- **Datos.**- Deben recoger toda la información necesaria, sin que exista repetición de la misma dentro de la base.
- **Hardware.**- Es el soporte físico que permite almacenar la información de la DB.
- **Software.**- Permite interactuar con la base de datos de manera eficiente.
- **Personal.**- Las bases de datos son manipuladas por personal con diferentes perfiles, entre ellos podemos mencionar:
  - *Administradores del DBMS (DBA por sus siglas en inglés).* Encargado del control y manejo de la Base.
  - *Administradores de la base (DBO por sus siglas en inglés).* Encargado de diseñar la base de datos, definir las estructuras de almacenamiento y métodos de acceso (tablas), modificar el diseño y organización física de la Base y una de las más importantes es el otorgar permisos de acceso a usuarios dentro de la base.
  - *Programadores de aplicaciones o Desarrolladores.* Diseñan y programan las aplicaciones necesarias para la utilización de las bases de datos.
  - *Usuarios.* Aquellos que utilizan el sistema de base de datos por medio de una interfaz gráfica e interactúan con los mismos dependiendo de los permisos que le hayan sido otorgados.

### 1.1.2 SISTEMA MANEJADOR DE BASES DE DATOS (DBMS)

En 1964, se conciben los primeros Gestores de Base de Datos (DBMS: Database Management System), por medio de los cuales se pretende dar un viraje a los Sistemas de Archivos, los cuales se limitan a la estructuración del almacenamiento físico de los datos. Con los DBMS se crea el concepto de Administración de los datos, por medio de actividades integradas que permiten verlos físicamente en un sólo almacenamiento pero lógicamente se manipulan a través de esquemas compuestos por estructuras donde se establecen vínculos de integridad, métodos de acceso y organización física sobre los datos, permitiendo así obtener valores agregados de utilización tales como: manejo de usuarios, seguridad, atomicidad e independencia física y lógica de los datos, entre otros.

Un sistema de Base de Datos es un sistema computarizado de información para el manejo de datos por medio de paquetes de software llamados sistemas de manejo de base de datos (DBMS). Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos por manejar. El DBMS interpreta y procesa las peticiones del usuario para recobrar información de la base.

El DBMS<sup>2</sup> interpreta y procesa las peticiones de los usuarios para extraer la información de la base de datos. Posteriormente responde a las peticiones llamando a los subprogramas apropiados para las tareas, los cuales realizarán su función especial para interpretar la petición o localizar los datos deseados y presentarlos, evitando con ello la realización de programas que organicen el almacenamiento y acceso de la información.

Una de las ventajas del DBMS es que puede ser invocado desde programas de aplicación que pertenecen a Sistemas Transaccionales escritos en algún lenguaje de alto nivel, para la creación o actualización de las bases de datos, o bien para efectos de consulta a través de lenguajes propios que tienen las bases de datos o lenguajes de cuarta generación.

---

<sup>2</sup> Conocido también como Gestor de Base de datos

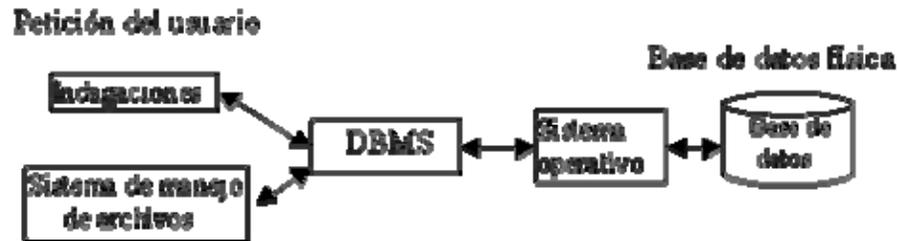


Fig 1.1 Interacción entre la BD, los DBMS y el usuario

La figura 1.1 muestra el DBMS como interfaz entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

Los sistemas de administración de bases de datos son usados para:

- Permitir a los usuarios acceder y manipular la base de datos proveyendo métodos para construir sistemas de procesamiento de datos para aplicaciones que requieran acceso a los datos.
- Proveer a los administradores las herramientas que les permitan ejecutar tareas de mantenimiento y administración de los datos.

### ***Funciones de un DBMS***

Las funciones de los DBMS se dividen en dos partes igual de importantes, las funciones principales y las funciones específicas.

Las funciones principales de un DBMS son:

- Definición de la Base de Datos.
- Creación de la Base de Datos.
- Establecer y mantener las trayectorias de acceso a la Base de Datos de tal forma que los datos puedan ser accedidos rápidamente.
- Administración de los datos almacenados.

- Seguridad.
- Control de Concurrencia<sup>3</sup>.
- Integridad de Datos.
- Lenguaje de acceso a la Base de Datos (Interfaz para la programación de aplicaciones).
- Interfaz de Comunicación de Datos<sup>4</sup>.
- Interacción con el manejador de archivos.
- Respaldo y Recuperación.

Las funciones *específicas* de los DBMS son:

- Compuesto de un Lenguaje de Definición de Datos (DDL), de un Lenguaje de Manipulación de Datos (DML) y de un Lenguaje de Consulta (SQL).
- Investigar quejas sobre el tiempo de respuesta del sistema, la precisión de los datos y la facilidad de uso. Si se requieren cambios serán planeados y puestos en práctica por el DBA.
- Vigilar periódica y continuamente las actividades de los usuarios en la base de datos por parte del DBA.
- Administrar y controlar la migración tanto de las estructuras, como de los datos y las aplicaciones.
- Realizar cambios en la configuración y/o en algunos parámetros de operación del DBMS.

---

<sup>3</sup> Enfocado a la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos

<sup>4</sup> Crea y organiza a la Base de datos de tal forma que el acceso a la información sea rápido y el control de la misma sea eficiente.

## **Arquitectura de un DBMS**

Las arquitecturas de bases de datos han evolucionado mucho desde sus comienzos, aunque la considerada estándar hoy en día es la descrita por el comité ANSI/X3/SPARC (*Standard Planning and Requirements Committee of the American National Standards Institute on Computers and Information Processing*), que data de finales de los años setenta. Este comité propuso una arquitectura general para DBMSs basada en tres niveles o esquemas: el nivel interno, el nivel externo y el nivel conceptual (fig 1.2). Así mismo describió las interacciones entre estos tres niveles y todos los elementos que conforman cada uno de ellos (fig 1.3).

La descripción de los diferentes niveles en un DBMS se detalla a continuación:

**Nivel interno, o de máquina** Describe en detalle la forma de cómo se almacenan los datos en los dispositivos de almacenamiento. Responde a las cuestiones de rendimiento planteadas al hacer el diseño físico de la BD y al ajustarlo posteriormente a nuevas necesidades.

**Nivel externo, o de usuario** Es lo que el usuario final puede visualizar del sistema terminado, muestra sólo una parte de la base de datos al usuario acreditado para verla.

**Nivel conceptual** Describe qué datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos, describe la base de datos completa en términos de su estructura de diseño.

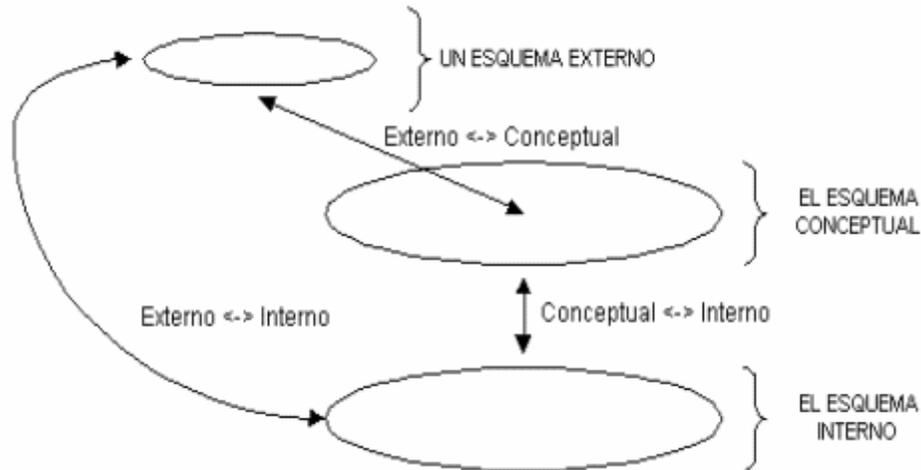


Fig. 1.2 Arquitectura descrita por el comité ANSI/X3/SPARC

## Arquitectura de un DBMS

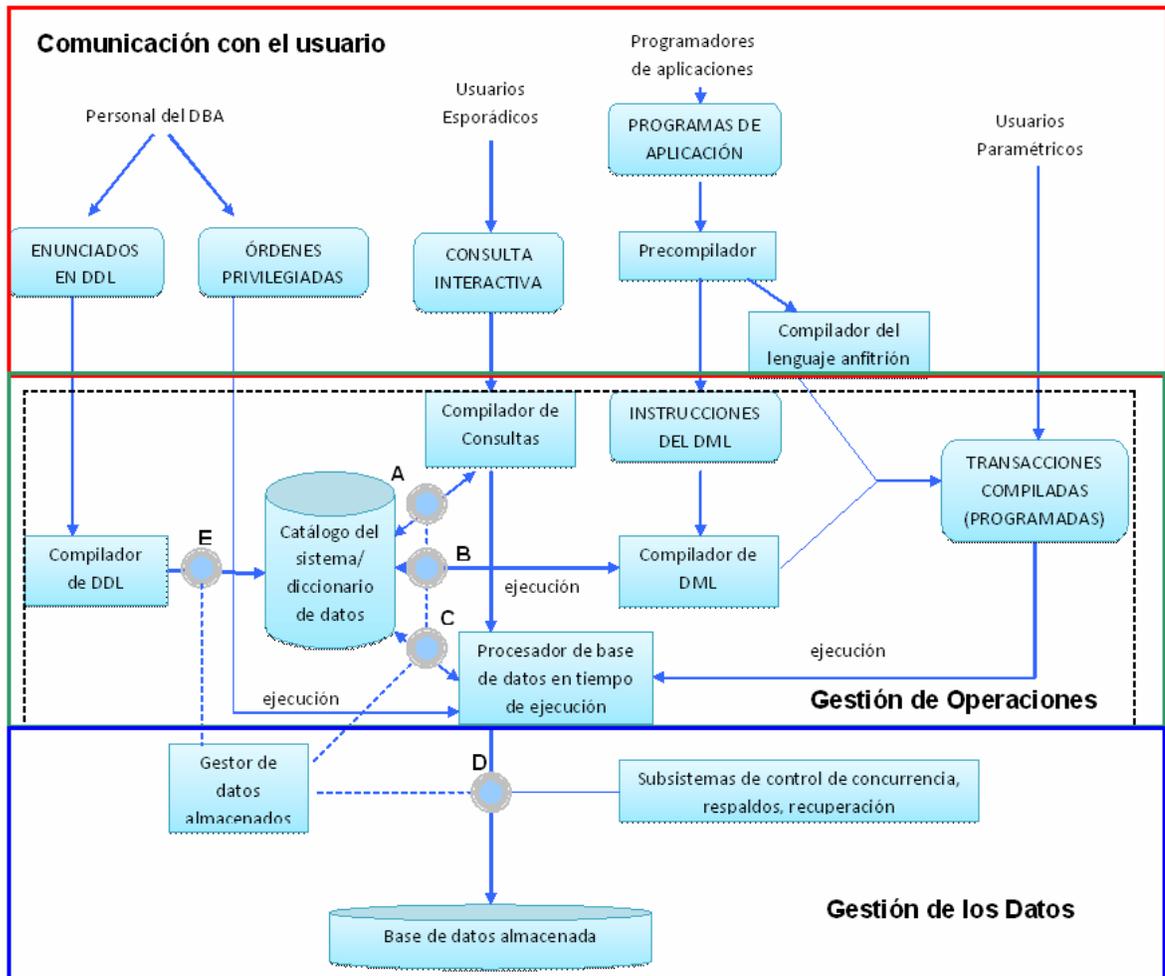


Fig. 1.3 Arquitectura Integrada

El tipo de arquitectura integrada es en general preferible a la arquitectura separada y el más común entre los DBMSs comerciales. La consecuencia de una integración de los lenguajes de definición de datos (DDL) y los de manipulación de datos (DML) en un sólo lenguaje (DMDL: *Data Manipulation and Description Language*) es que, por un lado, resulta muy cómoda para el DBA, puesto que le basta con aprender un solo lenguaje formal para realizar todas las tareas de creación y mantenimiento de la base de datos. Por otro lado, estos sistemas (tanto los separados como los uniformes) fuerzan una proyección directa desde el nivel externo al interno, haciendo que el nivel conceptual, el fundamental según la arquitectura ANSI/X3/SPARC, desaparezca o se implemente en el nivel externo como una vista global externa. Por esta razón algunos DBAs inexpertos tienden a obviar la fase de análisis, cuando de hecho es la vital para la correcta implementación de la base de datos.

### ***Manejador de bases de datos MySQL***

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar una licencia que les permita ese uso.

Algunas de las características de MySQL más importantes y utilizadas en las últimas versiones son:

- **Velocidad:** Mayor rapidez en comparación con la mayoría de sus rivales.
- **Funcionalidad:** Disponibilidad de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- **Portabilidad:** Ejecución en la mayoría de los sistemas operativos y, en la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.

- **Facilidad de uso:** Fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.
- **Múltiples motores de almacenamiento:** Permite al usuario escoger la que sea más adecuada para cada tabla de la base de datos<sup>5</sup>.

### 1.1.3. MODELO DE DATOS

Una de las características fundamentales de los sistemas de bases de datos es que proporcionan cierto nivel de abstracción de datos, al ocultar las características sobre el almacenamiento físico que la mayoría de usuarios no necesita conocer. Los modelos de datos son el instrumento principal para ofrecer dicha abstracción.

Para entender el tema, hay que definir los siguientes conceptos.

**Modelo:** Es una representación de la realidad que contiene las características generales de algo que se va a realizar. En base de datos, esta representación la elaboramos de forma gráfica.

**Modelo de datos:** Es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas y actualizaciones de datos. Además, los modelos de datos más modernos incluyen conceptos para especificar comportamiento, permitiendo especificar un conjunto de operaciones definidas por el usuario.

Los modelos de datos se pueden clasificar dependiendo de los tipos de conceptos que ofrecen para describir la estructura de la base de datos:

---

<sup>5</sup> Algunos de estos motores de almacenamiento son: MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x.

- **Modelos de datos de alto nivel**, o *modelos conceptuales* disponen de conceptos muy cercanos al modo en que la mayoría de los usuarios percibe los datos. Los modelos conceptuales utilizan conceptos como entidades, atributos y relaciones. Una *entidad* representa un objeto o concepto del mundo real como, por ejemplo, un empleado de la empresa inmobiliaria o una oficina. Un *atributo* representa alguna propiedad de interés de una entidad como, por ejemplo, el nombre o el salario del empleado. Una *relación* describe una interacción entre dos o más entidades, por ejemplo, la relación de trabajo entre un empleado y su oficina.
- **Modelos de datos de bajo nivel**, proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el ordenador.
- **Modelos físicos** están dirigidos al personal informático, no a los usuarios finales. Los modelos físicos describen cómo se almacenan los datos en el ordenador: el formato de los registros, la estructura de los ficheros (desordenados, ordenados, etc.) y los métodos de acceso utilizados (índices, etc.).
- **Modelos lógicos**, se encuentran entre estos dos extremos cuyos conceptos pueden ser entendidos por los usuarios finales, aunque no están demasiado alejados de la forma en que los datos se organizan físicamente. Los modelos lógicos ocultan algunos detalles de cómo se almacenan los datos, pero pueden implementarse de manera directa en un ordenador.

Los objetivos del modelo de datos son dos:

1. **Formalización:** definir formalmente las estructuras permitidas y las restricciones a fin de representar los datos de un SI.
2. **Diseño:** el modelo resultante es un elemento básico para el desarrollo de la metodología de diseño de la base de datos.

Los diferentes modelos de datos comparten, aunque con diferentes nombres y notaciones, unos elementos comunes, componentes básicos de la representación de la realidad que realizan. Estos componentes se identifican gracias a la clasificación, y pueden identificarse conceptos estáticos y conceptos dinámicos.

## **Modelo relacional**

En 1970, el modo en que se veían las bases de datos cambió por completo cuando E. F. Codd introdujo el modelo relacional. En aquellos momentos, el enfoque existente para la estructura de las bases de datos utilizaba punteros físicos (direcciones de disco) para relacionar registros de distintos ficheros. Si, por ejemplo, se quería relacionar un registro *A* con un registro *B*, se debía añadir al registro *A* un campo conteniendo la dirección en disco del registro *B*. Este campo añadido, un puntero físico, siempre señalaría desde el registro *A* al registro *B*. Codd demostró que estas bases de datos limitaban en gran medida los tipos de operaciones que los usuarios podían realizar sobre los datos. Además, estas bases de datos eran muy vulnerables a cambios en el entorno físico. Si se añadían los controladores de un nuevo disco al sistema y los datos se movían de una localización física a otra, se requería una conversión de los ficheros de datos. Estos sistemas se basaban en el modelo de red y el modelo jerárquico, los dos modelos lógicos que constituyeron la primera generación de los SGBD.

El modelo relacional representa la segunda generación de los SGBD. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Pero detrás de esa simple estructura hay un fundamento teórico importante del que carecen los SGBD de la primera generación, lo que constituye otro punto a su favor.

Dada la popularidad del modelo relacional, muchos sistemas de la primera generación se han modificado para proporcionar una interfaz de usuario relacional, con independencia del modelo lógico que soportan (de red o jerárquico). Por ejemplo, el sistema de red IDMS ha evolucionado a IDMS/R e IDMS/SQL, ofreciendo una visión relacional de los datos.

En los últimos años, se han propuesto algunas extensiones al modelo relacional para capturar mejor el significado de los datos, para disponer de los conceptos de la orientación a objetos y para disponer de capacidad deductiva.

El modelo relacional, como todo modelo de datos, tiene que ver con tres aspectos de los datos:

- Estructura de datos.
- Integridad de datos.
- Manejo de datos.

### Definiciones y conceptos

*Una relación es una tabla con columnas y filas.* Un SGBD sólo necesita que el usuario pueda percibir la base de datos como un conjunto de tablas. Esta percepción sólo se aplica a la estructura lógica de la base de datos (en el nivel externo y conceptual de la arquitectura de tres niveles ANSI-SPARC). No se aplica a la estructura física de la base de datos, que se puede implementar con distintas estructuras de almacenamiento.

*Un atributo es el nombre de una columna de una relación.* Representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Los atributos pueden ser simples o compuestos. Un *atributo simple* es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un *atributo compuesto* es un atributo con varios componentes, cada uno con un significado por sí mismo. Un grupo de atributos se representa mediante un atributo compuesto cuando tienen afinidad en cuanto a su significado, o en cuanto a su uso.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un *atributo monovalente* es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un *atributo polivalente* es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece.

Por último, los atributos pueden ser derivados. Un *atributo derivado* es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

El concepto de *dominio* es importante porque define todos los valores posibles que puede tomar un atributo. Esto hace que haya más información disponible para el sistema cuando éste va a ejecutar una operación relacional, de modo que las operaciones que son semánticamente incorrectas, se pueden evitar. Los SGBD relacionales no ofrecen un soporte completo de los dominios ya que su implementación es extremadamente compleja.

*Una tupla es una fila de una relación.* Los elementos de una relación son las tuplas o filas de la tabla. En la relación OFICINA, cada tupla tiene seis valores, uno para cada atributo. Las tuplas de una relación no siguen ningún orden.

*El grado de una relación es el número de atributos que contiene.* La relación OFICINA es de grado seis porque tiene seis atributos. Esto quiere decir que cada fila de la tabla es una tupla con seis valores. El grado de una relación no cambia con frecuencia.

*La cardinalidad de una relación es el número de tuplas que contiene.* Indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. Ya que en las relaciones se van insertando y borrando tuplas a menudo, la cardinalidad de las mismas varía constantemente.

*Una base de datos relacional es un conjunto de relaciones normalizadas.*

### **Propiedades de las relaciones**

Las relaciones tienen las siguientes características:

- Cada relación tiene un nombre y éste es distinto del nombre de todas las demás.
- Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están *normalizadas*.
- No hay dos atributos que se llamen igual.
- El orden de los atributos no importa: los atributos no están ordenados.
- Cada tupla es distinta de las demás: no hay tuplas duplicadas.
- El orden de las tuplas no importa: las tuplas no están ordenadas.

### **Tipos de relaciones**

En un SGBD relacional pueden existir varios tipos de relaciones, aunque no todos manejan todos los tipos.

- *Relaciones base.* Son relaciones reales que tienen nombre y forman parte directa de la base de datos almacenada (son autónomas).
- *Vistas.* También denominadas relaciones virtuales, son relaciones con nombre y derivadas: se representan mediante su definición en términos de otras relaciones con nombre, no poseen datos almacenados propios.

- *Instantáneas*. Son relaciones con nombre y derivadas. Pero a diferencia de las vistas, son reales, no virtuales: están representadas no sólo por su definición en términos de otras relaciones con nombre, sino también por sus propios datos almacenados. Son relaciones de sólo de lectura y se refrescan periódicamente.
- *Resultados de consultas*. Son las relaciones resultantes de alguna consulta especificada. Pueden o no tener nombre y no persisten en la base de datos.
- *Resultados intermedios*. Son las relaciones que contienen los resultados de las subconsultas. Normalmente no tienen nombre y tampoco persisten en la base de datos.
- *Resultados temporales*. Son relaciones con nombre, similares a las relaciones base o a las instantáneas, pero la diferencia es que se destruyen automáticamente en algún momento apropiado.

## Claves

Ya que en una relación no hay tuplas repetidas, éstas se pueden distinguir unas de otras, es decir, se pueden identificar de modo único. La forma de identificarlas es mediante los valores de sus atributos.

Una *superclave* es un atributo o un conjunto de atributos que identifican de modo único las tuplas de una relación.

Una *clave candidata* es una superclave en la que ninguno de sus subconjuntos es una superclave de la relación. Se consideran claves candidatas si y sólo si satisfacen las siguientes propiedades:

- *Unicidad*: nunca hay dos tuplas en la relación con el mismo valor en sus atributos.
- *Irreducibilidad (minimalidad)*: ningún subconjunto tiene la propiedad de unicidad, es decir, no se pueden eliminar componentes sin destruir la unicidad.

Cuando una clave candidata está formada por más de un atributo, se dice que es una *clave compuesta*. Una relación puede tener varias claves candidatas

La *clave primaria* de una relación es aquella clave candidata que se escoge para identificar sus tuplas de modo único.

Una *clave ajena* es un atributo o un conjunto de atributos de una relación cuyos valores coinciden con los valores de la clave primaria de alguna otra relación (puede ser la misma). Las claves ajenas representan *relaciones entre datos*.

### Modelo Entidad-Relación

El modelo Entidad-Relación<sup>6</sup> (Modelo ER) es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado *modelo entidad-relación extendido*.

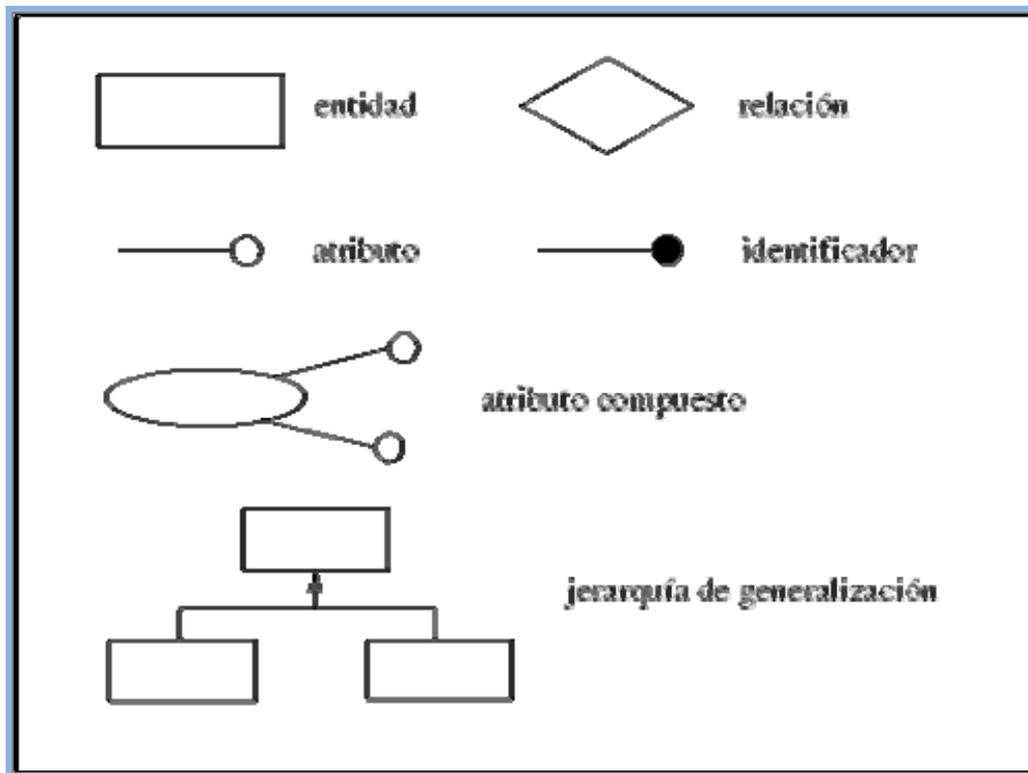


Fig. 1.4 Conceptos del modelo entidad-relación extendido.

<sup>6</sup> Este modelo fue introducido por Peter Chen en 1976.

## Entidad

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles. Una *entidad débil* es una entidad cuya existencia depende de la existencia de otra entidad. Una *entidad fuerte* es una entidad que no es débil.

## Relación (interrelación)

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan *entidades participantes*. El número de participantes en una relación es lo que se denomina *grado* de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación *binaria*; si son tres las entidades participantes, la relación es *ternaria*; etc.

Una *relación recursiva* es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

La *cardinalidad* con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. Las reglas que definen la cardinalidad de las relaciones son las *reglas de negocio*.

## Identificador

Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos. Las relaciones no tienen identificadores.

## Jerarquía de generalización

Cada jerarquía es total o parcial, y exclusiva o superpuesta. Una jerarquía es *total* si cada ocurrencia de la entidad genérica corresponde al menos con una ocurrencia de alguna subentidad. Es *parcial* si existe alguna ocurrencia de la entidad genérica que no corresponde con ninguna ocurrencia de ninguna subentidad. Una jerarquía es *exclusiva* si cada ocurrencia de la entidad genérica corresponde, como mucho, con una ocurrencia de una sola de las subentidades. Es *superpuesta* si existe alguna ocurrencia de la entidad genérica que corresponde a ocurrencias de dos o más subentidades diferentes.

## Reglas de integridad

Proporcionan la seguridad de que una información perteneciente a la base de datos no ha sido alterada, borrada, reordenada, copiada, etc. y son:

- *Integridad de Relaciones*.- Donde ningún atributo que forme parte de una *llave primaria* puede aceptar *valores nulos*.
- *Integridad Referencial*- La integridad referencial es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad.

Existen además dos formalismos importantes para el manejo de datos en el ámbito del modelo relacional, estos son el Álgebra relacional, que proporciona una serie de operaciones que pueden usarse para decir al sistema cómo *construir* la relación deseada a partir de las relaciones de la base de datos, El otro formalismo es el Cálculo relacional, el cual proporciona una notación para formular la *definición* de la relación deseada en términos de las relaciones de la base de datos.

### **Normalización**

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataban de manipular los datos. Las guías que la normalización provee crean el marco de referencia para simplificar una estructura de datos compleja.

Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco.

El proceso de normalización tiene un nombre y una serie de reglas para cada fase. Esto puede parecer un poco confuso al principio, pero poco a poco se va entendiendo el proceso, así como las razones para hacerlo de esta manera.

Existen básicamente tres niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización. No siempre es una buena idea tener una base de datos conformada en el nivel más alto de normalización, puede llevar a un nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

En la tabla 1.1 se describe brevemente en que consiste cada una de las reglas, y posteriormente se explican con más detalle.

<b>Regla</b>	<b>Descripción</b>
Primera Forma Normal (1FN)	Incluye la eliminación de todos los grupos repetidos.
Segunda Forma Normal (2FN)	Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (PK).
Tercera Forma Normal (3FN)	Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

### **Primera Forma Normal**

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Poner la base de datos en la Primera Forma Normal resuelve el problema de los encabezados de columna múltiples, ayudando a clarificar la base de datos y a organizarla en partes más pequeñas y más fáciles de entender. En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos y diferentes aspectos, sólo tenemos que entender los objetos pequeños y más tangibles, así como las relaciones que guardan con otros objetos también pequeños.

## **Segunda Forma Normal**

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos.

Una vez alcanzado el nivel de la Segunda Forma Normal, se controlan la mayoría de los problemas de lógica. Podemos insertar un registro sin un exceso de datos en la mayoría de las tablas.

## **Tercera Forma Normal**

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas<sup>7</sup>

Cuando las tablas están en la Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria, y no debe haber datos repetidos. Esto provee un esquema limpio y elegante, que es fácil de trabajar y expandir.

---

<sup>7</sup> una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen de otras columnas que tampoco son llave.

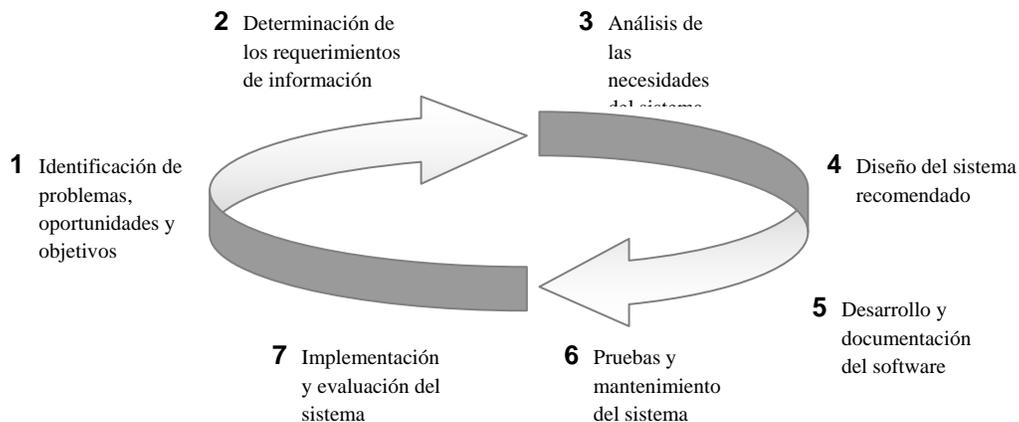
## 1.2. METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS

Una metodología completa es algo más que una notación, un proceso, y herramientas. Además de una "notación, de un proceso, y de herramientas," estas "metodologías completas" proporcionan:

- Guías para estimar costos,
- Manejo del proyecto en las tareas y entregas,
- Medidas y métricas,
- Formas definidas y dirección en las entregas de la construcción,
- Políticas y procedimientos para garantizar la calidad del software,
- Descripciones de los roles y programas de entrenamiento detallados,
- Ejemplos totalmente trabajados,
- Ejercicios de entrenamiento,
- Técnicas para adaptar el método, y
- Técnicas definidas

### 1.2.1. Definición de Metodología

La Metodología es el conjunto de métodos empleados para el desarrollo de sistemas automatizados mediante procedimientos, técnicas y herramientas que los hacen estándares y sistematizados. Dichas metodologías están compuestas por cuatro etapas que en conjunto se denominan "El Ciclo de vida de un sistema" (fig. 1.5). Estas etapas son la planeación, el análisis, el diseño y el desarrollo



**Fig. 1.5** Ciclo de vida del desarrollo de sistemas

En la figura anterior podemos observar el ciclo de vida de un sistema para lo cual se necesita identificar el problema para poder delimitar los requerimientos que se tengan y con esto poder iniciar con el análisis del mismo partiendo de un modelo sencillo hacia el diseño, y comenzar así con la programación. Cuando ya se cuenta con un prototipo del sistema podemos realizar las pruebas pertinentes y afinar los detalles que requieran atención para entregar el producto terminado. Cabe destacar que, en caso de tener nuevos requerimientos o no llegar a los resultados deseados, es posible regresar a la etapa de análisis y así incrementar la información creando nuevos modelos que incluyan las posibles soluciones de los requerimientos o problema.

A continuación se describen cada una de las etapas que integran este ciclo de vida.

### **1.2.2. PLANEACIÓN**

El objetivo de esta fase consiste en la elaborar junto con el equipo humano de la organización donde se va a implementar el sistema, los objetivos generales, específicos y los esquemas generales de la manera más clara y precisa. Para ello se realizan juntas con el cliente del sistema para identificar así las necesidades de éste.

En esta fase se debe responder a preguntas como:

- ¿Cuáles son los objetivos que deberán cumplir?
- ¿Cuáles son las necesidades de la organización que deben resolverse?
- ¿Quiénes son los usuarios del sistema y sus necesidades?
- ¿Cuál es la información y los datos que usan y generan en la organización para desarrollar sus funciones?
- ¿Qué sistemas se encuentran en funcionamiento en la organización?
- ¿Cuáles son los productos esperados del sistema?
- ¿Cuáles es el alcance del sistema?

### 1.2.3. ANÁLISIS

Este es el primer paso en el sistema clásico. Su propósito es “el estudio del sistema actual de la empresa y de información y la definición de las necesidades y prioridades de los usuarios para la construcción de un nuevo sistema de información”.

Incluye de forma iterativa las siguientes tareas o fases <sup>8</sup>:

- **Conceptualización** Consiste en obtener una visión de muy alto nivel del sistema, identificando sus elementos básicos y las relaciones de éstos entre sí y con el entorno.
- **Análisis funcional** Describe las acciones o transformaciones que tienen lugar en el sistema especificadas en forma de procesos que reciben unas entradas y producen unas salidas.
- **Análisis de condiciones** Debe reflejar todas aquellas limitaciones impuestas al sistema que restringen el margen de las soluciones posibles. Estas se derivan a veces de los propios objetivos del sistema:
  - Operativas, como son las restricciones físicas, ambientales, de mantenimiento, de personal, de seguridad, etc.
  - De calidad, como fiabilidad, mantenibilidad, seguridad, convivencia, generalidad, etc.

Sin embargo, en otras ocasiones las constricciones vienen impuestas por limitaciones en los diferentes recursos utilizables:

- Económicos, reflejados en un presupuesto
- Temporales, que suponen unos plazos a cumplir
- Humanos

---

<sup>8</sup> <http://www.daedalus.es/inteligencia-de-negocio/sistemas-complejos/ingenieria-de-sistemas/analisis-de-sistemas/>

- Metodológicos, que conllevan la utilización de técnicas determinadas
- Materiales, como espacio, herramientas disponibles, etc.
- **Construcción de modelos** Una de las formas más habituales y convenientes de analizar un sistema consiste en construir un prototipo del mismo.
- **Validación del análisis** A fin de comprobar que el análisis efectuado es correcto y evitar, en su caso, la posible propagación de errores a la fase de diseño, es imprescindible proceder a la validación del mismo. Para ello hay que comprobar que
  - El análisis debe ser consistente y completo
  - Si el análisis se plantea como un paso previo para realizar un diseño, habrá que comprobar además que los objetivos propuestos son correctos y realizables

#### **1.2.4. DISEÑO**

Evalúa las soluciones alternativas y especifica una solución detallada de tipo informático. En esta etapa se decide lo que se tiene que hacer y la manera en que se hará, es decir, se determinan las condiciones y restricciones técnicas necesarias para satisfacer la conducta esperada del sistema. Se realiza una vez planteado el problema y construidos los modelos que representan el proceso. Además se incorporan los componentes en equipo y programas que forman parte del sistema, proporcionando así un anteproyecto para la implementación.

##### **Fases del Diseño:**

- Elección de una solución de diseño entre las soluciones candidatas. Estas soluciones se evalúan con los siguientes criterios: Viabilidad técnica, operativa, económica, en tiempo.
- Evaluación del hardware y software requeridos

- Diseño e Integración del nuevo sistema.
  - *Diseño General.* El método comúnmente utilizado es la modelización <sup>9</sup>. Los modelos de diseño general describen:
    - La estructura de los archivos y de las bases de datos.
    - Los métodos y procedimientos de proceso mediante un diagrama de flujo.
    - La estructura de la red informática.
  - *Diseño Detallado.* Se divide en:
    - Diseño Externo. Conjunto de especificaciones de la interfaz del sistema con sus usuarios. Incluyen entradas, consultas, salidas, diseño de ventanas y transición entre ventanas.
    - Diseño Interno. Especificaciones de aplicación del sistema, los archivos, diseño de la base de datos.

### 1.2.5. Desarrollo

Es la construcción del nuevo sistema y el paso de dicho sistema a “producción”. Se debe tener perfectamente determinado que es lo que se va a hacer y cómo, para después realizar la programación y las pruebas de verificación del funcionamiento del mismo.

---

<sup>9</sup> Acto de elaborar una o más representaciones gráficas del sistema.

### 1.3. SISTEMAS DE INFORMACIÓN VÍA INTERNET

Cuando se piensa sobre el porqué del uso de computadoras en las empresas, la respuesta parece bastante obvia, para registrar los eventos que van sucediendo. Sin embargo existe un motivo aún más importante y no tan obvio, y es que la información puede usarse para respaldar la toma de decisiones. Sin importar el tamaño de una empresa u organización, la conducción exitosa de la misma estará dramáticamente influenciada por la precisión de sus registros y la adopción de decisiones acertadas.

Los enlaces de comunicación y las bases de datos permiten el acceso a recursos de información que están más allá de la inmediatez física, sobrepasando las limitaciones geográficas. Las computadoras permiten la utilización de masas de información; pero no sólo basta la disponibilidad de la cantidad de información, se trata de contar con información de calidad. Los sistemas de información basados en computadoras no sólo son capaces de suministrarnos información de calidad y oportuna, sino que también pueden respaldar la toma de decisiones.

#### **Elementos de un sistema de información**

Un sistema de información es un conjunto organizado de elementos, estos elementos son de 4 tipos:

- Personas.
- Datos.
- Actividades o técnicas de trabajo.
- Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todo ese conjunto de elementos interactúan entre sí para procesar los datos y la información (incluyendo procesos manuales y automáticos) y así distribuirla de la manera más adecuada posible en una determinada organización en función de sus objetivos. Dichos elementos y la forma en que se interrelacionan se describen a continuación.

### 1.3.1. INTERNET

Es una colección de redes de comunicación interconectadas por puentes o dispositivos de encaminamiento. Término que hace referencia la red global mas grande del mundo, la cual conecta a decenas de miles de redes en todo el mundo y tiene una “cultura” centrada en la investigación y una estandarización basada en el uso en la vida real (fig 1.6). Muchas tecnologías de punta en redes provienen de la comunidad de Internet. La red Internet se origino parcialmente de ARPANET. En un tiempo se le llamo la Internet de DARPA.

Internet fue desarrollado originariamente para los militares de Estados Unidos, y después se utilizó para el gobierno, investigación académica y comercial y para comunicaciones.



Fig 1.6 Red global o Internet

Es una combinación de hardware (ordenadores interconectados por vía telefónica o digital) y software (protocolos y lenguajes que hacen que todo funcione). Es una infraestructura de redes a escala mundial (grandes redes principales (tales como **MILNET**, **NSFNET**, y **CREN**), y redes más pequeñas que conectan con ellas) que conecta a la vez a todos los tipos de ordenadores.

Hay unos seis millones de ordenadores que utilizan Internet en todo el mundo y que utilizan varios formatos y protocolos internet:

- **Internet Protocol (IP):** protocolo que se utiliza para dirigir un paquete de datos desde su fuente a su destino a través de Internet.
- **Transport Control Protocol (TCP):** protocolo de control de transmisión, que se utiliza para administrar accesos.
- **User Datagram Protocol (UDP):** protocolo del datagrama del usuario, que permite enviar un mensaje desde un ordenador a una aplicación que se ejecuta en otro ordenador.

### 1.3.2. ARQUITECTURA CLIENTE SERVIDOR

Esta arquitectura consiste básicamente en que un programa -el cliente- que realiza peticiones a otro programa -el servidor- que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los *sistemas multicapa* en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes **computadoras** aumentando así el grado de distribución del sistema.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.

### 1.3.3. SERVIDORES DE PÁGINAS WEB

Un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

En el caso de un servidor web, el término anterior podría referirse a la máquina que almacena y maneja los sitios web, y en ese sentido es utilizado por las compañías que ofrecen hosting u hospedaje. Alternativamente el servidor web podría referirse al software como el servidor http de Apache, que funciona en la máquina y maneja una entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor web es un programa que implementa el *protocolo HTTP (hypertext transfer protocol)*. Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Es un programa que se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una *página web* que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

#### ***Servidor Web Apache***

El **servidor HTTP Apache** es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero

más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf eligió ese nombre porque quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft).

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

#### **1.3.4. Protocolos de comunicación**

Un protocolo es el conjunto de reglas que gobiernan la operación de unidades funcionales para llevar a cabo la comunicación. El Protocolo Internet es aquel que establece la interconexión de redes que proporciona servicios sin conexión a través de múltiples redes de comunicación de paquetes.

En el intercambio de datos entre computadoras, terminales y otros dispositivos de comunicación, los procesos involucrados pueden llegar a ser bastante complejos. Por ejemplo, la transferencia de un archivo entre dos computadoras; debe haber un camino entre las dos computadoras, directo a través de una red de comunicación, pero además normalmente se requiere la realización de tareas adicionales.

Los **Protocolos de Comunicación** más importantes, que determinan el contexto de intercambio de mensajes (correo electrónico), de las conexiones remotas<sup>10</sup> (telnet), o de la transferencia de archivos FTP, entre otras actividades de la red, se tienen los siguientes:

### **TELNET (*Telecommunications Network*)**

Es un protocolo estándar de sesión de trabajo remoto en internet, que permite la conexión hacia una computadora remota utilizando el principio de cliente-servidor, donde la computadora-cliente solicita abrir una sesión en la computadora remota (servicio o host) y ambas establecen los parámetros por los que se registrará la sesión.

### **FTP (*File Transfer Protocol*)**

Es un protocolo que permite la transferencia de archivos, entre computadoras de distintos tipos y distintos sistemas operativos en el universo de Internet. Se recomienda utilizarlo cuando se desea transferir un archivo extenso.

### **SMTP (*Simple Mail Transfer Protocol*)**

Permite realizar direccionamiento de los correos, es decir si una persona trabaja en distintas empresas y tiene asignada distintas direcciones de correo puede hacer que todos sus mensajes sean enviados a una misma casilla de correo en forma automática.

### **HTTP (*Hypertext Transfer Protocol*)**

---

<sup>10</sup> Conexiones remotas.- Hace referencia a computadoras o redes no importando donde estén localizadas, tan solo que sean accesibles a través de Internet.

Es un protocolo que controla las peticiones y servicios de documentos HTML (Hypertext Markup Language) y que se utiliza para manejar la transferencia de este tipo de páginas en la World Wide Web (expuesto en el siguiente subtema).

### **PPP (*Point To Point Protocol*)**

Es un protocolo que permite a una computadora establecer comunicación con una red de datos remota, convirtiéndose en un nodo de dicha red y pudiendo usar todos sus servicios tal cual lo haría si esa computadora estuviese conectada a la red directamente. Para lograr lo anterior la computadora utiliza un puerto serial, un módem y, como medio físico de enlace, una línea telefónica convencional.

### **DNS (*Domain Name System*)**

El Sistema de Nombres de Dominios es un conjunto de protocolos y servicios sobre una red TCP/IP. Se utilizan principalmente para buscar direcciones IP en hosts tomando como base los nombres de éstos, permitiendo a los usuarios comunicarse con otros equipos a través de nombres sencillos.

### **CGI (*Common Gateway Interface*)**

La creación de programas del tipo CGI es realizada bajo el lenguaje de programación PERL o PHP, permite la generación de páginas dinámicas, generadas a través de la ejecución de un programa en el servidor de Internet.

El protocolo perteneciente a la capa física más importante es:

### **TCP/IP (*Transport Control Protocol/Internet Protocol*)**

El protocolo TCP/IP (Transmission Control Protocol / Internet Protocol) hace posible enlazar cualquier tipo de computadoras, sin importar el sistema operativo que usen o el fabricante. Este protocolo fue desarrollado originalmente por el ARPA (Advanced Research Projects Agency) del Departamento de Defensa de los Estados Unidos. Actualmente, es posible tener una red mundial llamada Internet utilizando este protocolo. Este sistema de IP permite a las redes mandar correo electrónico (e-mail), transferencia

de archivos (FTP) y tener una interacción con otras computadoras (TELNET) no importando donde estén localizadas, tan solo que sean accesibles a través de Internet.

Para entender el funcionamiento de los protocolos TCP/IP debe tenerse en cuenta la arquitectura que ellos proponen para comunicar redes. Tal arquitectura ve como iguales a todas las redes a conectarse, sin tomar en cuenta el tamaño de ellas, ya sean locales o de cobertura amplia. Define que todas las redes que intercambiarán información deben estar conectadas a una misma computadora o equipo de procesamiento (dotados con dispositivos de comunicación); a tales computadoras se les denominan compuertas, también recibe otros nombres como enrutadores o puentes.

Para que en una red dos computadoras puedan comunicarse entre sí ellas deben estar identificadas con precisión. Este identificador puede estar definido en niveles bajos (identificador físico) o en niveles altos (identificador lógico) dependiendo del protocolo utilizado. TCP/IP utiliza un identificador denominado dirección Internet o dirección IP, cuya longitud es de 32 bytes. La dirección IP identifica tanto a la red a la que pertenece una computadora como a ella misma dentro de dicha red.

Dicha identificación se hace por medio de las “Clases de Direcciones IP”, para clarificar una red con dirección Clase A queda definida con el primer octeto de la dirección, la Clase B con los dos primeros y la Clase C con los tres primeros octetos. Los octetos restantes definen los nodos en la red específica.

Clases	Número de Redes	Número de Nodos	Rango de Direcciones IP
A	127	16,777,215	1.0.0.0 a la 127.0.0.0
B	4095	65,535	128.0.0.0 a la 191.255.0.0
C	2,097,151	255	192.0.0.0 a la 223.255.255.0

### 1.3.5. WORLD WIDE WEB

**Web o la web, la red o www de World Wide Web**, es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir,

la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

Fue creada en 1989 en un instituto de investigación de Suiza, la web se basa en buscadores y el protocolo de transporte de hipertexto (http por sus siglas en inglés, hypertext transport protocol). La mayoría de los documentos de la web se crean utilizando lenguaje HTML.

Es importante saber que web o www no son sinónimo de Internet, la web es un subconjunto de Internet que consiste en páginas a las que se puede acceder usando un navegador. Internet es la red de redes donde reside toda la información. Tanto el correo electrónico, como FTPs, juegos, etc. son parte de Internet, pero no de la Web.

Para buscar hipertexto se utilizan programas llamados buscadores web que recuperan trozos de información (llamados *documentos* o *páginas web*) de los servidores web y muestran en la pantalla del ordenador de la persona que está buscando la información gráfica, textual o video e incluso audio.

Después se pueden seguir enlaces o hyperlinks en cada página a otros documentos o incluso devolver información al servidor para interactuar con él. Al acto de seguir un enlace tras otro a veces se le llama navegar en Internet.

La web se ha convertido en un medio muy popular de publicar información en Internet, y con el desarrollo del protocolo de transferencia segura (https), la web es ahora un medio de comercio electrónico donde los consumidores pueden escoger sus productos on-line y realizar sus compras utilizando la información de sus tarjetas bancarias de forma segura

### ***Concepto de Página Web***

En inglés **website** o **web site**, un **sitio web** es un sitio (localización) en la World Wide Web que contiene documentos (**páginas web**) organizados jerárquicamente. Cada documento contiene texto y o gráficos que aparecen como información digital en la pantalla de un ordenador. Un sitio puede contener una combinación de gráficos, texto, audio, vídeo, y otros materiales dinámicos o estáticos.

Cada sitio web tiene una **página de inicio**, que es el primer documento que ve el usuario cuando entra en el sitio web poniendo el nombre del dominio de ese sitio web en un navegador. El sitio normalmente tiene otros documentos (páginas web) adicionales. Cada sitio pertenece y es gestionado y por un individuo, una compañía o una organización.

### ***Páginas Estáticas***

Una página Web estática es aquella que es básicamente informativa, el visitante y administrador Web no pueden interactuar con la página para modificar su contenido. No se utilizan bases de datos ni se requiere programación. Para desarrollar una página web estática es suficiente utilizar código HTML. Cualquier webmaster, debe saber los buscadores de Internet se han hecho muy buenos reconociendo Webs dinámicas, navegando a través de ella de una manera muy fácil y por tanto, localizando rápidamente la información. Sin embargo, las Web estáticas tienen una desventaja; necesitarás una página separada (fichero) para cada una de las páginas Web.

### ***Páginas Dinámicas***

En una página Web dinámica se pueden dar de alta, baja o modificar de forma instantánea productos, servicios etc. Puede almacenar información de pedidos, clientes, usuarios, etcétera y enviarles emails corporativos con ofertas o novedades automáticamente.

Un área de administración a medida y protegida por contraseña, permite realizar estas actualizaciones de forma automática y ver el resultado en tiempo real. En un sitio dinámico las personas o usuarios pueden administrar los contenidos

## **1.3.6. Lenguajes de Programación para web**

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde

los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

Entre los lenguajes más usados para el desarrollo de páginas web:

### **Lenguaje HTML**

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).

### **Lenguaje PHP**

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

### **Lenguaje ASP**

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitio web dinámicos. ASP significa en inglés (Active Server Pages), fue liberado por Microsoft en 1996. Las páginas web desarrolladas bajo este lenguaje es necesario tener instalado Internet Information Server (IIS).

ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl and Jscript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

### **Lenguaje ASP.NET**

Este es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET.

El ASP.NET fue desarrollado para resolver las limitantes que brindaba tu antecesor ASP. Creado para desarrollar web sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Los archivos cuentan con la extensión (aspx). Para su funcionamiento de las páginas se necesita tener instalado IIS con el Framework .Net. Microsoft Windows 2003 incluye este framework, solo se necesitará instalarlo en versiones anteriores.

### **Lenguaje JSP**

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor.

JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

### **Lenguaje Python**

Es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

Su código no necesita ser compilado, por lo que se llama que el código es interpretado. Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- Programación orientada a objetos.
- Programación estructurada.
- Programación funcional.
- Programación orientada a aspectos.

## **Lenguaje Ruby**

Es un lenguaje interpretado de muy alto nivel y orientado a objetos. Desarrollado en el 1993 por el programador japonés Yukihiro “Matz” Matsumoto. Su sintaxis está inspirada en Python, Perl. Es distribuido bajo licencia de software libre (Open-source).

Ruby es un lenguaje dinámico para una programación orientada a objetos rápida y sencilla. Para los que deseen iniciarse en este lenguaje pueden encontrar un tutorial interactivo de Ruby. Se encuentra también a disposición de estos usuarios un sitio con informaciones y cursos en español.

## 1.4 SISTEMA OPERATIVO WINDOWS

Es el sistema operativo de mayor difusión entre computadoras personales y servidores pequeños y medianos. Fue desarrollado por Microsoft, aunque muchas de sus ideas básicas provienen del sistema operativo Mac OS, de los equipos Macintosh.

### Historia

Las primeras versiones del Sistema Operativo de Microsoft recibieron la denominación de DOS (Disk Operating System – Sistema Operativo de Disco), software que trabajaba en ambiente de texto, es decir que las instrucciones o comandos debían de ser escritas desde teclado.

La primera versión gráfica exitosa del Sistema Operativo de Microsoft fue Windows 3.1 para equipos stand alone (aislados), que en realidad eran un ambiente gráfico que trabajaba sobre DOS (era una interfaz gráfica para DOS). Recibió el nombre de Windows pues la información se presentaba dentro de espacios rectangulares de la pantalla, con bordes a modo de ventanas.

Posteriormente apareció una versión mejorada de Windows, identificada como Windows 3.11, que básicamente tenía la misma funcionalidad que Windows 3.1, pero permitía trabajar al equipo en esquemas básicos de red, por lo que la identificación formal utilizada para describirlo fue Windows para Trabajo en Grupo.

El primer Sistema Operativo Gráfico de Microsoft para estaciones de trabajo (clientes dentro de una red) fue Windows 95, que ya no requería de DOS para su funcionamiento, aunque estaba en capacidad de reconocer y ejecutar un gran número de comandos del viejo DOS. Además fue el primer sistema Operativo de 32 bits para clientes tipo PC.

La siguiente versión del Sistema Operativo de Microsoft para estaciones de trabajo fue Windows 98, luego apareció Windows 2000 y actualmente se encuentran vigentes Windows XP y Windows 2003.

Las versiones correspondientes de sistema operativo para servidores son Windows NT, Windows 2000 Server, Windows XP Server y Windows 2003 Server.

Cada nueva versión del Sistema Operativo presenta más funcionalidades que la versión previa, y generalmente es compatible con la versión anterior (las aplicaciones que funcionaban en la versión antigua también funciona en la versión nueva).

### **1.3.7. SEGURIDAD DE WINDOWS**

Hace muchos años que la seguridad de Windows sigue causando controversias en todo el mundo. Todas las versiones de los Sistemas Operativos Windows han tenido graves fallas en la seguridad, teniendo que recurrir a parches o service packs para corregir estos errores. De hecho la cuestión de la seguridad es el tema que más destacan los competidores de Windows para alentar a los usuarios a cambiar de Sistema Operativo.

Lo cierto es que Windows, al ser el Sistema Operativo más popular, es el blanco más fácil y obvio para millones de piratas informáticos alrededor del mundo. Con tantos usuarios malintencionados, existen más probabilidades de que al sistema le sean encontrados errores y agujeros de seguridad. La popularidad trae aparejada estos problemas.

De todas maneras no es un justificativo para tanta cantidad de problemas. Existen otros Sistemas Operativos populares que incluso son de código abierto y que no tienen semejantes errores (que tenga el código abierto significa que pueden ser descubiertos agujeros de seguridad de forma mucho más fácil; pues cualquiera puede ver el “ADN” del sistema).

Sus problemas de seguridad obligan a Microsoft a desarrollar herramientas de actualizaciones, incluso publican de forma mensual parches de seguridad.

De todas maneras, la mayor parte de los problemas de seguridad provienen de la inexperiencia e ingenuidad de sus usuarios.

### **1.3.8. Windows Server 2003.**

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

En términos generales, Windows Server 2003 se podría considerar como un Windows XP modificado, no con menos funciones, sino que estas están deshabilitadas por defecto para obtener un mejor rendimiento y para centrar el uso de procesador en las

características de servidor, por ejemplo, la interfaz gráfica denominada *Luna* de Windows XP viene desactivada y viene con la interfaz clásica de Windows. Sin embargo, es posible volver a activar las características mediante comandos *services.msc*.

En la tabla 1.2 se muestran algunas especificaciones de este sistema operativo.

**Tabla 1.2** *Windows Server 2003*

Desarrollador	Microsoft
Modelo de desarrollo	Software propietario
Última versión estable	5.2 Service Pack 2 (5.2.3790.3959)
Núcleo (Kernel)	NT 5.2
Tipo de núcleo	Híbrido
Plataformas soportadas	IA-32 X86-64 Itanium
Método de actualización	Windows update
Licencia	Microsoft CLUF (EULA)
Idiomas	Multilinguaje
Soporte Técnico	Período estándar- 13 de julio de 2010 Periodo extendido – 14 de julio de 2015

### Características

A continuación se detallan las principales características del sistema operativo Windows Server 2003.

- Sistema de archivos NTFS (NT File system)
  - Definir cuotas por usuario
  - Cifrado y compresión de archivos, carpetas y no unidades completas.
  - Permite desmontar dispositivos de almacenamiento sobre sistemas de archivos de otros dispositivos al estilo UNIX.
- Gestión de almacenamiento, backups... incluye gestión jerárquica de almacenamiento, consiste en utilizar un algoritmo de caché para pasar los datos menos usados de discos duros a medios ópticos o similares más lentos, y volverlos a leer a disco duro cuando se necesitan.

- Windows Driver Model: Implementación básica de los dispositivos más utilizados, de esa manera los fabricantes de dispositivos sólo han de programar ciertas especificaciones de su hardware.
- ActiveDirectory Directorio de organización basados en LDAP (Lightweight Directory Access Protocol), permiteme gestionar de forma centralizada la seguridad de una red corporativa a nivel local.
- Autenticación Kerberos 5.
- DNS con registro de IP's dinámicamente
- Políticas de seguridad

### **Versiones.**

*Web Edition.*- Diseñado para los servicios y el hospedaje web

*Standard Edition.*- El más versátil de todos, ofrece un gran número de servicios útiles para empresas de cualquier tamaño.

*Enterprise Edition.*- Para empresas de mayor tamaño que la Standard Edition.

*Datacenter Edition.*- Para empresas que requieran bases de datos más escalables y un procesamiento de transacciones de gran volumen.

### **Beneficios**

En la tabla 1.3 se describen los beneficios principales de Windows Server 2003

Tabla 1.3 Beneficios de Windows Server 2003

Beneficio	Descripción
Seguro	<p>Windows Server 2003 es el sistema operativo de servidor más rápido y más seguro que ha existido. Ofrece fiabilidad al:</p> <ul style="list-style-type: none"> <li>• Proporcionar una infraestructura integrada que ayuda a asegurar que su información de negocios estará segura.</li> <li>• Proporcionar fiabilidad, disponibilidad y escalabilidad para que usted pueda ofrecer la infraestructura de red que los usuarios solicitan.</li> </ul>
Productivo	<p>Windows Server 2003 ofrece herramientas que le permiten implementar, administrar y usar su infraestructura de red para obtener una productividad máxima.</p> <ul style="list-style-type: none"> <li>• Proporcionar herramientas flexibles que ayuden a ajustar su diseño e implementación a sus necesidades organizativas y de red.</li> <li>• Ayudarle a administrar su red proactivamente al reforzar las políticas, tareas automatizadas y simplificación de actualizaciones.</li> <li>• Ayudar a mantener bajos los gastos generales al permitirles a los usuarios trabajar más por su cuenta.</li> </ul>
Conectado	<p>Windows Server 2003, puede ayudarle a crear una infraestructura de soluciones de negocio para mejorar la conectividad con empleados, socios, sistemas y clientes.</p> <ul style="list-style-type: none"> <li>• Proporcionar un servidor Web integrado y un servidor de transmisión de multimedia en tiempo real para ayudarle a crear más rápido, fácil y seguro una Intranet dinámica y sitios de Internet.</li> <li>• Proporcionar un servidor de aplicaciones integrado que le ayude a desarrollar, implementar y administrar servicios Web en XML más fácilmente.</li> <li>• Brindar las herramientas que le permitan conectar servicios web a aplicaciones internas, proveedores y socios.</li> </ul>
Mejor economía	<p>Windows Server 2003, cuando está combinado con productos Microsoft como hardware, software y servicios de los socios de negocios del canal brindan la posibilidad de ayudarle a obtener el rendimiento más alto de sus inversiones de infraestructura.</p> <ul style="list-style-type: none"> <li>• Proporcionar una guía preceptiva y de fácil uso para soluciones que permitan poner rápidamente la tecnología a</li> </ul>

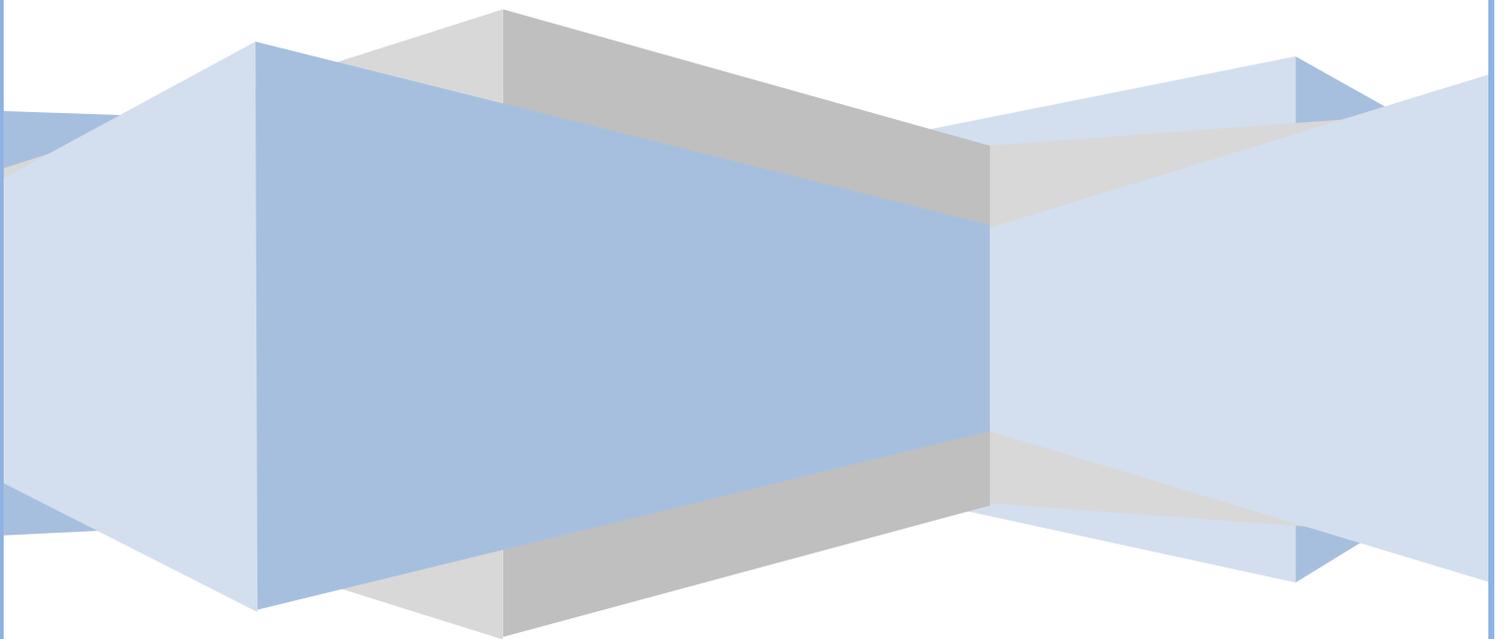
	<p>trabajar.</p> <ul style="list-style-type: none"><li>• Ayudarle a consolidar servidores aprovechando lo último en metodologías, software y hardware para optimizar la implementación de su servidor.</li><li>• Bajar el costo total de propiedad (TCO) para recuperar rápido la inversión.</li></ul>
--	--

Todos los puntos desarrollados en el presente capítulo de éste trabajo, sirven como marco teórico para poder visualizar el sistema que se pretende desarrollar en base a las necesidades y requerimientos que sean proporcionados por el cliente, por lo cual servirá como base para poder analizar, diseñar e implementar dicho sistema.

En el siguiente capítulo se realizará el análisis y diseño de un sistema lo suficientemente sencillo y eficiente, tomando en consideración las necesidades del cliente, el tipo de software que se pueda utilizar para poder plantear la metodología que más convenga para lograr dicho diseño.

# Capítulo 2

## Desarrollo del Sistema



# CAPÍTULO 2



## 2.1. ANÁLISIS

El paso del análisis es previo al desarrollo, en este paso se tendrá que analizar las características y condiciones del sistema para llevar a cabo los objetivos planteados al inicio del proyecto. Se debe tener en cuenta las diferentes condiciones en las que se plantea el problema para obtener una mejor solución. Además de no perder de vista las necesidades de los usuarios.

Teniendo en cuenta las necesidades de los usuarios y las condiciones del sistema además del software y hardware con el que se cuenta para el desarrollo del proyecto se podrá tomar la mejor decisión para su desarrollo y su implementación, adaptando todo lo que se tiene para un desarrollo y funcionamiento óptimo.

Partiendo de la necesidad de la División de Ingeniería en Ciencias de la Tierra de la Facultad de Ingeniería de la UNAM (DICT) de crear un directorio de su personal docente que sea consultado por sus alumnos en Internet de una forma sencilla y segura, el área de Sistemas de dicha división les plantea la necesidad de crear un "*Sistema integral para la gestión del personal*", que cumpla con las necesidades de la DICT, pero que al mismo tiempo pueda ser administrado con la misma sencillez, tomando en cuenta que gran parte

del personal que labora en dicha división posee los conocimientos básicos de computación.

Con base en los requerimientos de la DICT se propuso un sistema que permitirá el acceso desde Internet a tres niveles de usuarios, de los cuales dos serán identificados por el sistema mediante un usuario (login) y una contraseña (password), que serán validadas de forma dinámica mediante programación en PHP. Una vez identificados, el sistema les permitirá realizar diferentes acciones dependiendo los privilegios con los que cuente el usuario (Tabla 2.1).

**Tabla 2.1** Privilegios de los usuarios

<b>Nivel de usuario</b>	<b>Descripción</b>
<i>Visitante</i>	Se le otorgará el permiso exclusivo de consulta.
<i>Propietario</i>	Contará con permiso de consulta sobre todo el directorio y con privilegios para hacer modificaciones en el mismo, tales como la creación de nuevos usuarios; así como la inserción, modificación y eliminación de información correspondiente al personal perteneciente a su departamento. Necesita identificarse en el sistema.
<i>Administrador del Sistema</i>	Tendrá privilegios de crear nuevas coordinaciones y nuevos usuarios, así como permisos para la modificación y consulta de la información de todo el directorio. Necesita identificarse en el sistema.

### 2.1.1 DELIMITACIÓN DEL SISTEMA

En esta parte se debe considerar las diferentes estrategias para la solución del problema planteado por los usuarios del sistema, además de los elementos con los que se cuenta para el desarrollo del mismo. Otro punto importante es la forma en que la Base de Datos tiene que estar diseñada para evitar la menor cantidad de errores posible en la comunicación entre los diferentes bloques del sistema, esto es, que la Base de Datos no tenga conflictos entre la consulta y la inserción de los datos a esta.

La parte más importante para delimitar el sistema depende de la cantidad de usuarios que va a tener y de la cantidad de información que se le pretende insertar a la Base del

Sistema, con el fin de hacerla más funcional, y tener en cuenta todas las necesidades tanto de los recursos del sistema como de los usuarios.

Al realizar la delimitación del sistema se logra visualizar los alcances del mismo. Por ello que es importante identificar de las necesidades del cliente, para poder definir las funciones que debe realizar el sistema y con ello lograr representar el comportamiento del mismo, lo cual sirve para mejorar la manejabilidad de éste y con ello llegar a la etapa del diseño.

A continuación se detalla lo anteriormente expuesto.

### **Identificación de las necesidades del Cliente**

La DICT solicitó diseñar el sistema de acuerdo al análisis presentado y con base en las siguientes necesidades.

- Es muy difícil para los alumnos localizar a sus profesores fuera de sus horarios de clase.
- La DICT cuenta con un directorio electrónico, sin embargo éste no es un sistema, si no simplemente es una página web estática (programada en HTML), lo cual hace que éste no sea actualizado constantemente ya que no esta a la disposición de todo el personal de la división.
- Por otro lado el personal docente de la DICT tiene la necesidad de tener alguna información adicional de otros profesores, para la realización de diferentes proyectos o prácticas, por lo cual resulta complicado obtener dicha información cuando se encuentran fuera de las instalaciones de la división, o bien fuera de la ciudad.

Después de establecer las necesidades del cliente, es necesario establecer que es lo que podrá hacer el sistema, por lo cual hay que establecer la funcionalidad del mismo.

### **Asignación de Funciones**

En el sistema propuesto precisa contar con una interfaz Web, que maneje tres roles, los cuales son: Administrador, Propietario y Visitante.

De éstos roles se pretende que dos de ellos cuenten con ciertos privilegios y que por lo tanto requieran identificarse ante el sistema. Éstos son: un usuario Administrador de la

Base de Datos, que será el único que tendrá la capacidad de ver y modificar la información de toda la Base de Datos; y usuarios Propietarios de los diferentes departamentos con que cuenta la DICT, los cuales están diseñados para sólo insertar información de su correspondiente departamento, además de modificar la información del mismo. Sin importar a qué departamento pertenezca toda la información de la Base de Datos será visible para los propietarios.

Es importante mencionar que, para ingresar la información al sistema, se deberá contar con cierta información, la cual se divide en obligatoria y opcional (tabla 2.2), de la cual las personal sin privilegios podrán ver sólo la información que se ha catalogado como pública (tabla 2.3).

**Tabla 2.2** Información requerida por el Sistema

<b>Obligatorio</b>		<b>Opcional</b>
<ul style="list-style-type: none"> <li>• Apellido paterno</li> <li>• Apellido materno</li> <li>• Nombre</li> <li>• Nacionalidad</li> <li>• R.F.C.</li> <li>• CURP</li> <li>• Sexo</li> <li>• Edad</li> <li>• Fecha de nacimiento</li> <li>• No. Trabajador</li> <li>• Departamento</li> <li>• Área de conocimiento (licenciatura y/o posgrado)</li> <li>• Dirección UNAM</li> <li>• Dirección particular</li> <li>• Tel. Particular</li> <li>• Tel. Oficina</li> </ul>	<ul style="list-style-type: none"> <li>• Curriculum</li> <li>• Fotografía</li> <li>• E-mail</li> <li>• Condición</li> <li>• Grado</li> <li>• Categoría y nivel</li> <li>• Antigüedad</li> <li>• Ced. Profesional</li> <li>• Fecha de ingreso</li> <li>• Periodo de contratación</li> <li>• Licencia, sabático o permisos</li> <li>• Plaza</li> <li>• Horas</li> <li>• Asignatura(s)</li> <li>• Título del puesto</li> <li>• Perfil mínimo del puesto</li> <li>• Funciones mínimas del puesto</li> <li>• Categoría de cómputo</li> </ul>	<ul style="list-style-type: none"> <li>• Página web</li> <li>• Tel. Celular</li> <li>• Folio</li> <li>• Cod. Prog.</li> <li>• No. Estacionamiento</li> </ul>

**Tabla 2.3** Información visible para usuarios sin privilegios en el Sistema

<b>Información Pública</b>
<ul style="list-style-type: none"> <li>• Nombre</li> <li>• Departamento</li> <li>• Ubicación (UNAM)</li> <li>• Categoría y Grado</li> <li>• Teléfono oficina</li> <li>• Correo</li> </ul>

Una vez delimitado el sistema, se estableció la interacción hombre máquina que tendrá el sistema para una mejor manejabilidad del mismo, para ello se realizó la propuesta de diferentes prototipos que fueron mostrados al jefe de la DICT para delimitar y cubrir las necesidades requeridas de la manera más eficiente posible. Todo esto dio origen a la etapa del diseño, que se detalla a continuación.

## **2.2 DISEÑO**

La parte de diseño es importante, dado que es la base para la implementación de un sistema, para ello se debe tener en cuenta las características mencionadas en la parte del análisis. Por ello en esta etapa se hablará de todas aquellas herramientas o recursos necesarios para el desarrollo y funcionamiento del sistema, como lo es el software utilizado para la programación, el necesario para el diseño y creación de la Base de Datos y la implementación del servidor.

### **2.2.1. PROPUESTA DE SOFTWARE**

Debido a que desde el inicio del desarrollo de la propuesta del sistema se planteo que éste fuera utilizado en línea, la propuesta de software debe de tomar en cuenta el software que satisfaga tales necesidades. Por lo anterior se tomó la decisión de utilizar el siguiente software:

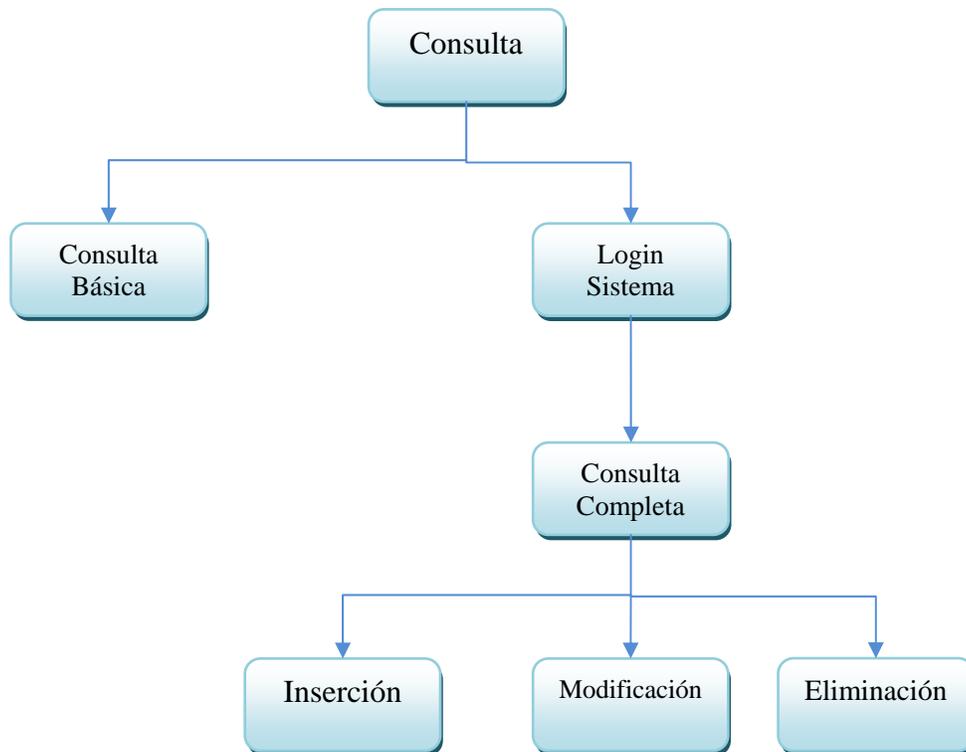
- El sistema operativo Windows Server 2003 como plataforma de desarrollo, dado que éste es robusto, amigable, seguro y sencillo de manejar, tomando en cuenta que la DICT no cuenta con personal especialista en servidores implementados en otras plataformas.
- Como lenguaje de programación se eligió PHP, por varias razones, entre las que destacan que es software libre, es compatible con el Sistema Operativo seleccionado y además es portable.

- El desarrollo del sistema se eligió hacer en Dreamweaver versión MX creado por Macromedia, dado que permite la creación de páginas dinámicas, los sistemas creados son soportados por la mayoría de los navegadores Web existentes en la actualidad, además permite una sencilla conexión con MySQL sin necesidad de tener mucha experiencia previa en programación..
- Por lo anterior se decidió utilizar el manejador de base de datos a MySQL.
- Se eligió phpMyAdmin como interfaz para la utilización del manejador de base de datos ya que es un software libre y es una aplicación web muy sencilla.
- Para el servidor web se eligió Apache en su versión 2, por ser compatible con el Sistema Operativo además de ser software libre.

El sistema presentado en esta tesis se basa en su totalidad en las herramientas expuestas anteriormente.

### **2.2.2. DIAGRAMA JERÁRQUICO FUNCIONAL DEL SISTEMA**

La relación que existe entre cada una de las diferentes secciones del sistema se representan mediante un diagrama jerárquico funcional (fig. 2.1), esto con el fin de que cada uno de los módulos sea lo más funcional para éste.



**Fig. 2.1** Diagrama Jerárquico Funcional General

Para el caso de la aplicación se tienen básicamente dos módulos uno para que las personas puedan hacer la consulta sin necesidad de tener privilegios en el sistema, lo que en la etapa del análisis se determinó que sería la consulta de información pública, y por otro lado se tiene el módulo en el que, al loguearse, el sistema reconocerá al usuario y permitirá ciertas actividades a realizar que son definidas de acuerdo a los privilegios que tenga dicha cuenta.

Es por ello que, como se observa en la figura 2.1, el usuario inicialmente visualiza una interfaz, llamada Consulta y a partir de ésta podrá ingresar a la consulta básica (publica) del sistema, o a la consulta completa teniendo la posibilidad de ingresar con un login y password para identificar permisos dentro del sistema y habilitar las funcionalidades correspondientes. El acceso, éste proporciona tres tipos de menús adicionales (Inserción, Modificación y Eliminación), donde podrán manipular la información de los diferentes departamentos.

### 2.2.3. DISEÑO DE LA BASE DE DATOS

El diseño de la Base de Datos partió de la necesidad de la División de Ingeniería en Ciencias de la Tierra de almacenar la mayor cantidad de información referente a las personas que laboran en la misma.

Para ello se propondrá usar una base de datos, que almacene toda la información requerida, así como los usuarios pertenecientes al sistema. Esta base de datos permitirá al usuario la manipulación de toda la información por medio de Internet, dependiendo de los permisos que se tenga en el sistema.

La estructura de la base de datos estará dividida en tres regiones definidas para la administración y manipulación del sistema. Dichas regiones son:

- **CONTROL.** En esta región se controlará a los usuarios que ingresen al sistema, además de la información que se muestre a éstos (fig 2.2)
- **PÚBLICA.** En esta región se mostrara la información de la Base de Datos que es catalogada como "pública" y no les es permitido la modificación o eliminación de la información alojada en la Base (fig 2.3).
- **CONFIDENCIAL.** Esta región, además de contener la información de la región pública y compartir con ella ciertas tablas, consultará y manipulará toda la información que será de uso exclusivo para los usuarios que cuenten con una sesión en el sistema, de acuerdo a los privilegios con los que cuente (fig 2.4).

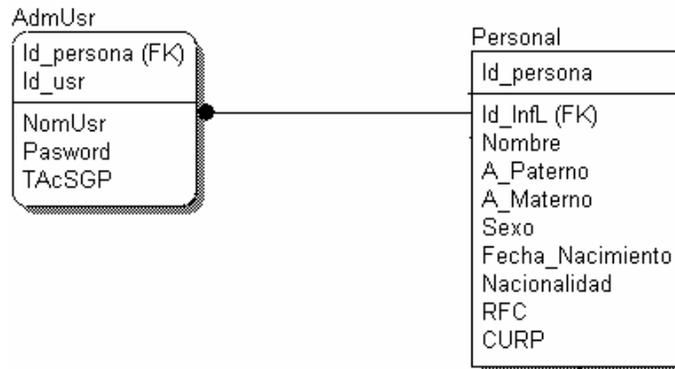


Fig. 2.2 Diagrama entidad-relación de la región de control

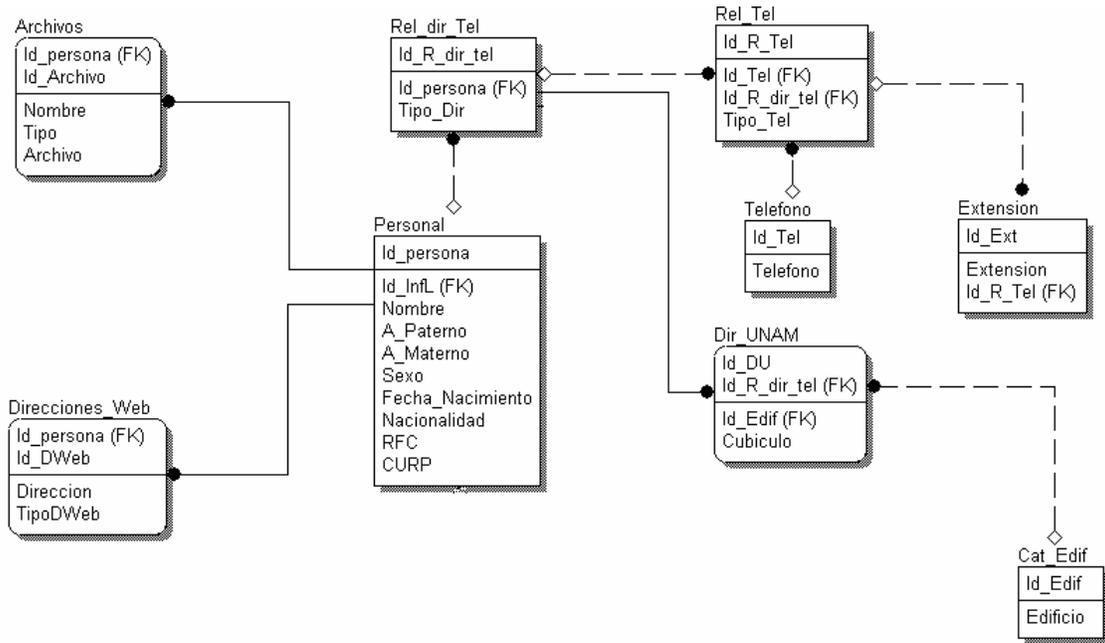


Fig. 2.3 Diagrama entidad-relación de la región pública

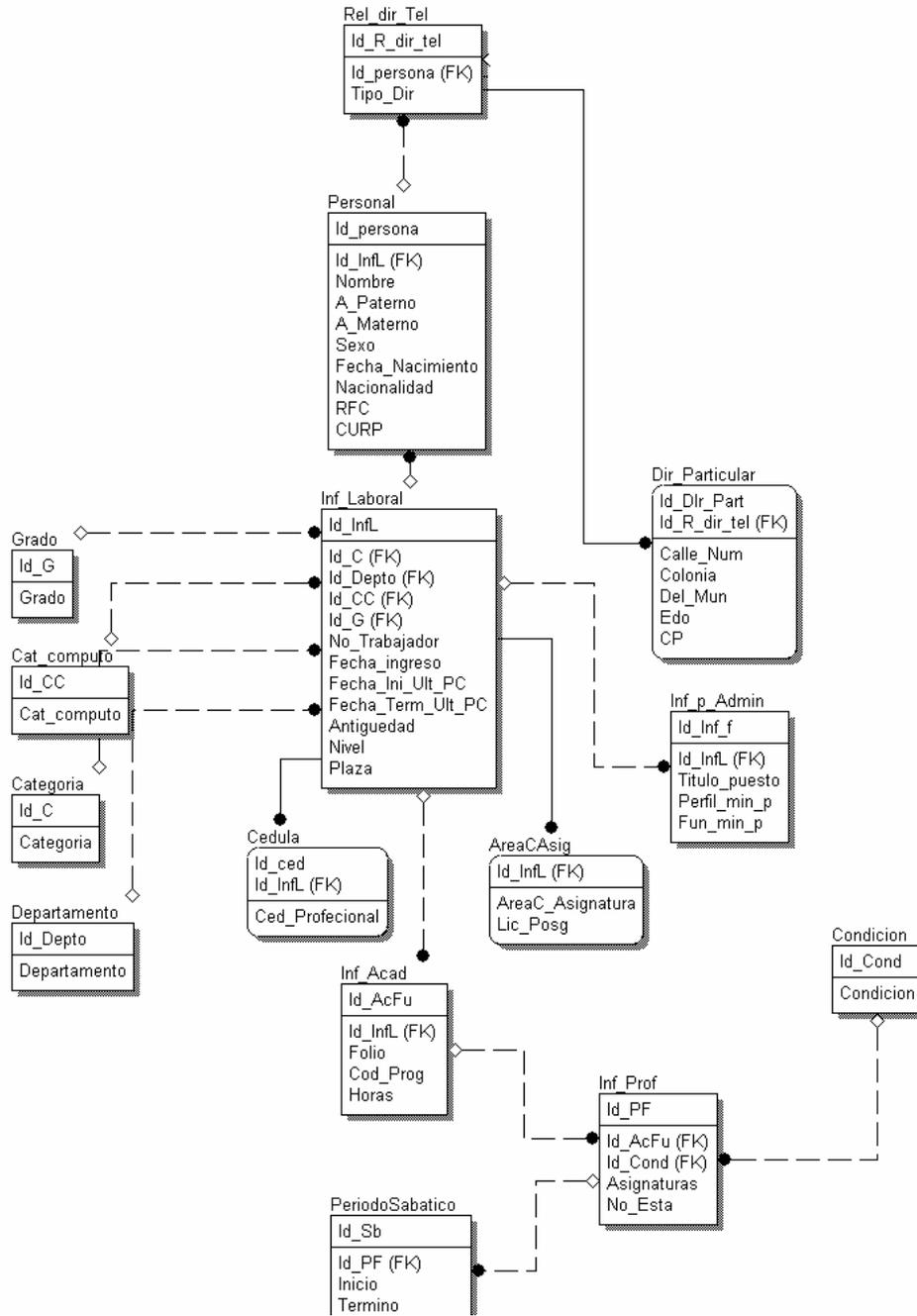


Fig. 2.4 Diagrama entidad-relación de la región confidencial

**Definición de Entidades**

Teniendo en cuenta el diagrama del Diseño de la Base de Datos y la separación de las regiones del mismo, la definición de entidades propuesta para cubrir las necesidades de la DICT se muestra desglosada en las tablas 2.4, 2.5 y 2.6.

En la tabla 2.4 se muestran los atributos de las entidades que conforman la región de control, los cuales engloban la información básica de los usuarios registrados en el sistema y que tendrán algún privilegio sobre el mismo.

En la tabla 2.5 se muestran los atributos de las entidades que conforman la región pública, que contendrá la información mínima general que será mostrada a todos los usuarios finales del sistema, sin importar que tengan o no privilegios sobre el sistema; es por ello que dicha información fue catalogada como “pública” y no puede ser modificada por cualquier usuario.

En la tabla 2.6 se muestran los atributos de las entidades que conforman la región confidencial, los cuales complementan la región anterior para así lograr el almacenamiento total y confidencial de la información del personal de la DICT

**Entidades del Sistema**

Tabla 2.4 REGION DE CONTROL

<b>Nombre</b>	<b>Atributos</b>	<b>Nombre</b>	<b>Atributos</b>
AdmUsr	Id_persona (FK) Id_usr NomUsr Password TAcSGP	Personal	Id_persona Id_InfL(FK) Nombre A_Paterno A_Materno Sexo Fecha_Nacimiento Nacionalidad RFC CURP

Tabla 2.5 REGION PÚBLICA

<b>Nombre</b>	<b>Atributos</b>	<b>Nombre</b>	<b>Atributos</b>
Archivos	Id_persona (FK) Id_Archivo Nombre Tipo Archivo	Rel_dir_Tel	Id_R_dir_tel Id_persona (FK) Tipo_Dir
Personal	Id_persona Id_InfL(FK) Nombre A_Paterno A_Materno Sexo Fecha_Nacimiento Nacionalidad RFC CURP	Rel_Tel	Id_R_Tel Id_Tel (FK) Id_R_dir_tel (FK) Tipo_Tel
Extension	Id_Ext Extension Id_R_Tel (FK)	Telefono	Id_Tel Telefono
Dir_UNAM	Id_DU Id_R_dir_tel (FK) Id_Edif (FK) Cubiculo	Direcciones_Web	Id_persona (FK) Id_DWeb Direccion TipoDWeb
Cat_Edif	Id_Edif Edificio		

Tabla 2.6 REGION CONFIDENCIAL

<b>Nombre</b>	<b>Atributos</b>	<b>Nombre</b>	<b>Atributos</b>
Rel_dir_Tel	Id_R_dir_tel Id_persona (FK) Tipo_Dir	Personal	Id_persona Id_InfL(FK) Nombre A_Paterno A_Materno Sexo Fecha_Nacimiento Nacionalidad RFC CURP
Grado	Id_G Grado	Cat_computo	Id_CC Cat_computo
Categoria	Id_C Categoria	Departamento	Id_Depto Departamento
Inf_Laboral	Id_InfL Id_C (FK) Id_Depto (FK)	Dir_Particular	Id_Dir_Part Id_R_dir_tel (FK)

	Id_CC (FK) Id_G (FK) No_Trabajador Fecha_ingreso Fecha_Ini_Ult_PC Fecha_Term_Ult_PC Antigüedad Nivel Plaza		Calle_Num Colonia Del_Mun Edo CP
Cedula	Id_ced Id_InfL (FK) Ced_Profesional	Inf_Acad	Id_AcFu Id_InfL (FK) Folio Cod_Prog Horas
AreaCAsig	Id_InfL (FK) AreaC_Asignatura Lic_Posg	Inf_p_Admin	Id_Inf_f Id_InfL (FK) Titulo_puesto Perfil_min_p Fun_min_p
PeriodoSabatico	Id_Sb Id_PF (FK) Inicio Termino	Condicion	Id_Cond Condicion
Inf_Prof	Id_PF Id_AcFu (FK) Id_Cond (FK) Asignaturas No_Esta		

### ***Diseño Conceptual y Físico***

En las figuras 2.5 y 2.6 se observa el diagrama conceptual de la base de datos, en este diseño se muestra la relación, tanto de las regiones como de las entidades previamente definidas, para con ello establecer la integridad de la base de datos.



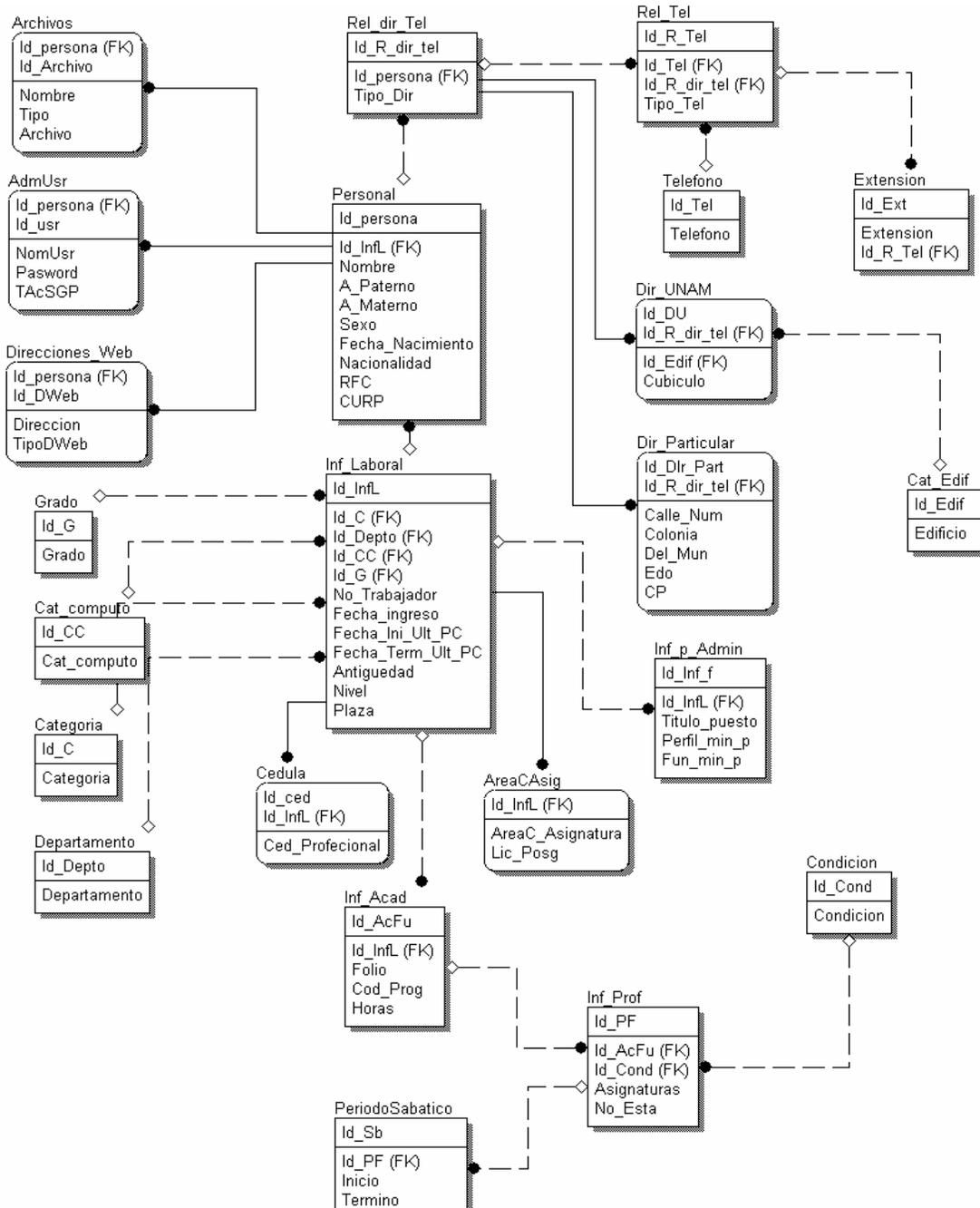
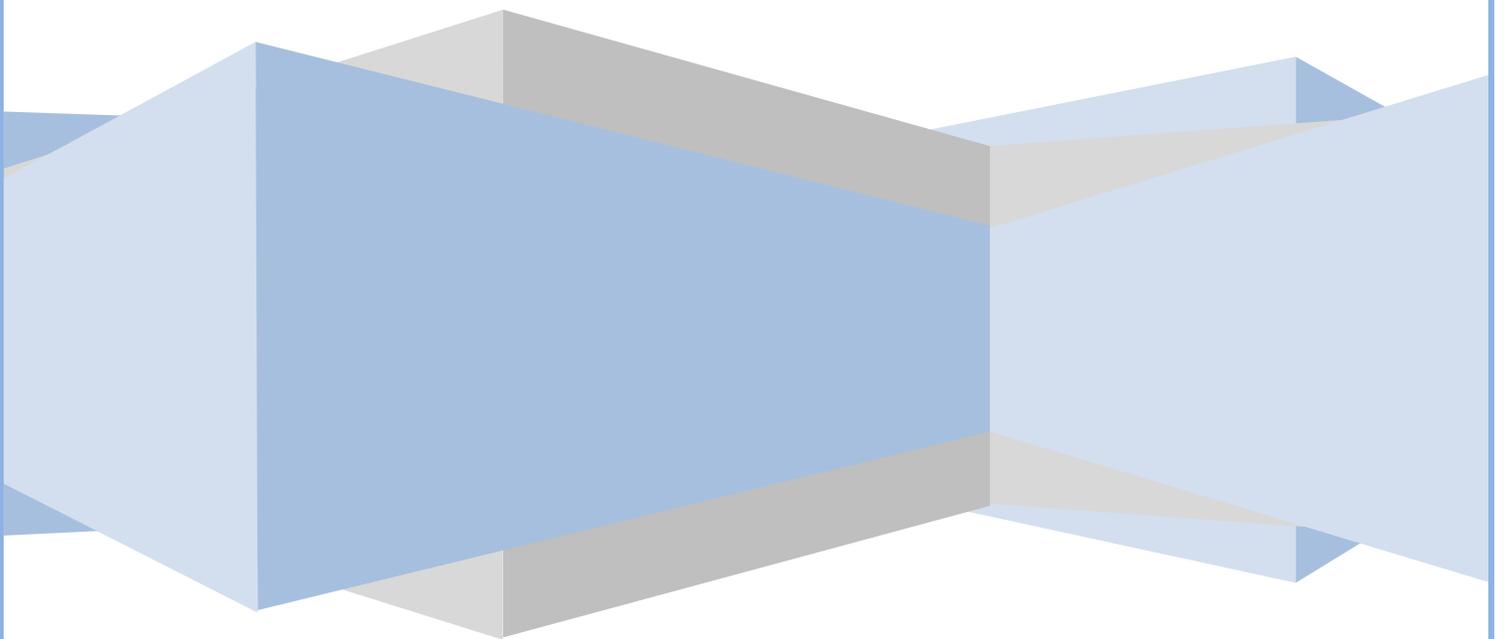


Figura 2.6 Diseño Lógico

Una vez analizado y planteado el diseño del sistema, se continúa con la implementación del mismo. Para ello en el siguiente capítulo se hablará de las especificaciones del hardware y software que se utilizará, además de la creación tanto la base de datos del sistema, como las interfaces que se manejarán a lo largo del mismo de acuerdo a los privilegios con los que cuente el usuario.

# Capítulo 3

## Implementación



# *CAPÍTULO 3*



## *3.1 INTRODUCCIÓN*

En la etapa de Implementación se dan a conocer los puntos principales para la creación del sistema, partiendo de las especificaciones de hardware junto con el sistema operativo que se utilizará, de acuerdo a los recursos con los que cuenta en la División de Ingeniería en Ciencias de la Tierra (DICT) en el momento de plantear el problema. Además de explicar la creación de la Base de Datos junto con la interfaz de usuario.

De acuerdo a lo explicado en el capítulo anterior referente al software utilizado y diseño planteado en base a los requerimientos realizados, se utilizó como sistema operativo Windows Server 2003, además de que la implementación se realizó con un lenguaje de programación PHP 5, manejador de Base de Datos MySQL y con el servidor web Apache 5.

A continuación se detallara más esta etapa abarcando las herramientas utilizadas, así como el tipo de seguridad manejado en el sistema.

### 3.1.1 ESPECIFICACIONES DEL HARDWARE

En la instalación de Windows Server 2003, fue necesario conocer los requerimientos mínimos de hardware previos a la instalación del Sistema Operativo, todo esto con el fin de evitar fallas, tanto en la implementación del proyecto como en su uso. Las características principales son las siguientes:

	Windows Server 2003 Web Edition	Windows Server 2003 Standard Edition	Windows Server 2003 Enterprise Edition	Windows Server 2003 Datacenter Edition
Nº Máximo de procesadores soportados	2	4	8	64
Memoria máxima	2 GB	4 GB	32 GB (32 bit) 64 GB (64 bit)	32 GB (32 bit) 512 GB (64 bit)
Requerimientos del sistema recomendados	Procesador a 550 Mhz, 256 MB RAM, 1.5 GB de espacio en disco	Procesador a 550 Mhz, 256 MB RAM, 1.5 GB de espacio en disco	Procesador a 550 Mhz, 256 MB RAM, 1.5-2.0 GB de espacio en disco	Procesador a 550 Mhz, 1 GB RAM, 1.5 GB de espacio en disco

Teniendo en cuenta las características enunciadas anteriormente se tomó la decisión de tener el servidor con la versión de Windows Server 2003 Standard Edition y en una computadora con las siguientes características:

DISPOSITIVO	CARACTERÍSTICAS
Marca	DELL
Procesador	Intel Core 2, 1.86 GHz
Memoria RAM	1.00 GB
Disco duro	120 GB

### 3.1.2 SEGURIDAD DEL SERVIDOR

La instalación del sistema operativo Windows Server se hizo satisfactoriamente basándose en los manuales de operación y manuales publicados en la red. La seguridad del servidor se basó en la que tiene Apache y la programación que se utilizó en el sistema dado que se utiliza un usuario para cada uno de los diferentes bloques del sistema, esto con el fin de que no se puedan descifrar con facilidad las cuentas de usuario junto con sus contraseñas para proporcionar a la información mayor seguridad. Para la creación de la Base de Datos la configuración corrió a cargo de PHP y MySQL. Dando de esta forma por terminado la parte de la configuración para la seguridad del servidor y con ello la seguridad del sistema.

## 3.2 CREACIÓN DE LA BASE DE DATOS

El funcionamiento del sistema parte de una base de datos llamada control, como se mencionó anteriormente en el capítulo 2 de esta tesis, la cual ayuda a administrar e identificar a los usuarios que ingresen al mismo. A través de ésta el sistema reconoce el tipo de usuario registrado para con ello proporcionarle los distintos tipos de consulta existentes, ya sea de la base Principal o la Confidencial.

Posteriormente, dependiendo del tipo de privilegios con los que cuenten los usuarios registrados, se podrá manipular la base llamada Confidencial, en la cual podrán consultar, modificar, insertar o eliminar tablas de la misma.

La forma en que el sistema interactúa con esta Base de Datos se explica a continuación (tabla 3.1)

Tabla 3.1 Código para la conexión del sistema a la BD

<b>Código</b>	<b>Descripción</b>
<pre> &lt;?PHP //Definicion objeto ConexionDB  class ConexionBD {      protected \$ENLACE;     protected \$BASE;      //infirmacion de acceso a la base     private \$USR;     private \$PW;     private \$HOST;      function __construct(\$BD,\$Usuario,\$Password)     {         \$this-&gt;HOST = "localhost";         \$this-&gt;USR =\$Usuario;         \$this-&gt;PW = \$Password;         \$this-&gt;Conectar(\$BD);     }      private function Conectar(\$BD)     {         \$this-&gt;BASE = \$BD;          if      (!(\$this-&gt;ENLACE=mysql_connect(\$this- &gt;HOST,\$this-&gt;USR,\$this-&gt;PW))) //validacion de conexion             die("Error conectando a la base de datos.");         else      if(!mysql_select_db(\$this-&gt;BASE,\$this- &gt;ENLACE)) //validacion a la base de datos             die("Error seleccionando la base de datos.");     }      function Get_Link()     {         return \$this-&gt;ENLACE;     }      function __destroy()     {         if(!mysql_close(\$this-&gt;ENLACE)) echo "ERROR en la desconexion de la base.";     } } ?&gt; </pre>	<p>En esta parte del sistema lo que hace el código es la conexión a la base de datos utilizando las variables estándar del manejador de Base de Datos utilizado (phpMyAdmin)</p> <p>Esta conexión se realizó mediante programación orientada a objetos de dichas variables utilizando el lenguaje de programación elegido (php)</p>

### 3.3 CREACIÓN DE LA INTERFAZ DE USUARIO

La parte fundamental del sistema es la interfaz de usuario dado que es la parte visual y la que el usuario utiliza, la interfaz está dividida de tal forma que sea fácil para el usuario su utilización en la inserción y consulta de información (tabla 3.2).

El acceso al Sistema (tabla 3.3) dependerá de los privilegios que se tenga como usuario, dado que las personas que no cuentan con privilegios en el Sistema solo podrán visualizar la información básica del personal de la DICT alojada en la Base de Datos.

La consulta, dependerá de la forma en la que se acceda al Sistema esto es, si no se cuentan con privilegios solo se podrá tener una consulta pública, pero si se cuentan con privilegios se podrá tener una consulta completa, además de poder insertar nueva información o mas información de las personas que existentes en la Base, también se puede eliminar la información que así se requiera

Tabla 3.2 Código de la librería de funciones

<i>Código</i>	<i>Descripción</i>
<pre> &lt;? /* Libreria de funciones complementarias al Sistema de Gestion de Personal */  /*      Inicicion de sesion y declaracion de variables de sesion      */ session_start();  if(!session_is_registered('Sesion_Act'))     {     session_register('Sesion_Act');     \$_SESSION['Sesion_Act']=true;     }  if(!session_is_registered('TipoUSR'))     {     session_register('TipoUSR');     \$_SESSION['TipoUSR']="C";     }  if(!session_is_registered('AccesoSist'))     {     session_register('AccesoSist');     \$_SESSION['AccesoSist'] = false;     } </pre>	<p>La librería de funciones es la parte del código del sistema que permite la seguridad del mismo ya que cada módulo implementado manda llamar las funciones que necesite para dar la funcionalidad a lo requerido en el sistema.</p>

```

/*          Funcion de autocarga de clases          */

function __autoload($Class)
{
    $Class_file="Obj/Class_".$Class.".php";

    if(file_exists($Class_file)) require_once $Class_file;
    else die(sprintf("Archivo no encontrado %s",$Class_file));

}

/*          funciones */

function ValidacionUSR($usr,$pw)
{
    if(empty($usr)or empty($pw))
    {
        $val=false;
    }
    else
    {
        $q = new ConsultaBD("personal_dict");

        $SQL = sprintf("SELECT TAcSGP FROM
admusr WHERE NomUsr = '%s' AND Password =
PASSWORD('%s')",$usr,$pw);

        $q->consulta($SQL);

        if($q->RenAfec > 0)
        {
            $val=true;
            $_SESSION["TipoUSR"]=$q-
>Array_Reg[0]->TAcSGP;
        }
        else
        {
            $val=false;
        }
    }

    return $val;
}

function ValidarInfoUSR($Datos)
{
    $valusr['val'] = false;

    $Con = new ConsultaBD("personal_dict");

```

```

        $SQL = sprintf("SELECT `NomUsr` FROM
`admusr` WHERE `NomUsr` = '%s'", $Datos[1]);

        $Con->consulta($SQL);

        $Con->__destroy();

        if($Con->NumReg > 0)
        {

                printf("</td></tr><tr><td><label
class='ERROR' >El nombre de usuario %s ya existe seleccione
otro.</label>", $Datos[1]);
                $valusr[0] = "class='incorrecto'";
                $valusr['val'] = true;
                }
            else if(strlen($Datos[1]) < 8)
            {
                printf("</td></tr><tr><td><label
class='ERROR' >El nombre de usuario (%s) no tiene una longitud de
8 caracteres.</label>", $Datos[1]);
                $valusr[0] = "class='incorrecto'";
                $valusr['val'] = true;
                }

            if($Datos[2] != $Datos[3])
            {
                printf("</td></tr><tr><td><label
class='ERROR' >La Contraseña no coincide con la de
confirmación.</label>");
                $valusr[1] = "class='incorrecto'";
                $valusr['val'] = true;
                }
            else if(strlen($Datos[2]) < 8)
            {
                printf("</td></tr><tr><td><label
class='ERROR' >La Contraseña no tiene una longitud de 8
caracteres.</label>");
                $valusr[1] = "class='incorrecto'";
                $valusr['val'] = true;
                }

            if($Datos[4] == "N")
            {
                printf("</td></tr><tr><td><label
class='ERROR' >No a seleccionado un tipo de usuario.</label>");
                $valusr[2] = "class='incorrecto'";
                $valusr['val'] = true;
                }

            return $valusr;
        }
    }

```

```

function MensajeAlerta($mensaje)
{
    echo "<script language=javascript>";
    printf("window.alert('%s');",$mensaje);
    echo "</script> ";
}

function insertar_Acc_Sist($Datos)
{
    $ObjIns = new InsertBD("personal_dict");
    $Valores = sprintf("NULL , %d, '%s', PASSWORD( '%s' ) ,
's'", $Datos[0], $Datos[1], $Datos[2], $Datos[3]);
    $ObjIns->insertar("admusr", $Valores);
    $ObjIns->__destroy();
}

function validacionPrivilegios($Ide)
{
    $validacion = false;

    if($_SESSION['TipoUSR'] == "A") $validacion = true;
    else
    {
        $InfPer = new ConsultaBD("personal_dict");

        $SQL = sprintf("SELECT `Tipoc`
                                FROM
`inf_laboral`,`departamento`
                                WHERE
`inf_laboral`.`Id_Depto` = `departamento`.`Id_Depto`
                                AND
Id_persona = %d ", $Ide);

        $InfPer->Consulta($SQL);

        if($_SESSION['TipoUSR'] == $InfPer-
>Array_Reg[0]->Tipoc) $validacion = true;

        $InfPer->__destroy();
    }

    return $validacion;
}
?>

```

Tabla 3.3 Código del módulo de seguridad para el acceso al sistema

<b>Código</b>	<b>Descripción</b>
<pre> &lt;? class AccesoUSR extends QueryBD {     public \$TipoUsr;      private \$SesionActiva = true;     //private \$NomUsr;      function __construct(\$Enlace)     {         parent::__construct(\$Enlace);         \$this-&gt;SesionActiva=\$_SESSION['objeto'];         \$this-&gt;TipoUsr="comun";     }      function Login()     {         if(\$this-&gt;SesionActiva)         {             \$this-&gt;SesionActiva = false;             header('WWW-Authenticate: Basic realm="Acceso restringido");             header('HTTP/1.0 401 Unauthorized');             echo '- Authorization Required -';             exit;         }         else if(ValidacionUSR())         {             echo "Ha introducido el nombre de usuario: \$PHP_AUTH_USER&lt;br&gt;";             echo "Ha introducido la contraseña: \$PHP_AUTH_PW&lt;br&gt;";         }         else         {             echo "Nombre de usuario o contraseña incorrectos, porfavor intente de nuevo!!";         }     }      private function ValidacionUSR() </pre>	<p>Este módulo permite identificar el tipo de usuario registrado, para con ello poder proporcionar el tipo de información permitida dependiendo los privilegios con los que cuente dicho usuario.</p>

```

        {
            $this->Query("SELECT TAcSGP FROM admusr
WHERE NomUsr = ".$PHP_AUTH_USER." AND Password =
PASSWORD(".$PHP_AUTH_PW."");

            if($this->RenAfec > 0)
                {
                    $val = true;
                    $tipo = mysql_fetch_array($this->Result);
                    $this->TipoUsr = $tipo[0];
                }
            else $val = false;

            return $val;
        }

function __destroy()
    {
        parent::__construct($Enlace);
        $_SESSION['objeto']=$this->SesionActiva;
        $this->TipoUsr="comun";
    }

    }

?>

```

### 3.3.1 TIPOS DE USUARIO

El Sistema está diseñado para contemplar tres tipos de usuarios: administrador, propietarios y cualquiera. Cada uno tendrá distintos privilegios para poder manipular el sistema:

- Administrador: El Administrador tiene a su disposición la funcionalidad total del sistema, con la capacidad de consultar la información completa de cualquier persona de la base así como modificar o eliminar al personal, también tiene la facultad de administrar a los usuarios que tengan acceso al sistema.

- Propietario: El propietario puede consultar la información completa de cualquier persona pero sólo puede agregar, modificar y eliminar al personal que esté en su departamento, para esto existen 4 tipos de propietarios:
  - P – Petrolera
  - G – Geología
  - F – Geofísica
  - M – Minas
- Cualquiera: Un usuario cualquiera puede acceder al sistema sin poder realizar ningún tipo de modificación al mismo; no puede agregar o eliminar personal, solamente puede observar cierta información de los trabajadores.

El no poder agregar personal de otro departamento que no sea el suyo, se debe a que cada propietario sólo tiene la posibilidad de manejar a su propio personal y no el de los demás, esto se manipula de esta manera por seguridad de cada departamento.

### 3.3.2 TIPOS DE CONSULTA

La consulta en el Sistema se divide en dos tipos, esto debido a que no toda la información puede ser vista por todos los usuarios que ingresen al sistema, los tipos de consulta son: consulta pública y consulta completa.

#### **Consulta pública**

En la figura 3.1 la sección marcada hace mención a la consulta, ésta consiste en buscar a personas que trabajan en la DICT (División de Ingeniería en Ciencias de la Tierra) y que se encuentran registradas en el sistema. Este apartado se puede realizar de tres formas diferentes: por nombre, por número de trabajador y de una forma avanzada.

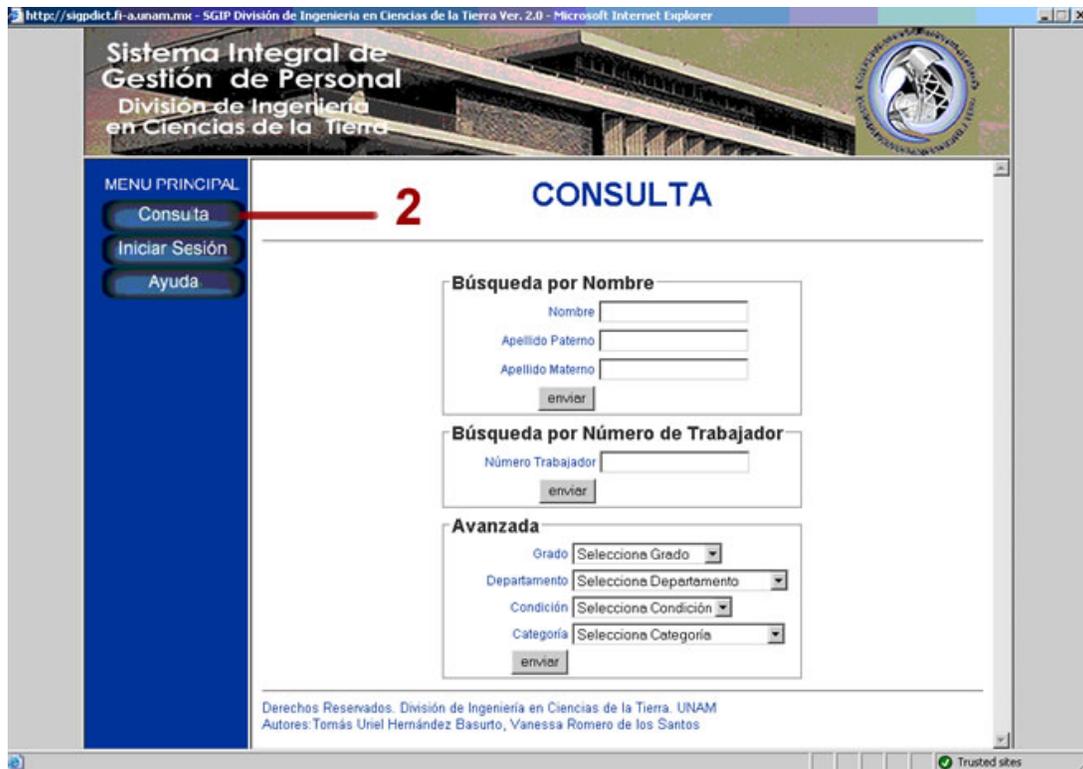


Fig. 3.1 Consulta pública

- **Por Nombre:** La consulta se puede realizar por nombre, por apellido paterno o por apellido materno de la persona que se desea buscar en el sistema, la búsqueda se realiza escribiendo el nombre o apellidos de la persona deseada o simplemente introduciendo la letra inicial de cualquiera de los campos mencionados.
- **Búsqueda por Número de Trabajador:** Sólo se necesita escribir el número de trabajador de quien se desea buscar.
- **Avanzada:** Seleccionar una o más opciones del menú en cualquiera de estas categorías: Grado, Departamento, Condición o Categoría, con esto se desplegará la información de la(s) persona(s) que se encuentran en los campos seleccionados.

Se realiza la consulta al ingresar cualquiera de los datos anteriores, y se mostrará por resultado el desglose de los datos más significativos de la persona a quien se estaba buscando.

### Consulta completa

Primero se requiere iniciar una sesión (figura 3.2). Una vez iniciada, en la sección de lado izquierdo (figura 3.3) en el icono "Consulta completa" se puede consultar de una forma más detallada los datos de las personas que se desea buscar, y posteriormente se realizan los mismos pasos que en la consulta pública (figura 3.1): ingresar nombre, apellido paterno o ingresar algún otro campo conocido de la persona.



Fig. 3.2 Inicio de sesión

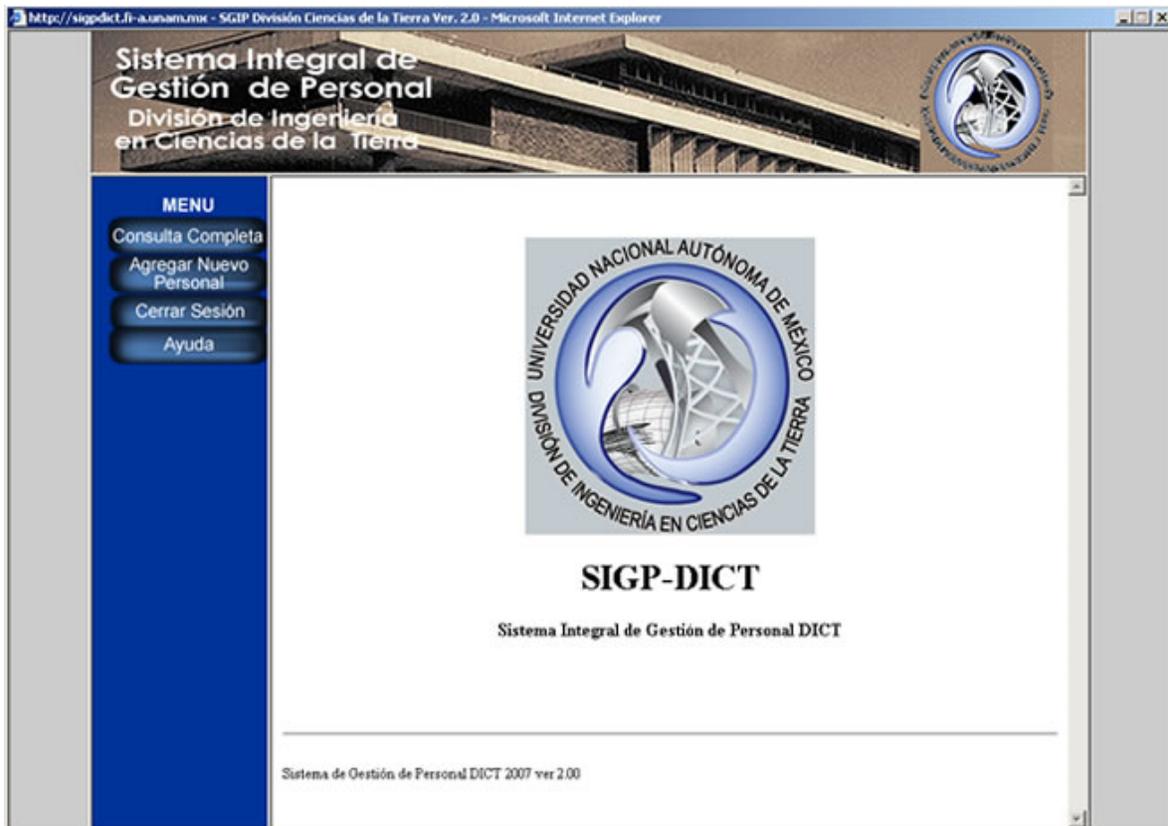


Fig. 3.3 Interfaz principal del sistema.

Al realizar la consulta, el sistema desplegará toda la información simple de la persona requerida.

### 3.3.3 ALTA Y BAJA DEL PERSONAL

La figura 3.4 muestra las opciones que tiene el sistema para dar de alta y/o baja la información del personal registrado en el Sistema. Es importante señalar, como ya se escribió anteriormente, que ésta acción sólo la pueden realizar los propietarios a su respectivo personal o el administrador de la Base de Datos bajo las condiciones expuestas.

## CONSULTA

[Imprimir](#)

Selección: Activos / Nombre con t / 0 a 30 de 1

SEL	NOMBRE	DEPARTAMENTO	UBICACIÓN	CATEGORÍA Y GRADO	TELÉFONO	CORREO
<input checked="" type="radio"/>	Correa Fernández Tania	Explotación del Petróleo	Principal Ala A PB, 56	Investigador Asociado Maestría	2651 245 Ext 123	tania@hotmail.com

[Imprimir](#)

Derechos Reservados. División de Ingeniería en Ciencias de la Tierra. UNAM  
Autores: Tomás Uriel Hernández Basurto, Vanessa Romero de los Santos

Fig. 3.4 Opciones para dar de alta y/o baja

### Alta

Para dar de alta a una persona se tiene que seleccionar a la persona que se quiere dar de alta y después dar clic en “alta”, para comprobar dicha acción seleccione la opción “personal activo” y agregue algún campo (figura 3.5).

## CONSULTA

**Situación**

Personal activo
  Personal inactivo.
  Ambos

---

**Búsqueda por Nombre**

Nombre

Apellido Paterno

Apellido Materno

---

**Búsqueda por Número de Trabajador**

Número Trabajador

---

**Avanzada**

Grado

Departamento

Condición

Categoría

Fig. 3.5 Alta del personal

Para verificar la acción de alta basta con revisar en la parte superior de la página principal de consulta para ver que este seleccionada la parte de "Activos".

**CONSULTA**

---

Selección: Activos Nombre con Tania / [Imprimir](#)

0 a 30 de 1

SEL	NOMBRE	DEPARTAMENTO	UBICACIÓN	CATEGORÍA Y GRADO	TELÉFONO	CORREO
☺	Correa Fernández Tania	Explotación del Petróleo	Principal Ala A PB, 56	Investigador Asociado Maestría	2651245 Ext 123	tania@hotmail.com

[Imprimir](#)

---

**Fig. 3.6** Confirmación del alta del personal

## Baja

Para dar de baja a una persona sólo requiere dar clic en “baja” (figura 3.4), para verificar que esta acción se realizó con éxito podrá ir a la ventana inicial y ver las opciones en personal inactivo (figura 3.7). En la parte inferior del nombre se mostrará entre paréntesis el estado del trabajador: “(Baja)”, de esta manera comprobará que el trabajador fue dado de baja (figura 3.8).

## CONSULTA

**Situación**

Personal activo
  Personal inactivo.
  Ambos

---

**Búsqueda por Nombre**

Nombre

Apellido Paterno

Apellido Materno

---

**Búsqueda por Número de Trabajador**

Número Trabajador

---

**Avanzada**

Grado

Departamento

Condición

Categoría

Fig. 3.7 Selección del personal dado de baja

## CONSULTA

[Imprimir](#)

Selección: Inactivos / Nombre con Tania / 0 a 30 de 1

SEL	NOMBRE	DEPARTAMENTO	UBICACIÓN	CATEGORÍA Y GRADO	TELÉFONO	CORREO
<input checked="" type="checkbox"/>	Correa Fernández Tania (Baja)	Explotación del Petróleo	Principal Ala A PB, 56	Investigador Asociado Maestria	2651245 Ext 123	tania@hotmail.com

[Imprimir](#)

Fig. 3.8 Verificación de la baja del personal

### 3.3.4 INSERCIÓN Y MODIFICACIÓN DEL PERSONAL

#### Inserción

Para agregar o registrar una persona nueva en el sistema se requiere de llenar dos formularios (figuras 3.9 y 3.10) con información del trabajador; dicha información es personal, laboral y académica. Existen campos que son obligatorios al llenar los formularios, es decir, es información mínima que se requiere para poder registrar a una persona dentro del sistema. Existe información mínima requerida, la cual se explico en el capítulo anterior, la cual debe ser proporcionada para poder registrar a una persona

HOJA 1

**Información Personal**

Nombre(s)  RCF

Apellido Paterno  CURP

Apellido Materno  Fecha de Nacimiento día  Mes  Año

Sexo

Nacionalidad  Mexicana  Extranjero:

---

**Dirección particular**

Calle y Numero

Colonia  C.P.

Delegación/Municipio  Estado

Teléfono Particular  Celular\*

---

**Direcciones Electrónicas**

Pagina Web\*

Correo electrónico 1

Correo electrónico 2\*

---

**Información Adicional**

Grado

Categoría

Categoría de Cómputo

Departamento

¿Se localiza dentro de la División?  Si  No

Fig. 3.9 Formulario de inserción, primer hoja

**Información laboral**

**HOJA 2**

No. Trabajador \*\*\*  Cedula Prof.

Antigüedad

Nivel

Condición

Fecha de ingreso  día  Mes  Año

Ultimo Periodo de contratación  día  Mes  Año  al  día  Mes  Año

No. Estacionamiento \*

Fotografía:  Examinar...

**Nota:** El formato de la Fotografía puede ser jpg, bmp, gif y png, no deberá exceder de los 160KB.

Curriculum:  Examinar...

**Nota:** El formato del archivo tiene que ser en PDF y no exceder los 952KB.

---

**Información Académica**

Folio\*  Cod. Prog\*

Horas

Área de conocimiento en Licenciatura \*\*:

Área de conocimiento en Posgrado \*\*:

Asignaturas:

---

**Ubicación dentro de la DICT**

Edificio

Cubículo

Teléfono

Fax \*

Ext

Fig. 3.10 Formulario de inserción, segunda hoja

## Modificación

Esta opción permite ir actualizando la información que se tenga del personal de la DICT como se muestra en la figura 3.11. Se debe tener en cuenta que al igual que la inserción, la modificación también presentara las dos páginas de información para su modificación (figura 3.12).

AdminROOT

---

[Imprimir](#)

Selección: Activos / Maestria /

0 a 30 de 2

SEL	Nombre (s)	Apellido paterno	Apellido materno	Departamento	Ubicación	Categoría	Grado	Teléfono	Correo
<input checked="" type="radio"/>	Thomas Eduardo	Perez	Garcia	Explotación del Petróleo	EdificioPrincipal Ala C 3er Piso, cubículo 113	Profesor Asignatura	Maestria	56223017	teperez@dictfi.unam.mx
<input type="radio"/>	Martin Carlos	Vidal	Garcia	Geología	EdificioPrincipal Ala A 400, cubículo 9	Técnico Académico Asociado	Maestria	56220851 Ext 109	martincarlosv@yahoo.com.mx

[Imprimir](#)

---

Derechos Reservados. División de Ingeniería en Ciencias de la Tierra. UNAM  
 Autores: Tomás Uriel Hernández Basurto, Vanessa Romero de los Santos

Fig. 3.11 Modificación del personal

http://132.248.54.197 - Actualizar - Microsoft Internet Explorer

**Modificaciones**

**Gonzalez Matrinez Julio Cesar**

**Información Personal**

Nombre(s)  RCF   
 Apellido Paterno  CURP   
 Apellido Materno  Fecha de Nacimiento     
 Sexo   
 Nacionalidad  Mexicana  Extranjero:

**Dirección particular**

Calle y Numero   
 Colonia  C.P.   
 Delegación/Municipio  Estado   
 Teléfono Particular  Celular

**Direcciones Electrónicas**

Pagina Web   
 Correo electrónico 1   
 Correo electrónico 2

**Información Adicional**

Grado   
 Categoría   
 Categoría de Cómputo   
 Departamento   
 ¿Se localiza dentro de la División?  Si  No

\* Estos campos son opcionales.  
 \*\* Esta información deberá ser llenada en caso de tener alguno.  
 \*\*\* Esta información es opcional únicamente para personal con condición externa o comisionados.

Done Trusted sites

Fig. 3.12 Formularios para las modificaciones

### 3.4 EXPECTATIVAS DEL SISTEMA

Los sistemas diseñados y desarrollados necesitan una serie de pruebas para asegurar su buen funcionamiento de acuerdo a las necesidades definidas en el inicio del desarrollo y la definición de proyecto.

En este caso se define el inicio de prueba de caja blanca esto es, prueba de ruta que ayuda a saber la complejidad lógica del sistema.

#### Prueba de Caja Blanca

Filosofía de diseño de casos de prueba que usa estructura de control como parte del diseño a nivel de componentes para derivar casos de prueba. Esta filosofía puede derivar casos de prueba que:

- 1- garanticen que todas las rutas independientes de un módulo se revisaron una vez.
- 2- revisen las decisiones lógicas en los casos verdadero y falso.
- 3- ejecuten los bucles en las fronteras.
- 4- revisen estructuras de datos internas para garantiza la validez.

La **prueba de ruta o trayectoria básica** es una técnica de prueba de caja blanca, esta prueba permite derivar una medida de complejidad lógica de un diseño de procedimientos y usar dicha medida como medida para definir un conjunto básico de rutas de ejecución.

#### ➤ Notación de gráfico grafo de flujo

El método de ruta básica, debe tener una notación simple para la representación del flujo de control al cual se le denomina gráfico de flujo.

Para el sistema presentado en la tesis se tiene la siguiente figura que representa el diagrama del grafo de flujo (figura 3.13)

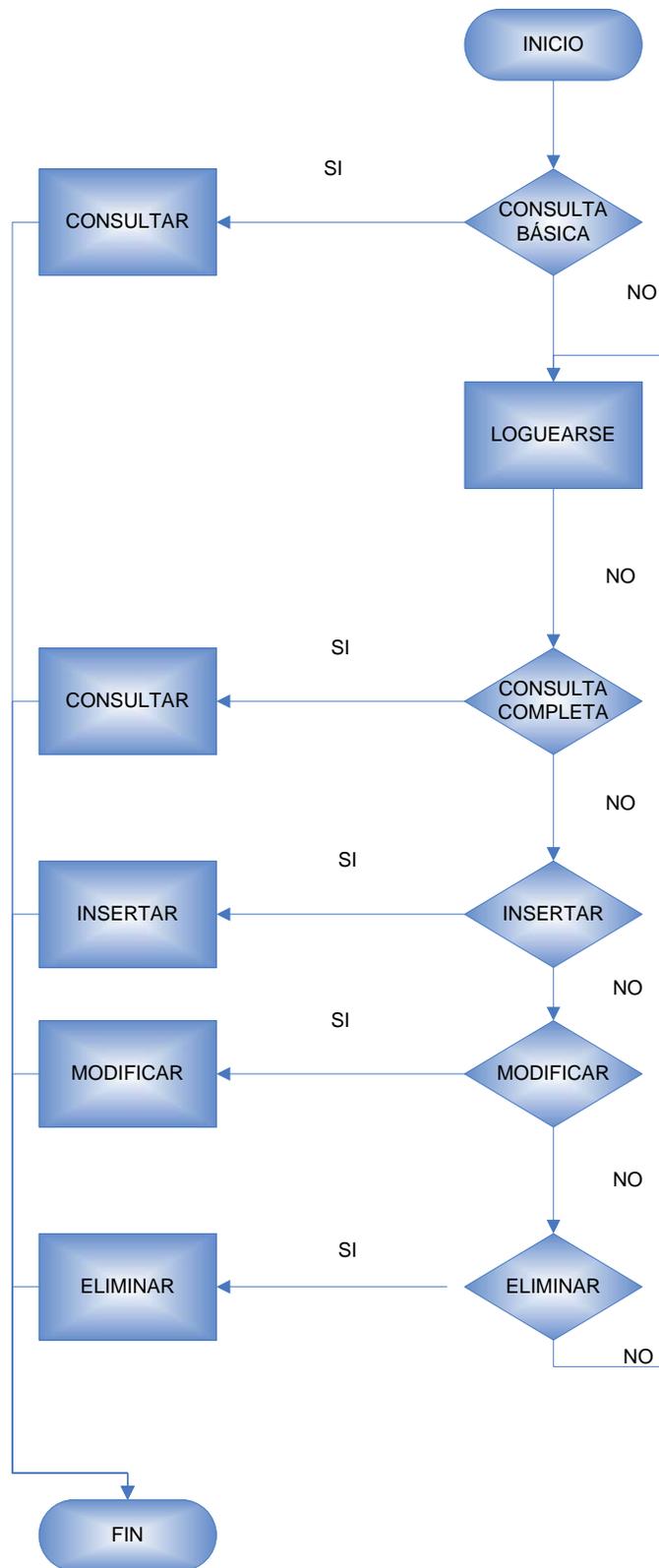
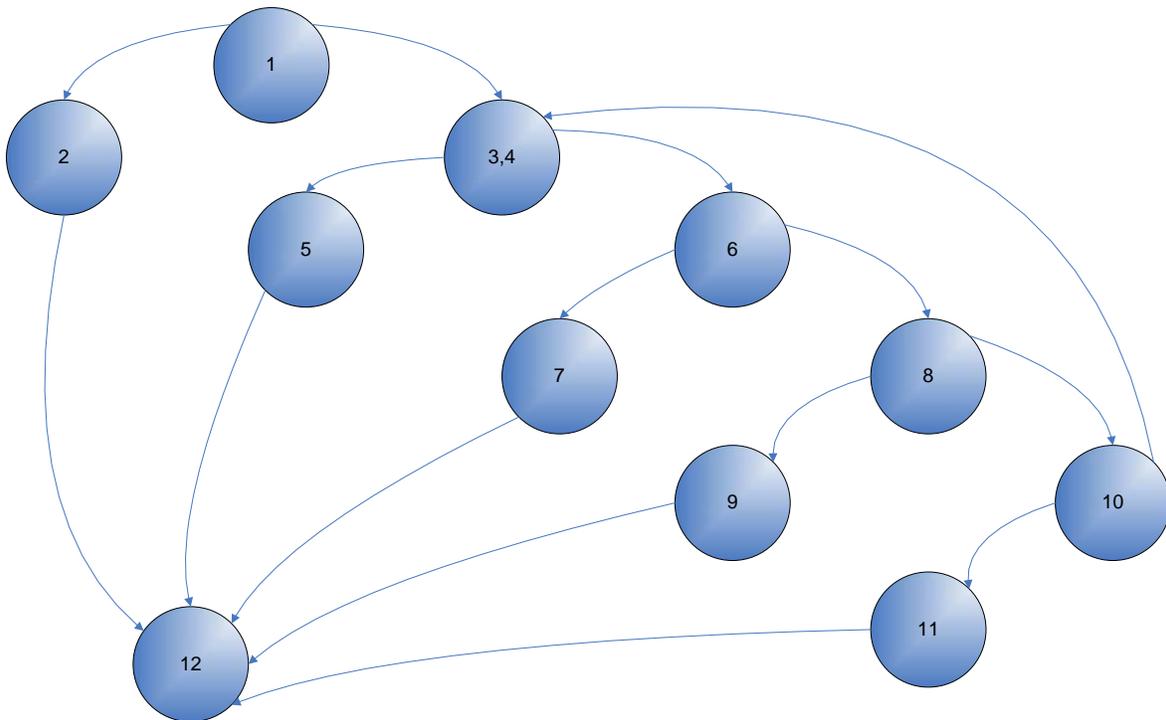


Fig. 3.13 Diagrama de flujo del sistema

➤ Rutas de programa independientes

Una ruta independiente es cualquiera que introduce al menos un nuevo conjunto de enunciados de procesamiento o una nueva condición de programa. Cuando se establece un gráfico de flujo, una ruta independiente debe moverse a lo largo de al menos una arista que no se haya recorrido antes al definir la ruta. (fig. 3.14)



**Fig. 3.14** Grafo de flujo

De acuerdo al grafo de flujo anterior se consideran las siguientes rutas.

Ruta 1: 1-2-12

Ruta 2: 1-3-4-5-12

Ruta 3: 1-3-4-6-7-12

Ruta 4: 1-3-4-6-8-9-12

Ruta 5: 1-3-4-6-8-10-11-12

Ruta 6: 1-3-4-6-8-10-3-4-5-12

La *complejidad ciclomática* es una medición de software que proporciona una evaluación cuantitativa de la complejidad lógica del programa. La complejidad ciclomática tiene fundamentos en la teoría de gráficos y proporciona una medición de software. La complejidad se calcula de la siguiente forma, para el sistema desarrollado.

1. El número de regiones del gráfico de flujo corresponde a la complejidad ciclomática.

$$V(G)=5$$

2. La complejidad ciclomática  $V(G)$  para un gráfico de flujo se define como:

$$V(G) = E - N + 2$$

donde E es el número de aristas del gráfico de flujo y N el número de nodos del gráfico de flujo,

$$E = 15$$

$$N = 11$$

sustituyendo

$$V(G) = 15 - 11 + 2$$

$$V(G) = 6$$

3. La complejidad ciclomática  $V(G)$  para un gráfico de flujo también se define como

$$V(G) = P + 1$$

donde P es el número de nodos predicado contenidos en el gráfico de flujo,

$$P = 5$$

sustituyendo

$$V(G) = 5 + 1$$

$$V(G) = 6$$

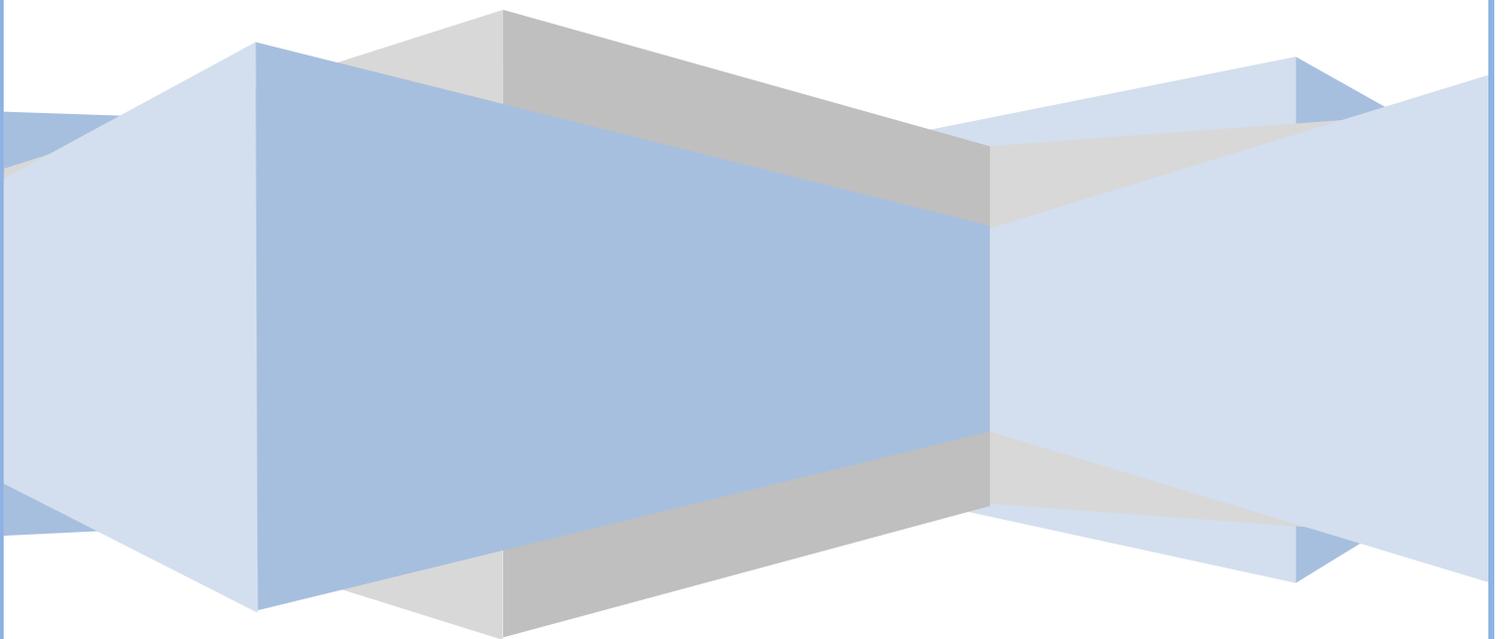
Por tanto lo anterior, la complejidad ciclomática del sistema presentado en este trabajo es de  $V(G) = 6$ .

Además de la prueba definida anteriormente son necesarias otras pruebas como la prueba de arquitectura cliente-servidor.

La **prueba de software cliente-servidor** ocurre en tres niveles diferentes 1) las aplicaciones cliente individuales se prueban de un modo “desconectado”; no se considera la operación del servidor ni la red subyacente. 2) El software cliente y las aplicaciones servidor asociadas se prueban en concierto, pero las operaciones de red no se revisan de manera explícita. 3) Se prueba la arquitectura cliente-servidor completa, incluidos la operación de red y el rendimiento.

# Capítulo 4

## Conclusiones



# CAPÍTULO 4



Al término de la presente se cubrieron con los objetivos planteados al inicio del desarrollo de la misma, esto es, tener un sistema confiable que cubriera con las necesidades de los usuarios, la manejabilidad y el acceso sencillo además de las seguridad de la información que maneja la Base de Datos.

La parte de la manejabilidad y acceso sencillo hacia los usuarios se resolvió por medio de la interfaz gráfica, dado que ésta permite una consulta sencilla de forma amigable hacia el usuario, basta con que el usuario ingrese el número de trabajador o el nombre del mismo para que el sistema le regrese la información de dicha búsqueda.

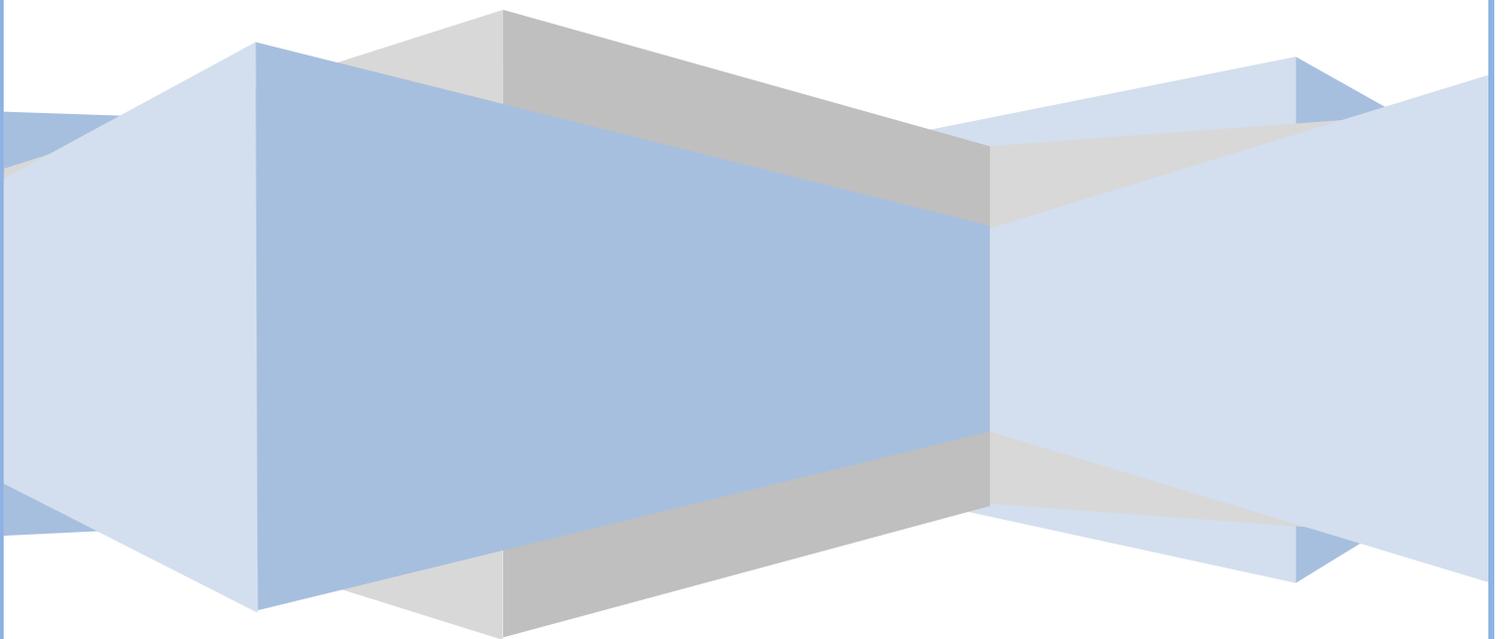
La seguridad se resolvió de dos formas, por un lado la seguridad de la información, para que dicha información no sea vista por las personas no debidas. Por otro lado la seguridad de la Base de Datos, teniendo en cuenta que las Bases de Datos hoy día pueden ser vulnerables, se resolvió restringiendo la modificación de la misma a usuarios con privilegios, sólo los usuarios con los privilegios necesarios pueden hacer las modificaciones o inserciones de nuevos datos. Los usuarios se dividieron de acuerdo a las áreas de la División para que de esta forma las áreas sólo puedan hacer las

modificaciones sobre la misma, aunque no con ellos significa que no puedan ver la información de otras áreas, eso si esta permitido.

Por otro lado la definición de las necesidades del sistema desde el inicio, fue importante para el buen desarrollo del mismo, esto sirvió para una mejor definición de la Base de Datos, así como la de las necesidades de los usuarios.

Con el desarrollo de este sistema se abre la posibilidad de crecimiento del mismo en un futuro dado que después de esto se puede hacer, por ejemplo, un manejador de estadísticas en base a lo ya establecido e ingresado en la Base de Datos. Sin embargo es importante mencionar que en la actualidad, por cuestiones administrativas de la División, el sistema NO se encuentra funcionando.

# Referencias y Bibliografía



# REFERENCIAS Y BIBLIOGRAFÍA



## LIBROS

---

1. Roger S. Pressman, *Ingeniería del Software*, McGraw-Hill, 7ª. Edición
2. Ford Merilee, *Tecnologías de inter conectividad de redes*, Prentice Hall, México, 1998
3. Laudon Kenneth C., *Administración de los sistemas de información*, Prentice Hall, 3ª. Edición
4. Whitten; Benthley y Barlow, *El desarrollo de sistemas de información*, McGraw-Hill, 3ª. Edición
5. F. Korth Henry, et.al., *Fundamentos de Bases de Datos*, McGraw-Hill, 3ª. Edición.
6. Powell Thomas A., *Diseño de sitios Web. Manual de referencia*, McGraw-Hill, Madrid 2001.
7. Laudon Kenneth C., *Administración de los Sistemas de información*, 3ª. Edición.
8. Ullman Jeffrey y Widom Jennifer, *Introducción a los Sistemas de Bases de Datos*, Editorial Prentice Hall, México 1999

## SITIOS WEB

---

9. <http://www.masadelante.com/faq-servidor.html>
10. <http://www.masadelante.com/faq-servidor-web.html>
11. [http://es.wikipedia.org/wiki/Servidor\\_web](http://es.wikipedia.org/wiki/Servidor_web)
12. [http://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache)
13. <http://www.algoritmia.net/articles.php?id=12>
14. [http://www.lafacu.com/apuntes/informatica/algorit\\_ordena/default.html](http://www.lafacu.com/apuntes/informatica/algorit_ordena/default.html)
15. [http://decsai.ugr.es/docencia/ii/ii\\_mod\\_av\\_bd.html](http://decsai.ugr.es/docencia/ii/ii_mod_av_bd.html)
16. [http://www.itlp.edu.mx/publica/tutoriales/admonbasedat/tema2\\_1.html](http://www.itlp.edu.mx/publica/tutoriales/admonbasedat/tema2_1.html)
17. <http://es.kioskea.net/contents/bdd/bddansi.php3>
18. <http://www.unalmed.edu.co/~mstabare/Dbms.html>
19. <http://ict.udlap.mx/people/carlos/is341/bases06.html>
20. <http://www3.uji.es/~mmarques/f47/apun/node83.html>
21. <http://www.masadelante.com/faq-internet.html>