



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

---

FACULTAD DE ESTUDIOS SUPERIORES  
ARAGON

"SISTEMA DE EVALUACIÓN DOCENTE VÍA WEB  
DE LA FACULTAD DE ECONOMÍA - SISTEMA  
UNIVERSIDAD ABIERTA"

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN

PRESENTA

MARCO ANTONIO ROSALES RODRÍGUEZ

ASESOR

ING. RODOLFO VÁZQUEZ MORALES

MÉXICO, 2006



Universidad Nacional  
Autónoma de México



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

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

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

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

---

## ***Agradecimientos***

Gracias a la *Universidad Nacional Autónoma de México* por haber forjado mi educación desde el nivel bachillerato hasta la licenciatura.

A mi jurado:

Ing. Blanca Estela Cruz Luévano. Gracias por sus acertadas observaciones, por su confianza y por el apoyo otorgado a mi tesis.

Ing. María Gabriela González Hernández. Gracias por el interés dedicado a mi tesis y por los conocimientos en el salón de clase.

Ing. Jeanett Figueroa Martínez. Gracias por los conocimientos impartidos en clase así como por el tiempo dedicado a mi tesis.

Ing. Rodolfo Vázquez Morales. Inge muchas gracias por la paciencia, apoyo, empeño y dedicación hacía mi tesis, por sus observaciones y comentarios. Además del conocimiento y experiencia impartidos en el salón de clase, los cuales han influido en mi desarrollo profesional. Muchas gracias.

M. en I. César Cárdenas Desales. Gracias por el interés y dedicación hacía mi tesis, además por tu amistad y compañerismo.

---

---

## ***Dedicatoria***

A mi mamá:

Gracias por el amor, cariño, amistad y comprensión. Sobretudo por haberme dado la vida y tu tiempo. Eres una persona a la que admiro por su fortaleza y valentía. Te agradezco infinitamente por tus cuidados y desvelos en los años de formación escolar y personal, que hasta la fecha siguen. Te quiero mucho ma.

A mi papá:

Gracias por tu dedicación, amor, protección y cariño. Así como también por tus enseñanzas, sacrificios, consejos y principios que en conjunto han hecho la persona que soy. Donde quiera que estés se que aún nos cuidas. Te quiero mucho pa.

A mi abuelita:

Gracias por tu amor, confianza y protección; así como por tus cuidados y enseñanzas. Ahora estoy seguro de que cuento con dos angelitos que me cuidan. Te quiero mucho abue.

A mis hermanos:

Mari, Bety y Chava; gracias por su amor, comprensión, amistad, por sus ejemplos y por su ayuda para alcanzar mis metas. No hubiera tenido mejores hermanos que ustedes y sabemos que podemos contar unos con otros, sobretudo en los momentos difíciles. Los quiero mucho.

A mi familia:

Mis cuñados Luis y Fer, mi sobrino Josué, tío Toño, tía Sonia, mis primos: Edgar, Christian, Nataly y Yael. Gracias por su apoyo, cariño y enseñanzas, los quiero mucho.

A Erika:

Años de conocernos, gracias por tu amistad, apoyo, cariño, confianza, por esas largas conversaciones y por todo, a ti y a tu familia. Te quiero mucho amiga.

A Xóchitl:

Gracias por tu amistad, cariño, comprensión, apoyo en los momentos difíciles y por tu confianza. Te quiero mucho amiga.

A Liz:

Gracias por tu amistad, cariño, apoyo, por compartir conmigo tu tiempo y todos lo momentos inolvidables, te quiero mucho amiga, salud!!!

A Marcos:

Gracias por tu amistad, apoyo y por todos los momentos que hemos pasado. Muchas gracias amigo.

A Fer:

Gracias por tu amistad y compañerismo demostrado durante toda la carrera y hasta la fecha. En esos días de exámenes, tareas, trabajos, por las dudas resueltas y por ser el gran sensei, muchas gracias amigo.

---

---

A Mario, César, Ivo, Eri, Sandra, Estela e Irma:

Gracias por el compañerismo durante la carrera, por esos inolvidables días sin dormir, sin comer, las platicas, las clases, por su ayuda y por todos los momentos que pasamos en Aragón.

A mis amigos:

Daniel R, Martín, Maricela, Juan e Ignacio, gracias por su amistad, por su cariño y por compartir conmigo inolvidables momentos. Muchas gracias amig@s.

Al SUAFE:

Mtro. Paz, Laura, Mariana V, Mariana D, Beto, MASH, Sra Catita y Mtra. Margarita, gracias por su apoyo, confianza y compañerismo.

Al LNC:

Dra. Selene, muchas gracias por su confianza y apoyo.

Paty, gracias amiga por tu confianza, amistad, apoyo, cariño, consejos y por las conversaciones, te quiero mucho.

Ale, eres una persona especial de la cual he aprendido a seguir mis objetivos, muchas gracias por tu amistad y apoyo, te quiero mucho amiga.

Lis, Mariana y Sandra, gracias por su confianza, apoyo, amistad y por las conversaciones, las quiero mucho amigas.

Mel, Haydée, Eve, Cin, Andrea y David, muchas gracias por su amistad, compañerismo y apoyo.

---

---

"Mi afán de estudio no es para saber más,  
sino para ignorar menos"  
Sor Juana Inés de la Cruz

"Je préfère être détesté pour ce que je suis,  
que être aimé pour ce que je ne suis pas"

---

---

# ÍNDICE

<b>Índice</b>	I
<b>Introducción</b>	III
<b>Capítulo 1. Sistemas Web</b>	1
1.1. Antecedentes	2
1.1.1. Internet	2
1.1.2. World Wide Web	5
1.2. Sistema web	6
1.3. Elementos de un sistema web	8
1.3.1. Servidor (Sistema operativo)	10
1.3.2. Aplicaciones	11
1.3.3. FTP (File Transfer Protocol)	11
1.3.4. Correo	11
1.3.5. Chat, Listas y Noticias	12
1.3.6. Web	12
1.4. Software libre	13
1.5. Aplicaciones web	15
<b>Capítulo 2. Metodología y herramientas de desarrollo</b>	16
2.1. UML (Lenguaje Unificado de Modelado)	18
2.1.1. Historia	19
2.1.2. Objetivos	20
2.1.3. Diagramas	20
2.1.3.1. Diagrama de casos de uso	22
2.1.3.1.1. Elementos	22
2.1.3.2. Diagrama de clases	23
2.1.3.2.1. Elementos	23
2.1.3.3. Diagrama de secuencia	26
2.1.3.3.1. Elementos	26
2.2. Bases de datos	27
2.2.1. Dato	27
2.2.2. Información	27
2.2.3. Campo	27
2.2.4. Registro	27
2.2.5. Archivo	27
2.2.6. Base de datos	27
2.2.7. Sistema Manejador de Base de Datos (DBMS)	28
2.2.8. Administrador de base de datos (DBA)	29
2.2.9. Objetivos de las bases de datos	29
2.2.10. Abstracción de la información	30
2.2.10.1. Nivel físico	30
2.2.10.2. Nivel conceptual	30
2.2.10.3. Nivel de visión	30
2.2.11. Modelos de datos	31
2.2.11.1. Modelos lógicos basados en objetos	31
2.2.11.1.1. Modelo Entidad – Relación	31
2.2.11.2. Modelos lógicos basados en registros	32
2.2.11.2.1. Modelo relacional	32
2.2.11.2.2. Modelo de red	33
2.2.11.2.3. Modelo jerárquico	33
2.2.11.3. Modelos físicos de datos	33
2.2.12. Modelo de datos relacional	34
2.2.12.1. Normalización de bases de datos	36
2.3. Red Hat Linux	36
2.3.1. Historia	37
2.3.2. Características	37
2.4. MySQL	38

2.4.1. Historia	38
2.4.2. Características	38
2.4.3. Instalación	39
2.4.4. Tipos de datos en MySQL	41
2.4.5. Sintaxis y comandos	41
2.5. Apache	47
2.5.1. Historia	47
2.5.2. Configuración	49
2.6. PHP	49
2.6.1. Historia	49
2.6.2. Sintaxis y estructura	51
2.6.3. Características	52
2.6.4. Instalación y configuración	54
2.6.5. Funciones para el manejo de bases de datos (MySQL)	55
2.7. HTML	57
2.8. JavaScript	58
2.9. Cookies	58
<b>Capítulo 3. Análisis, diseño y programación</b>	<b>60</b>
3.1. Análisis	61
3.1.1. Descripción del problema	61
3.1.2. Requerimientos generales del sistema de información	64
3.2. Diseño y programación	64
3.2.1. Descripción general del sistema	64
3.2.2. Diagrama de casos de uso	65
3.2.3. Diagrama de clases	69
3.2.4. Diagrama de secuencia	72
3.2.5. Base de datos	75
3.2.5.1. Diseño de la base de datos	75
3.2.5.2. Diagrama Entidad – Relación de la base de datos	76
3.2.5.3. Diccionario de datos	77
3.2.6. Especificación de los módulos del sistema	80
3.2.6.1. Módulo de usuarios	80
3.2.6.2. Módulo de administración	81
3.2.6.3. Módulo de reportes	81
3.2.7. Reglas de negocio	82
<b>Capítulo 4. Implementación</b>	<b>84</b>
4.1. Mapa de navegación	85
4.2. Mapa de archivos	86
4.3. Diseño de pantallas	88
4.4. Características del servidor	105
4.5. Plan de implementación	105
<b>Conclusiones</b>	<b>109</b>
<b>Bibliografía</b>	<b>112</b>
<b>Anexos</b>	<b>114</b>
Glosario	115
Bloques de código	117
Formatos	123



# **INTRODUCCIÓN**

La evolución de Internet ha permitido el desarrollo de importantes aplicaciones que son de gran utilidad para todos los sectores, ha acortado distancias y ha hecho que la información sea accesible sin importar el lugar. Uno de los servicios más utilizados en Internet es la Web, que por medio de un programa llamado navegador permite acceder a páginas de hipertexto o páginas web, las cuales contienen la información requerida.

En algunos casos la información no se encuentra dentro de la página web; sino que está almacenada en Sistemas de Información y sólo puede ser consultada y/o almacenada por ciertas personas y en lugares específicos. Debido a esto se desarrollaron programas que interactúan con las bases de datos (parte central de los Sistemas de Información) que son compatibles con el código de las páginas de hipertexto.

Cuando las bases de datos pueden ser explotadas vía web, se denominan Sistemas Web, sin dejar de ser un Sistema de Información. Los Sistemas de Información son parte fundamental en toda empresa u organización, ya sea lucrativa o no lucrativa; ya que los informes que emiten son útiles para tomar decisiones.

Dicho esquema de trabajo ofrece disponibilidad para consultar, almacenar, eliminar y/o modificar los datos almacenados en el banco de datos desde cualquier parte, evitando desplazarse al lugar donde se encuentra físicamente el servidor de base de datos; eficiencia y eficacia de los métodos de explotación de la base de datos y veracidad de la información obtenida. Así como facilidad para los usuarios de interactuar con el sistema; mediante interfaces amigables y sencillas.

Los Sistemas Web han sido utilizados por un gran número de instituciones, tanto del sector público como del sector privado, debido a las ventajas mencionadas anteriormente entre otras. Además; el software más utilizado para el desarrollo del Sistema Web es gratuito (software libre), por lo que el número de instituciones que han optado por la opción sigue creciendo en la actualidad, ya que el egreso económico es menor comparado con los beneficios obtenidos.

En la División Sistema Universidad Abierta de la Facultad de Economía, se solicita cada semestre a los alumnos que evalúen a la planta docente. La forma de recolectar los datos y de generar los reportes es muy complicada, ya que se lleva a cabo de forma manual por sólo dos personas, motivo por el cual se pueden perder datos importantes y se generen resultados poco confiables y el proceso consume demasiado tiempo. Motivo por el cual se planteó la creación de un Sistema de Información vía web, decisión que se tomó por las características mencionadas anteriormente, con lo que se automatizará el proceso de administración de la información y la generación de reportes en forma confiable y oportuna.

El proyecto planteado se desarrollará en cuatro capítulos.

Capítulo 1, Sistemas Web. Se presentarán los antecedentes de los Sistemas Web para obtener una definición concisa. Además, se describirán los elementos que conforman a un Sistema Web y se justificará el software a utilizar.

Capítulo 2, Metodología y herramientas de desarrollo. Se definirá la metodología a seguir y los conceptos relacionados con las bases de datos. También se describirán las características, la forma de instalación y los comandos a utilizar de las herramientas necesarias.

Capítulo 3, Análisis, diseño y programación. Se detallará el proceso de análisis del nuevo sistema, en base a la metodología planteada anteriormente. Teniendo los requerimientos identificados se procederá al diseño de la base de datos; además, se describirá el comportamiento del sistema y la especificación de los módulos y las reglas de validación.

Capítulo 4, Implementación. Se elaborarán las interfaces de usuario así como la ubicación de archivos y forma de navegación. Esto con el objetivo de tener un diagrama general de los archivos y facilitar la localización en el momento necesario. Además, se incluirá el plan de implementación.

En base a lo anterior se cumplirá el principal objetivo del proyecto de forma satisfactoria: desarrollar un Sistema de Información vía Web que automatice el proceso de administración de la información, generando reportes de forma oportuna y confiable.

# **CAPÍTULO 1**

## **SISTEMAS WEB**

El entorno de operación de Internet ha diversificado los usos y aplicaciones, pasando desde las primeras aplicaciones simples hasta las conexiones inalámbricas a las bases de datos. Este crecimiento se ha basado en muchas áreas de cómputo como redes, sistemas operativos, bases de datos, etc., pero sin lugar a duda, gran parte de éste tiene base en el software libre, el cual ha sido el complemento ideal para desarrollar Sistemas Web.

## 1.1. Antecedentes

### 1.1.1. Internet

La historia de Internet se remonta a principios de los años sesenta, cuando investigadores de instituciones de reconocido prestigio como el Instituto Tecnológico de Massachussets (MIT) sentaron las bases tecnológicas que facilitaron en años posteriores la creación de la red Internet. Leonard Kleinrock (MIT) fue el primero que habló sobre la teoría de conmutación por paquetes (PS) en su artículo "Flujo de información en redes amplias de comunicación". J.C.R. Licklider y W. Clark, también del MIT, escribieron "Comunicación hombre – computadora en línea" y Paul Baran, RAND publicó "Redes de comunicación distribuida", en el que hablaba de redes conmutadas por paquetes, sin punto único de interrupción.

En 1965, la U.S. DARPA (*Defense Advanced Research Projects Agency*) promueve un estudio sobre "Redes cooperativas de computadoras de tiempo compartido" y al año siguiente Larry Roberts, del MIT, publicó "Hacia una red cooperativa de computadoras de tiempo compartido". En los años sucesivos se presentan proyectos sobre redes conmutadas por paquetes, como el Symposium sobre Principios Operativos de 1967. El objetivo principal era la investigación y desarrollo de protocolos de comunicación para redes de área amplia, para ligar redes de transmisión de paquetes de diferentes tipos, capaces de resistir condiciones de operación más difíciles y continuar funcionando aún con la pérdida de la parte de una red, por ejemplo, en caso de guerra.

En efecto, a finales de los años sesenta, una de las preocupaciones de las Fuerzas Armadas de los Estados Unidos era conseguir una manera de que las comunicaciones estuvieran descentralizadas, es decir, evitar un centro neurálgico de comunicaciones que pudiera ser destruido durante un eventual ataque militar con armas nucleares y que así, aún sufriendo el ataque, las comunicaciones no se bloquearan, sino que solamente se perdiera un nodo.

Estas investigaciones dieron como resultado el protocolo TCP/IP (*Transmission Control Protocol / Internet Protocol*), un sistema de comunicaciones muy sólido y robusto bajo el cual se integran todas las redes que conforman lo que se conoce actualmente como Internet, por lo tanto a DARPA se le conoce como la madre de Internet.

En 1969, la DARPA, junto con la compañía Rand Corporation, desarrollaron una red sin nodos centrales basada en conmutación de paquetes. La información se dividía en paquetes y cada uno contenía la dirección de origen, la de destino, un número de secuencia y cierta información. Los paquetes al llegar al destino se ordenaban según el número de secuencia y se juntaban para dar lugar a la información. Al viajar por la red paquetes, era más difícil perder datos ya que, si un paquete concreto no llegaba al destino o arribaba defectuoso, la computadora que debía recibir la información sólo tenía que solicitar a la computadora emisora el paquete que le faltaba. El protocolo de comunicaciones se llamó NCP (*Network Control Protocol*). Esta red en principio sólo unía a un pequeño número de computadoras y se denominó DARPA, pero en 1972 se cambió el nombre por ARPANET, cuando ya conectaba a unos cuarenta nodos. En 1971 se creó el primer programa para enviar correo electrónico. Fue Ray Tomlinson del BBN, y combinaba un programa interno de correo electrónico y un programa de transferencia de archivos. También en este año un grupo de investigadores del MIT presentaron la propuesta del primer "Protocolo para la transmisión de archivos en Internet" (RFC 114). Era un protocolo muy sencillo basado en el sistema de correo electrónico, pero que sentó las bases para el futuro protocolo de transmisión de archivos (*FTP, File Transfer Protocol*).

Durante el desarrollo del protocolo se incrementó notablemente el número de redes locales de agencias gubernamentales. Fue en este momento cuando las instituciones académicas se interesaron por estas posibilidades de conexión. La NSF (*National Science Foundation*) dio acceso a sus seis centros de supercómputo y a cuatro universidades de los Estados Unidos, las cuales lograron enlazarse entre sí a través de la ARPANET, dando origen a la red de redes más grande del mundo. A partir de aquí se fueron conectando otras redes y se evitó la existencia de centros para preservar la flexibilidad y la escalabilidad.

Se pensó que la red debía ser lo más sencilla posible para facilitar las implementaciones. Así, los cambios de tecnología afectarían a los extremos de la red, las computadoras, pero no al tejido que las unía. La red únicamente debía encargarse de entregar bien los paquetes que eran enviados a través de ella y las tareas más complejas deberían hacerse en los extremos. El empeño de DARPA en lograr implementaciones a bajo costo contribuyó decisivamente a esta sencillez.

Los años setenta transcurren con instituciones conectándose directamente o conectando otras redes a ARPANET y con los responsables desarrollando estándares y protocolos, como Telnet, la especificación de transferencia de archivos o el protocolo de voz en redes (*NVP, Network Voice Protocol*). Vinton Cerf y Bob Kahn publican "Protocolo para intercomunicación de redes por paquetes", el cual especifica con detalle el diseño del Programa de Control de Transmisión (*TCP, Transmission Control Protocol*).

En 1979, ARPA crea la primera comisión de control de la configuración de Internet y tras varios años de trabajo, por fin en 1981 se termina de definir el protocolo TCP/IP y ARPANET lo adopta como estándar en 1982, sustituyendo a NCP. Estas son las primeras referencias a Internet, como "una serie de redes conectadas entre sí, específicamente aquellas que utilizan el protocolo TCP/IP". Internet es la abreviatura de *Interconnected Networks*, es decir, Redes interconectadas o red de redes. Además, en estos años se fundan Microsoft (1975) y Apple (1976).

A partir de la década de los 80 se forman dos nuevos bloques. En 1983 ARPANET se separa de MINET, la red militar con información no clasificada que la originó, por lo que las funciones militares se separaron permitiendo así que todo aquel que lo requiriera, sin importar el país que lo solicitara, siempre y cuando fuera para fines académicos y de investigación, pudiera tener acceso a la red (claro está, pagando sus gastos de conexión) facilitando de esta manera una comunicación continua. De modo que ya sin fines militares se puede considerar esta fecha como el nacimiento de Internet. Es el momento en que el primer nodo militar se desliga, dejando abierto el paso para todas las empresas, universidades y demás instituciones que ya por esa época poblaban la joven red. En este año sale la primera versión del Windows de Microsoft. En esta década de los ochenta, la expansión es enorme. Otras redes experimentales que utilizaban paquetes de radio y satélites, se conectaron a ARPANET, utilizando la tecnología interconectada por DARPA. En un principio, esta interconexión de redes experimentales y de producción se denominó INTERNET DARPA y posteriormente se le quedó el nombre de INTERNET.

En 1985, quince años después de la primera propuesta, se termina el desarrollo del aún vigente protocolo para la transmisión de archivos en Internet (*FTP*), basado en la filosofía de cliente-servidor. Un punto fundamental en su éxito fue el hecho de que ARPA distribuyó a bajo costo los protocolos, que fueron adoptados por el UNIX de BSD (*Berkeley Software Distribution*), cuyo uso fue muy difundido entre las universidades. De esta forma se crearon una gran cantidad de servicios y se provocó un importante avance en el desarrollo de la red. Por esta época se crea el sistema de nombres de dominios (*DNS, Domain Name System*).

A partir de 1987 empezó la gran expansión; que en parte se debió a que un año antes se creó la NSFNET (*National Science Foundation's Network*), lo cual estableció cinco centros de supercomputadoras para proveer un alto poder de proceso; es ahora cuando se incorporan a Internet diversas redes de Europa. También en ese año se generó la primera aplicación informática de hipertexto. Fue HyperCard para Macintosh, y estaba pensada para crear y compartir pilas de

información; dentro de cada pila podía haber vínculos de hipertexto de un elemento de información a otro.

En los Estados Unidos de América el gran aumento de usuarios provocó en 1990 la retirada de la agencia ARPA y su red pasó a estar a cargo de la NSF. Internet comenzó a saturarse y para evitar el colapso, se restringieron los accesos. Eran años de incertidumbre ya que nadie había ideado la red para los fines y las dimensiones que se estaban alcanzando; y los responsables se veían desbordados. Durante estos años de incertidumbre se reforzaron las redes dorsales y se ideó el World Wide Web en el CERN (*Centre Européen pour la Recherche Nucléaire*), gracias a Tim Berners-Lee, su inventor; que creó las bases del protocolo de transmisión HTTP (*HyperText Transfer Protocol*), el lenguaje de documentos HTML (*HyperText Markup Language*) y el concepto de los URL (*Uniform Resource Locator*).

Al contrario de lo que comúnmente se piensa, Internet no es sinónimo de World Wide Web. Ésta última es parte de la primera, siendo la World Wide Web uno de los muchos servicios proporcionados en la red Internet. La Web es un Sistema de Información más reciente que emplea la red Internet como medio de transmisión.

El grupo con mayor autoridad sobre el desarrollo de la red es la *Internet Society*, creada en 1992 y formada con miembros voluntarios, su propósito principal es promover el intercambio de información global a través de la tecnología de Internet; por lo tanto este grupo tiene la responsabilidad de la administración técnica y dirección de Internet aunque no es el único; existen además otros tres grupos:

- ❖ *Internet Architecture Board*. Toma las decisiones sobre los estándares de comunicación entre las diferentes plataformas, para que puedan interactuar máquinas de distintos fabricantes sin problemas, y a la vez, siendo responsable de asignar las direcciones y otros recursos.
- ❖ *Network Information Center (NIC)*. El grupo es administrado por el Departamento de Defensa de E.U.A. y se encarga de autorizar las asignaciones.
- ❖ *Internet Engineering Task Force (IETF)*. Los usuarios de Internet expresan sus opiniones sobre la forma en que se deben de implementar soluciones para problemas operacionales y cómo deben cooperar las redes para lograrlo.

En 1993 se creó Mosaic, el primer navegador y la World Wide Web comenzó a despuntar. En septiembre de 1993 se inició el primer servidor Web en español. En esos momentos se incrementó la potencia de las redes troncales de E.U.A. y en 1994 se eliminan las restricciones de uso comercial de la red, y el gobierno de este país deja de controlar la información de Internet. Para entonces, nace una empresa: *Netscape*, y con ella un nuevo navegador, Navigator.

1995 es el año del gran “boom” de Internet. Puede ser considerado como el nacimiento de Internet comercial y desde ese momento el crecimiento de la red ha superado todas las expectativas. Este hecho se produce porque es en este año cuando la WWW supera a FTP transformándose en el servicio más popular de la red, después de que el año anterior superase a Telnet. A partir de aquí empiezan a incrementarse de una manera casi exponencial el número de servicios que operan en la red, por lo que la escalada de tecnología es impresionante. Se desarrollan los motores de búsqueda que rápidamente añaden búsquedas inteligentes en varios idiomas. El lenguaje Java empieza a pegar fuerte y se desarrollan tecnologías como entornos virtuales (*VRML*) o el teléfono por Internet, que permite la conexión con todo el mundo a precio de llamada local. Se desarrolla de una manera definitiva el comercio electrónico, para comprar productos y servicios a través de Internet. Se pueden ver cientos de televisiones y escuchar radios de todo el mundo en tiempo real. Los bancos se asientan en la red y la gente empieza a ceder en su miedo inicial, confiando en la confiabilidad que otorgan los servidores seguros. Aparecieron los primeros virus de HTML; son virus de macro incrustados en

documentos de Word que se transmiten por correo electrónico como archivos adjuntos y se ejecutan en las máquinas sin protección contra ellos.

La tecnología de telefonía móvil y la de Internet finalmente se unen para poder acceder desde los teléfonos móviles a la red de redes. Si bien es cierto que los dispositivos inalámbricos ya accedían a la red, es con la definición del conjunto de protocolos WAP (*Wireless Application Protocol*) cuando los dispositivos inalámbricos, y fundamentalmente los teléfonos móviles, se conectan a Internet. WAP ha tenido un importante respaldo por parte de fabricantes de teléfonos, operadoras, compañías de software y desarrolladores, lo que ha provocado que en muy poco tiempo se convirtiera en estándar. Surgió entonces el WAP Forum, que hoy agrupa el 90% de los fabricantes de teléfonos móviles y cubre unos 100 millones de estos dispositivos en todo el mundo.

### 1.1.2. World Wide Web

La *World Wide Web* (del inglés, Telaraña Mundial), la Web o *www*, es un sistema de hipertexto que funciona en Internet. Para ver la información, se utiliza una aplicación llamada navegador web que extrae elementos de información (llamados “documentos” o “páginas web”) de los servidores web (o “sitios”) y los muestra en la pantalla del usuario.

Una página de Internet o página Web es un documento electrónico que contiene información específica de un tema en particular y que es almacenado en algún sistema de cómputo que se encuentre conectado a la red mundial de información denominada Internet, de tal forma que este documento pueda ser consultado por cualquier persona que se conecte a esta red mundial de comunicaciones y que cuente con los permisos apropiados para hacerlo.

Una página Web tiene la característica peculiar de que el texto se combina con imágenes para hacer que el documento sea dinámico y permita que se puedan ejecutar diferentes acciones, una tras otra, a través de la selección de hiperenlaces o de las imágenes.

El usuario puede entonces seguir hiperenlaces que hay en la página a otros documentos o incluso enviar información al servidor para interactuar con él. A la acción de seguir hiperenlaces se le suele llamar “navegar” por la Web. No se debe confundir la Web con Internet, que es la red física mundial sobre la que circula la información.

La Web nació alrededor de 1989 a partir de un proyecto del CERN, en el que Tim Berners-Lee construyó el prototipo que dio lugar al núcleo de lo que hoy es la World Wide Web. La intención original era hacer más fácil el compartir textos de investigación entre científicos y permitir al lector revisar las referencias de un artículo mientras lo fuera leyendo. Un sistema de hipertexto enlazaría todos los documentos entre sí para que el lector pudiera revisar las referencias de un artículo mientras lo fuera leyendo. El nombre original del prototipo era “*Enquire Within Upon Everything*”.

La funcionalidad elemental de la Web se basa en tres estándares: El Localizador Uniforme de Recursos (*URL*), que especifica cómo a cada página de información se asocia una “dirección” única en donde encontrarla; el Protocolo de Transferencia de Hipertexto (*HTTP*), que especifica cómo el navegador y el servidor intercambian información en forma de peticiones y respuestas, y el Lenguaje de Marcación de Hipertexto (*HTML*), un método para codificar la información de los documentos y sus enlaces. Berners-Lee dirige en la actualidad el World Wide Web Consortium, que desarrolla y mantiene estos y otros estándares que permiten a las computadoras de la Web almacenar y comunicar todo tipo de información.

El programa inicial del CERN, “*www*”, sólo presentaba texto, pero navegadores Web posteriores, como “Viola” de Pei Wei (1992) añadieron la capacidad de presentar también gráficos. Marc Andreessen de NCSA presentó un navegador Web llamado “Mosaic para X” en 1993 que disparó la popularidad de la Web entre principiantes. Andreessen fundó Mosaic Communications Corporation (hoy Netscape Communications), añadiendo características adicionales como contenido dinámico, música y animación que están incluidas en los modernos navegadores. A menudo la capacidad de los



navegadores y servidores avanza más rápido que los estándares, con lo cual es habitual que las características nuevas no funcionen en todas las máquinas, impidiendo la accesibilidad universal.

El imparable avance técnico de la WWW permite hoy incluso servicios en tiempo real como webcasts, radio web y webcams en directo. Una de las tecnologías en constante evolución en el mundo de la WWW es Flash, un formato registrado por la compañía Macromedia que aporta un gran dinamismo a la Web. El lenguaje de scripting que usan, Actionscript, goza de un gran potencial que abarca desde la aplicación visual hasta la interactividad con el servidor. Otro avance importante fue la plataforma Java de Sun Microsystems, que permitió a las páginas Web incluir pequeños programas (llamados applets) que se ejecutan en la máquina del cliente y mejoran tanto la presentación como la interactividad.

La Web ha crecido hasta arrinconar otras formas de comunicación anteriores, como los tableros electrónicos, los servicios de noticias (*News*), buscadores de documentos (*Archie*) y ha mantenido cierta convivencia con el protocolo de transferencia de archivos (*FTP*) y los protocolos de correo electrónico: SMTP, POP3 e IMAP.

La W3C es el organismo que regula los estándares Web, necesario para asegurar así el acceso universal a la información. Aún así, hay quienes los ignoran, dando lugar a páginas web que solo se ven bien con uno o unos pocos navegadores concretos.

Las posibilidades que ofrece Internet se denominan servicios. Cada servicio es una manera de sacarle provecho a la Red. Uno de los servicios más utilizados, es la *World Wide Web* antes mencionada, además de los siguientes:

- ❖ FTP (File Transfer Protocol). Permite la transferencia de archivos por Internet.
- ❖ Correo electrónico. Permite enviar y recibir mensajes escritos a otras personas que tengan acceso a la Red.
- ❖ Telnet. Permite realizar sesiones remotas, es decir, se puede acceder a algún servidor remoto en Internet y ejecutar programas u obtener información en él desde una máquina local.
- ❖ Grupos de noticias. Son un servicio para entablar debate sobre temas diversos, a través del correo electrónico.
- ❖ IRC (Internet Relay Chat). Servicio utilizado para llevar a cabo conversaciones con otros usuarios en tiempo real.
- ❖ Servicios de telefonía. Permiten establecer una conexión con voz entre dos personas conectadas a Internet; en algunos casos no sólo incorporan voz, sino también imagen.

## 1.2. Sistema Web

La evolución de Internet como red de comunicación global y el surgimiento y desarrollo de la Web como servicio imprescindible, creó un excelente espacio para la interacción del hombre con la información, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en Internet.

Se puede definir a un Sistema de Información como el medio por el cual los datos fluyen de una persona o departamento hacia otros y puede ser cualquier cosa, desde la comunicación interna o externa entre los diferentes componentes de la organización y líneas telefónicas, hasta sistemas de cómputo que generan reportes periódicos para varios usuarios. Los Sistemas de Información

proporcionan servicio a todos los demás sistemas de una organización, enlazando todos sus componentes de tal forma que estos trabajen con eficiencia para alcanzar el mismo objetivo.

En las organizaciones o empresas conviven diferentes subsistemas para las distintas funciones, pero para la administración se necesita una mayor disponibilidad de la información, por eso la necesidad de centralizar los datos en bases de datos o bancos de datos centrales.

El banco de datos central recoge la información de las distintas áreas y del entorno, agrupándola para que pueda estar a disposición de toda la organización.

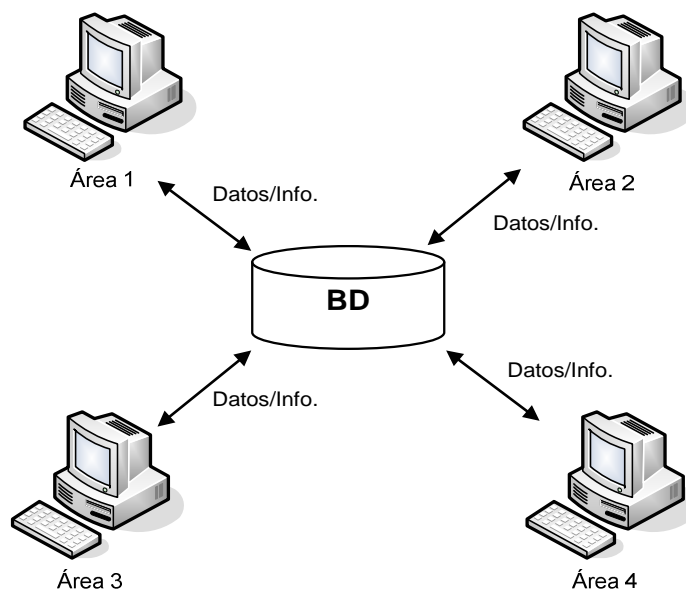


Figura 1.1. Esquema de operación.

Con esta capacidad de manejo de datos, se puede hacer la implementación de un Sistema Web, que es propiamente un Sistema de Información que está instalado en un servidor Web, en el cual puede o no estar la(s) base(s) de datos (banco de datos central); así como las interfaces que permiten al usuario interactuar con el sistema. Al estar instalado en un servidor Web, facilita el ingreso al sistema por medio de uno de los servicios que proporciona Internet, la World Wide Web, haciendo que el sistema pueda ser consultado, modificado e incluso eliminar información desde cualquier parte del mundo, siempre y cuando se tengan los permisos necesarios.

Los Sistemas Web, como expresión de Sistemas de Información, deben poseer los siguientes componentes principalmente: usuarios, mecanismos de entrada y salida de la información, almacenes de datos y mecanismos de recuperación de la información.

El crecimiento de los Sistemas Web ha sido enorme en los últimos años, ya que el ritmo de las empresas u organizaciones, tanto lucrativas como no lucrativas (bancos, secretarías de estado, escuelas, tiendas, etc.), exige cada vez información eficiente, eficaz y confiable. Este tipo de sistemas soluciona las necesidades antes mencionadas, al estar disponible en cualquier hora y momento, presentando información actual, confiable y veraz.

Actualmente, los Sistemas Web se encuentran al alcance de las grandes masas de usuarios por el medio que utiliza, Internet; así se crean las bases de un nuevo modelo en el que los usuarios interactúan directamente con ellos para satisfacer sus necesidades de información.

Las ventajas que ofrecen los Sistemas Web son:

- ❖ Flexibilidad. Posibilidad de interacción con otras aplicaciones así como exportación e importación de datos.
- ❖ Accesibilidad. Acceso a la información desde cualquier lugar, ya sea por medio de una Intranet o Internet.
- ❖ Los usuarios pueden gestionar la información de los sistemas de múltiples centros y/o múltiples empresas desde una única base de datos.
- ❖ Dentro de los recursos disponibles para el desarrollo de los Sistemas Web, existen herramientas open source que facilitan la puesta en marcha de los Sistemas.
- ❖ Mejoran la comunicación con las divisiones, empleados, asociados y en general con todas las áreas de la empresa u organización.
- ❖ Optimiza procesos de administración, comercialización y atención a clientes.
- ❖ La información y los datos son confiables y veraces.

### 1.3. Elementos de un Sistema Web

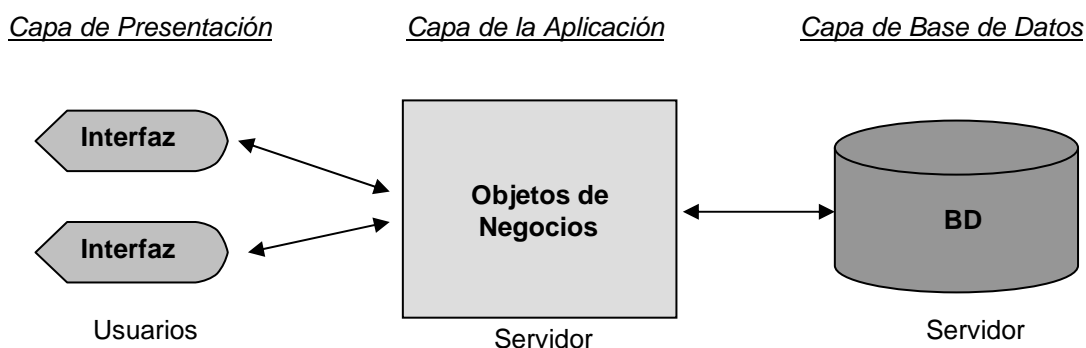


Figura 1.2. Elementos del Sistema Web.

El elemento principal de un Sistema Web es el servidor, éste a su vez proporciona varios servicios. Los servicios son proporcionados mediante una arquitectura cliente-servidor.

La arquitectura cliente-servidor; es una forma de dividir y especializar programas y equipos de cómputo, a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia y permita simplificarlas.

El modelo es utilizado para construir Sistemas de Información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información.

Esta tecnología proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o clientes, resultan en un trabajo realizado por otras computadoras llamadas servidores.

En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes. El trabajo se reparte entre dos computadoras, de acuerdo con la distribución de la lógica de la aplicación hay dos posibilidades:

- ❖ Cliente liviano. Sí el cliente sólo se hace cargo de la presentación.
- ❖ Cliente pesado. Sí el cliente asume también la lógica del negocio.

El principal motivo detrás de esta evolución es la necesidad que tienen las organizaciones (empresas o instituciones públicas o privadas), de realizar sus operaciones ágil y eficientemente, debido a la creciente presión competitiva a la que están sometidas, lo cual se traduce en la necesidad de que su personal sea más productivo, que se reduzcan costos y gastos de operación, al mismo tiempo que se generen productos y servicios más rápidamente y con calidad.

En este contexto, es necesario establecer una infraestructura de procesamiento de información, que cuente con los elementos requeridos para proveer información adecuada, exacta y oportuna en la toma de decisiones y para proporcionar un mejor servicio a los clientes.

El modelo cliente-servidor reúne las características necesarias para proveer esta infraestructura, independientemente del tamaño y complejidad de las operaciones de las organizaciones públicas o privadas, y consecuentemente desempeña un papel importante en este proceso de evolución.

En esta arquitectura, la computadora de cada uno de los usuarios llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores, estos últimos responden a la demanda del cliente que la produjo.

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es Internet. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes, y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura.

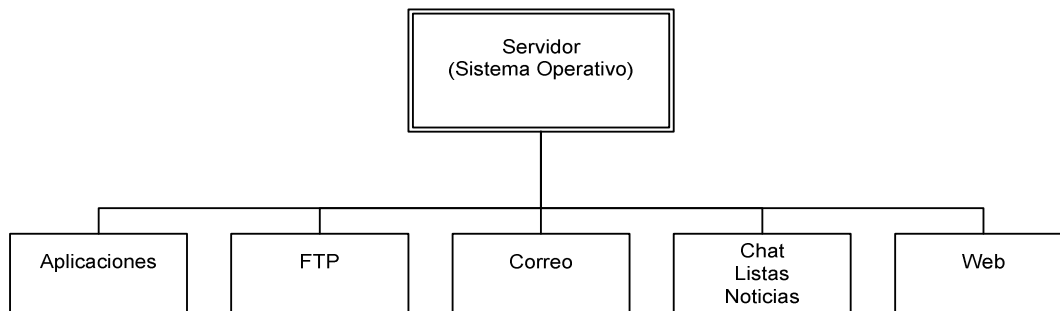
Un cliente es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

Un servidor es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LAN o WAN, para proveer de múltiples servicios a los clientes, tales como impresión, acceso a base de datos, etc. Las funciones del servidor son las siguientes:

- ❖ Espera las solicitudes de los clientes.
- ❖ Ejecuta muchas solicitudes al mismo tiempo.
- ❖ Atiende primero a los clientes prioritarios.
- ❖ Emprende y opera actividades de tareas en segundo plano.
- ❖ Mantenerse activo en forma permanente.

Las ventajas que presenta este modelo es que el servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes. Además que se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que requiere y cierra la conexión dejando libre la red.

En la siguiente figura se muestra la estructura de operación del servidor que a su vez son elementos importantes en los Sistemas Web:



*Figura 1.3. Estructura de operación del servidor.*

### 1.3.1. Servidor (Sistema Operativo)

El término servidor se utiliza para referirse a la computadora, cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos. Un servidor sirve información a las computadoras que se conectan a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información contenida en él.

Los servidores se conectan a la red mediante una interfaz que puede ser una red de área local o una red de área amplia.

Parte fundamental de los servidores, es el sistema operativo. La mayor parte de ellos utilizan alguno de los sistemas operativos Unix.

AT&T es propietario del nombre Unix, el cual es un sistema operativo multitarea, es decir, puede realizar más de una tarea al mismo tiempo. También es un sistema multiusuario, es decir, que soporta actividades de más de una persona a la vez. Además de ser flexible y portable.

La mayor parte del software disponible en forma pública, se distribuye como programas fuente en C que deben ser compilados antes de usarse. Esta tradición de Unix surgió de la necesidad de distribuir programas en forma transportable, para que funcionarán en distintos tipos de computadoras. La mayor parte del sistema Unix está escrito en lenguaje C.

La fuerza de Unix es su portabilidad a través de muchas plataformas de distintos fabricantes, su interconexión en red independiente de los distribuidores y la fuerza de su interfaz para la programación de aplicaciones.

La filosofía Unix se basa en la convicción de que un sistema potente y complejo debe ser simple, general y extensible. Este enfoque queda especialmente reflejado en la forma en que trata los archivos, y en la manera de enfocar sus herramientas.

El sistema Unix contempla los archivos de una manera extremadamente simple y general dentro de un modelo único (modelo jerárquico). Toma de la misma forma los directorios, los archivos ordinarios, los dispositivos, tales como impresoras, discos, teclados y terminales de pantalla. El sistema de archivos oculta al usuario los detalles del hardware subyacente, por ejemplo, no se necesita conocer sobre qué unidad de disco se encuentra un archivo.

Un sistema Unix debe ser considerado por niveles. En el fondo se encuentra el hardware, que consiste en discos, terminales, el CPU, memoria y demás dispositivos. Corriendo sobre el hardware está el sistema operativo; su función es controlar el hardware y proporcionar una interfaz de llamada al sistema a todos los programas. Estas llamadas al sistema, permiten a los programas del usuario crear y manejar procesos, archivos y otros recursos.

En adición al sistema operativo y al sistema de librerías, Unix tiene una serie de programas estándar; éstos incluyen el procesador de comandos (shell), compiladores, editores, procesadores de texto y utilerías de manejo de archivos. Son estos programas los que el usuario invoca desde una terminal.

### 1.3.2. Aplicaciones

Proporciona la infraestructura y servicios clave a las aplicaciones alojadas en el servidor. El servicio es dedicado a ejecutar ciertas aplicaciones software. Se refiere al software instalado en tal servidor para facilitar la ejecución de otras aplicaciones, las cuales pueden ser utilizadas por los usuarios autorizados que se conectan al servidor, como pueden ser hojas de cálculo, editores de texto, etc.

Así como también agrupa los recursos, interfaces de servicios, estado de las aplicaciones, entre otras funciones.

### 1.3.3. FTP (File Transfer Protocol)

Es un programa que funciona con cualquier tipo de archivo, a través del Protocolo de Transferencia de Archivos (*FTP*). Es posible conectar con este servicio mediante clientes FTP o con algún navegador web que lo soporte.

El protocolo de transferencia de archivos es uno de los servicios más importantes, dada la actividad tan grande que ocurre con la transferencia de información entre más de dos computadoras. El servicio permite intercambiar información entre dos o más equipos conectados a la red. Mediante este servicio los usuarios pueden conectarse al servicio FTP y obtener una copia de la información que existe en el servidor en su propia computadora local, o bien, el usuario puede depositar uno o más archivos de su computadora local en el servidor. Además de realizar tareas básicas, como mover, renombrar y trabajar con directorios de forma remota.

Algunos servidores permiten a cualquier persona con acceso a Internet, que pueda acceder a ellos de manera anónima.

En los inicios del servicio, la conexión era por medio de comandos en línea. En la actualidad, ya hay programas gráficos que permiten descargar o subir uno o más archivos al servidor, sin tener que teclear o aprender comandos.

### 1.3.4. Correo

El servicio de correo mueve y almacena los mensajes a través de las redes corporativas y a través de Internet.

Es una aplicación que permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando. Para lograrlo, se utilizan una serie de protocolos, cada uno con una finalidad concreta:

- ❖ SMTP (*Simple Mail Transfer Protocol*). Es el protocolo que se utiliza para que dos servidores de correo intercambien mensajes.
- ❖ POP (*Post Office Protocol*). Se utiliza para obtener los mensajes guardados en el servidor y enviárselos al usuario.
- ❖ IMAP (*Internet Message Access Protocol*). Su finalidad es descargar los encabezados; requiere conexión para poder leer los mensajes completos.

También de ciertos programas y técnicas:

- ❖ MTA (*Mail Transport Agent*). Son programas que se encargan de distribuir a través de la red los mensajes generados en el sistema.

- ❖ MUA (*Mail User Agent*). Son programas que permiten la edición, lectura y respuesta de correo electrónico, por ejemplo: Eudora, Outlook, Pine, etc.
- ❖ MIME (*Multipurpose Internet Mail Extensions*). Es una técnica para codificar archivos y anexarlos a un mensaje SMTP. Permite enviar mensajes binarios como parte de un mensaje de correo electrónico.

Así pues, el servicio de correo consta en realidad de dos partes, un servicio SMTP que será el encargado de enviar y recibir mensajes, y un servicio POP/IMAP que permitirá a los usuarios obtener sus mensajes. Para obtenerlos del servidor, los usuarios se sirven de clientes; es decir, programas que implementan un protocolo POP/IMAP. En algunas ocasiones el cliente se ejecuta en la máquina del usuario. Sin embargo, existe otra posibilidad, que el cliente de correo no se ejecute en ésta, tal es el caso de los clientes vía web.

En cualquier caso, los protocolos son inseguros en cuanto a que los mensajes viajan en claro por la red, es decir, es fácil obtener mensajes y contraseñas. Para ello se suele añadir una capa SSL, es un método de encriptación que puede implementar tanto el servidor como el cliente.

### 1.3.5. Chat, Listas y Noticias

El servicio de Chat permite intercambiar información a una gran cantidad de usuarios ofreciendo la posibilidad de llevar a cabo discusiones en tiempo real. Otra opción para usuarios que buscan la discusión en tiempo real, IRC (*Internet Relay Chat*), un sistema de comunicación que permite mantener conversaciones en tiempo real con otros usuarios. IRC ofrece diferentes canales y cada canal trata un determinado tema o está compuesto por un determinado círculo de participantes.

Las listas ofrecen una manera mejor de manejar listas de correo electrónico, bien sean discusiones interactivas abiertas al público o listas unidireccionales de anuncios, boletines de noticias o publicidad. Sirve para entablar debates sobre temas diversos. Se basa en el servicio de correo electrónico. Los mensajes que se envían a las listas se hacen públicos y cualquier persona puede enviar una contestación. Este servicio es de gran utilidad para resolver dudas difíciles, cuya respuesta sólo la conozcan unas pocas personas en el mundo.

El servicio de noticias actúa como fuente de distribución y entrega, para los millares de grupos de noticias públicos actualmente accesibles a través de la red de noticias USENET.

### 1.3.6. Web

Es un programa que implementa el protocolo HTTP (*HyperText Transfer Protocol*). Este protocolo está diseñado para transferir las páginas web o páginas HTML.

Sin embargo, aunque HTTP y HTML estén íntimamente ligados no hay que confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

El servicio web se encarga de mantenerse a la espera de peticiones HTTP llevadas a cabo por un cliente llamado navegador. El navegador o browser realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. El servidor tan sólo se limita a transferir el código de la página sin llevar a cabo alguna interpretación de la misma. Sobre el servicio web se puede disponer de aplicaciones web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones. Hay que distinguir entre:

- ❖ Aplicaciones en el lado del cliente. El cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o JavaScript; el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones o scripts, mediante el uso de plugins.

- ❖ Aplicaciones en el lado del servidor. El servidor ejecuta la aplicación, ésta una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo.

#### 1.4. Software libre

En 1984, Richard Stallman comenzó a trabajar en el proyecto GNU, y un año más tarde fundó la FSF (*Free Software Foundation*). Stallman introdujo una definición para free software y el concepto de copyleft, el cual desarrolló para dar a los usuarios libertad y para restringir las posibilidades de apropiación del software.

Es el software que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y aunque conserve su carácter de libre, puede ser vendido comercialmente.

De acuerdo con la definición, el software es libre sí garantiza las siguientes libertades:

- ❖ Libertad 0. Ejecutar el programa con cualquier propósito, ya sea privado, educativo, público, etc.
- ❖ Libertad 1. Analizar cómo funciona el programa, modificarlo y adaptarlo a las necesidades. El acceso al código fuente es una condición previa para esto.
- ❖ Libertad 2. Distribuir copias del programa de manera que se pueda ayudar a cualquiera.
- ❖ Libertad 3. Mejorar el programa y hacer públicas las correcciones, de forma que se beneficie toda la comunidad. El acceso al código fuente es un requisito previo para esto.

Es importante señalar que la libertad 2, hace referencia a modificar y redistribuir el software libremente, autorizado bajo algún tipo de licencia de software libre que beneficie a la comunidad.

Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado, para ejercer actos de explotación legales. Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el usuario. Desde el punto de vista del software libre, existen distintas variantes del concepto o grupos de licencias.

Las libertades definidas anteriormente están protegidas por las licencias de software libre, de las cuales una de las más utilizadas es la GNU GPL (Licencia Pública General). El autor conserva los derechos de autor (*copyright*), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. En el sitio web de la FSF (<http://www.gnu.org/licenses/>) hay una lista de licencias que cumplen las condiciones impuestas por la GNU GPL y otras que no.

Existe una cierta controversia sobre la seguridad del software libre frente al software no libre. Un método usado de forma habitual para determinar la seguridad relativa de los productos, es determinar cuántos fallos de seguridad no corregidos o parchados existen en cada uno de los productos involucrados. Por lo general; los usuarios de este método recomiendan que cuando un producto no proporcione un método de corregir los fallos de seguridad, no se use dicho producto, al menos hasta que no esté disponible un arreglo.

En la siguiente tabla se muestra el software libre más utilizado, así como sus principales características.



Elemento	Herramientas
<b>Sistema operativo</b>	<ul style="list-style-type: none"> <li>❖ Linux. Es multitarea, multiusuario, shells programables, independencia de dispositivos, portabilidad, comunicaciones y capacidades de red. Muy utilizado en computadoras personales y servidores.</li> <li>❖ Solaris. Es portable, escalable. El código binario esta disponible. Es un sistema modular, por lo que se le puede añadir y quitar módulos sin interrumpir el servicio. Es un sistema escalable, por lo que puede adaptarse a nuevos cambios. Posee herramientas administrativas. El kernel es dinámico. Es un sistema operativo Unix, un sistema bastante seguro, de alta disponibilidad y escalabilidad. Posee herramientas que permiten la creación de sistemas de archivos de cualquier tamaño, sin importar la configuración del hardware. Muy utilizado tanto para una computadora personal como para un servidor.</li> </ul>
<b>Aplicaciones</b>	<ul style="list-style-type: none"> <li>❖ JBoss AS. Ofrece una plataforma de alto rendimiento para aplicaciones de e-bussines. Cumple con los estándares obligatorios. Confiable a nivel de empresa. Orientado a arquitectura de servicios. Flexibilidad consistente. Servicios de middleware para cualquier objeto Java. Soporte completo y ayuda profesional.</li> <li>❖ Tomcat (también llamado Jakarta Tomcat o Apache Tomcat). Es considerado como un servidor de aplicaciones. Implementa las especificaciones de los servlets y de JavaServer Pages (<i>JSP</i>), para esto cuenta con el compilador Jasper, que transforma JSPs en servlets. Tomcat puede funcionar como servidor web por sí mismo. Opera de tal manera en entornos de desarrollo poco exigentes en términos de velocidad y de manejo de transacciones. Funciona en cualquier sistema operativo que disponga de la máquina virtual de Java.</li> <li>❖ OpenOffice. Es un conjunto de aplicaciones de oficina que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos. Está disponible para muchos y diversos sistemas operativos, como Microsoft Windows y varios sistemas de tipo Unix. Soporta el estándar OpenDocument para el intercambio de datos.</li> <li>❖ Gnome Office. Es un conjunto de aplicaciones de oficina que se compone de un procesador de textos (AbiWord), hoja de cálculo (Gnumeric), aplicación gestora de base de datos (GnomeDB), aplicación para manipular gráficos (GIMP), además de otras aplicaciones pendientes de anexarse al proyecto como puede ser Magic Point que es un editor de presentaciones.</li> <li>❖ Mozilla Thunderbird (anteriormente Minotaur). Es un cliente de correo electrónico. El objetivo de Thunderbird es desarrollar un Mozilla más liviano y rápido, mediante la extracción y rediseño del gestor de correo del Mozilla; es multiplataforma y utiliza el lenguaje de interfaz XUL. Soporta IMAP/POP, correo HTML, noticias, RSS, etiquetas, corrector ortográfico incorporado, soporte de extensiones y skins, buscadores, cifrado PGP, un filtro bayesiano de correo basura (spam), entre otras características. Hay una versión que puede ser transportada y usada directamente sin necesidad de instalarse, desde una memoria USB o CD.</li> <li>❖ Evolution. Es un gestor de información personal y de trabajo en grupo para GNOME. Combina administración de correo electrónico, calendario, agenda y lista de tareas. Su interfaz gráfica y funcionalidad imita bastante a la aplicación Microsoft Outlook. Puede conectarse a servidores Microsoft Exchange usando su interfaz web y un añadido conocido como Connector.</li> <li>❖ Además de muchas otras aplicaciones.</li> </ul>

Tabla 1.1. Software libre.

## 1.5. Aplicaciones web

Una aplicación web; es aquella que los usuarios utilizan accediendo a un servidor web a través de Internet o de una Intranet. Las aplicaciones web son ampliamente utilizadas debido a la practicidad del navegador web como cliente ligero. La habilidad para actualizar y mantener las aplicaciones sin distribuir e instalar software en miles de potenciales clientes, es otra razón de su popularidad.

Las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

Las interfaces web tienen ciertas limitantes en la funcionalidad del cliente. Se utiliza comúnmente lenguajes interpretados del lado del cliente para añadir más funcionalidad, especialmente para crear una experiencia interactiva que no requiera recargar la página constantemente. Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías del lado del servidor, como por ejemplo PHP; técnica de desarrollo web que utiliza una combinación de diversas tecnologías.

Aunque muchas variaciones son posibles, una aplicación web está comúnmente estructurada como una aplicación de tres capas. En su forma más común, el navegador web es la primer capa, un motor usando alguna tecnología web dinámica (por ejemplo; CGI, PHP, Java Servlets o ASP) es la capa de en medio, y una base de datos como última capa. El navegador manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

Por último, cabe destacar que el software libre es una excelente elección para el desarrollo de aplicaciones web. Cuando se desarrolla un software libre, todo el mundo colabora y el software gana, se descubren errores, mejoras, el usuario puede ver el código, lo puede modificar y aprende sobre él, se realza la creatividad del usuario y no existe un monopolio, por ende, este producto no es caro y está al alcance de todos.

No se paga por utilizar el software, además se puede configurar o reprogramar para adaptarse a las necesidades requeridas, en comparación con el software de licencia copyright.

El software para el desarrollo se puede obtener directamente de Internet; así como también las actualizaciones y parches o correcciones. Parte importante son los manuales, los cuales provienen de la misma fuente que el software.

Los Sistemas Web al ser una aplicación que automatiza procesos y maneja información, necesita de un proceso formal de desarrollo, que permita establecer adecuadamente las etapas, tareas y herramientas necesarias, por ello es fundamental elegir correctamente los elementos que conformaran nuestra aplicación.

**CAPÍTULO 2**  
**METODOLOGÍA Y HERRAMIENTAS DE**  
**DESARROLLO**

El desarrollo de un Sistema Web como de cualquier otro sistema, necesita de una metodología que nos permita definir las etapas, actividades, productos terminados y herramientas; convirtiéndose en un punto estratégico para conseguir la funcionalidad y rendimiento de nuestra aplicación y así cumplir con su objetivo de manera eficiente.

El ciclo de vida de un sistema es un enfoque por fases del análisis y diseño, el cual sostiene que los sistemas son desarrollados de mejor manera mediante el uso de un ciclo específico de actividades del analista y del usuario. Según James Senn, existen tres estrategias para el desarrollo de sistemas; el método clásico del ciclo de vida de desarrollo de sistemas, el método de desarrollo por análisis estructurado y el método de construcción de prototipos de sistemas.

El método clásico del ciclo de vida para el desarrollo de sistemas, es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implementar un sistema. La metodología consta de las siguientes actividades a realizar en el desarrollo del proyecto:

- ❖ **Análisis.** En esta etapa se conversa con varias personas para reunir detalles de los procesos de la división, las opiniones del porqué ocurren las cosas, las soluciones que proponen y las ideas para cambiar los procesos. La observación en condiciones reales de las actividades del trabajo y en algunas ocasiones, muestras de formas y documentos con el fin de comprender el proceso en su totalidad. Conforme se reúnen los detalles, se estudian los requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema.
- ❖ **Diseño.** El diseño del sistema produce los detalles de cómo cumplirá con los requerimientos identificados durante la fase de análisis. Es decir; identificar reportes y demás salidas que debe producir el sistema, se indican los datos de entrada, detalle de los procedimientos de cálculo, etc. El proceso traduce los requerimientos en una representación del sistema que pueda ser establecida de forma que se obtenga la calidad requerida, antes de que comience la implementación o desarrollo del sistema. El Lenguaje Unificado de Modelado (*UML*) es una de las herramientas más utilizadas en esta etapa, ya que es una técnica de modelado confiable, permite visualizar, especificar y construir el sistema de forma gráfica. Con lo que también se consigue documentar el sistema.
- ❖ **Programación.** A partir del diseño final generado en la anterior etapa, en esta de programación, se codificarán y probarán los nuevos programas usando herramientas apropiadas. La documentación es parte esencial para probar el programa y llevar a cabo el mantenimiento una vez terminado. Durante el desarrollo de esta actividad, se debe tener en cuenta el entorno tecnológico donde se implementará el sistema. Los resultados son los programas probados y la base de datos afinada.
- ❖ **Implementación.** Es el proceso de verificar e instalar el nuevo sistema, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarlo. Una vez implementado el sistema se debe llevar a cabo el mantenimiento, ya que se pueden experimentar cambios.

A continuación se presenta un resumen de las actividades y productos de cada una de las etapas del proceso para el desarrollo de sistemas.

<b>Etapa</b>	<b>Actividad</b>	<b>Producto</b>
<b>Análisis</b>	<ul style="list-style-type: none"> <li>❖ Estudio de factibilidad.</li> <li>❖ Revisión de documentos utilizados.</li> <li>❖ Entrevistas.</li> <li>❖ Observación del proceso actual.</li> </ul>	<ul style="list-style-type: none"> <li>❖ Alcance del proyecto.</li> <li>❖ Modelo lógico actual de procesos.</li> <li>❖ Esquema lógico actual de datos.</li> <li>❖ Requisitos de usuarios.</li> </ul>
<b>Diseño</b>	<ul style="list-style-type: none"> <li>❖ Identificar datos de entrada y salida, para la base de datos.</li> <li>❖ Procesamiento de datos de entrada y salida.</li> <li>❖ Identificar usuarios y sus necesidades.</li> </ul>	<ul style="list-style-type: none"> <li>❖ Diseño de la base de datos, estructuración y normalización de tablas de datos.</li> <li>❖ Especificaciones del proceso. Flujo, operación, cálculos, validaciones y seguridad. Diagramas UML.</li> <li>❖ Identificación de reportes y demás salidas que debe producir el sistema. Diagramas UML.</li> <li>❖ Definición de módulos y especificaciones de programas.</li> </ul>
<b>Programación</b>	<ul style="list-style-type: none"> <li>❖ Escribir rutinas y procedimientos.</li> <li>❖ Especificar contenido de pantallas y características.</li> <li>❖ Probar cada módulo.</li> <li>❖ Documentación y especificaciones.</li> <li>❖ Estructurar archivos.</li> </ul>	<ul style="list-style-type: none"> <li>❖ Codificación, es decir, conjunto de instrucciones en código fuente.</li> <li>❖ Creación de interfaces.</li> <li>❖ Pruebas preeliminares.</li> <li>❖ Manual de usuario.</li> </ul>
<b>Implementación</b>	<ul style="list-style-type: none"> <li>❖ Identificar áreas críticas.</li> <li>❖ Instalar aplicación.</li> <li>❖ Capacitación y asesoría a usuarios.</li> <li>❖ Realizar pruebas con usuarios y generar reporte.</li> <li>❖ Hacer cambios identificados.</li> </ul>	<ul style="list-style-type: none"> <li>❖ Preparar plan de conversión.</li> <li>❖ Prueba de funcionalidad de todo el sistema.</li> <li>❖ Impartir formación a los usuarios.</li> <li>❖ Conversión y/o carga de datos iniciales.</li> <li>❖ Pasar al nuevo sistema.</li> <li>❖ Prueba de aceptación.</li> <li>❖ Integrar documentación técnica y administrativa.</li> <li>❖ Mantenimiento.</li> </ul>

*Tabla 2.1. Etapas, actividades y productos de la metodología para el desarrollo de sistemas.*

## 2.1. UML (Lenguaje Unificado de Modelado)

A continuación se describirá la metodología para llevar a cabo la etapa de análisis y parte del diseño del sistema, lo último en conjunto con la sección 2.2.

Desde los inicios de la informática, se han estado utilizando distintas formas de representar los diseños; personal o con algún modelo gráfico. La falta de estandarización en la manera de

representar gráficamente un modelo, impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores. Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores; sino también para servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (*UML: Unified Modeling Language*). UML se ha convertido en ese estándar para representar y modelar la información con la que se trabaja en la fase de análisis, especialmente de diseño. Además, UML es una especificación de la programación orientada a objetos (POO).

El lenguaje UML tiene una notación gráfica muy expresiva, que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue.

### 2.1.1. Historia

El lenguaje UML comenzó a gestionarse en octubre de 1994, cuando Rumbaugh se unió a la compañía Rational fundada por Booch (dos importantes investigadores en el área de metodología del software). El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (*Object Modeling Technique*). El primer borrador apareció en octubre de 1995. En esa misma época otro importante investigador, Jacobson, se unió a Rational y se incluyeron ideas suyas. Además, este lenguaje se abrió a la colaboración de otras empresas para que aportaran sus ideas; todas estas colaboraciones condujeron a la definición de la primera versión de UML.

Esta primera versión se ofreció a un grupo de trabajo para convertirlo en 1997 en un estándar del OMG (*Object Management Group*). Este grupo, que gestiona estándares relacionados con la tecnología orientada a objetos (metodologías, bases de datos, etc.), propuso una serie de modificaciones y una nueva versión de UML; la 1.1, que fue adoptada por el OMG como estándar en noviembre de 1997. Desde aquella versión ha habido varias revisiones que gestiona la OMG, conocidas como Revision Task Force (*RTF*).

UML no le da un protagonismo excesivo al diagrama de clases, éste representa una parte importante del sistema, pero sólo una vista estática, es decir, muestra al sistema parado. Se sabe su estructura pero no se sabe qué le sucede a sus diferentes partes cuando el sistema empieza a funcionar. UML introduce nuevos diagramas que representan una visión dinámica del sistema. Gracias al diseño de la parte dinámica del sistema; se puede dar cuenta en la fase de diseño, de problemas de la estructura al propagar errores o de las partes que necesitan ser sincronizadas, así como del estado de cada una de las instancias en cada momento. El diagrama de clases continúa siendo muy importante, pero se debe tener en cuenta que su representación es limitada, y que ayuda a diseñar un sistema robusto con partes reutilizables, pero no a solucionar problemas de propagación de mensajes ni de sincronización o recuperación ante estados de error. En resumen, un sistema debe estar bien diseñado, pero también debe funcionar bien.

UML permite la modificación de todos sus miembros mediante estereotipos y restricciones. Un estereotipo nos permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML. Una restricción identifica un comportamiento forzado de una clase o relación; es decir, mediante la restricción estamos forzando el comportamiento que debe tener el objeto al que se le aplica.

Tal como indica su nombre, UML es un lenguaje de modelado. Un modelo es una simplificación de la realidad. El objetivo del modelado de un sistema, es capturar las partes esenciales de éste. Para facilitar el modelado, se realiza una abstracción y se plasma en una notación gráfica, esto se conoce como modelado visual. El modelado visual permite manejar la complejidad de los sistemas a analizar o diseñar.

UML sirve para el modelado completo de sistemas complejos y no tan complejos. Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando esta metodología, se puedan implementar en cualquier lenguaje de programación.

Un modelo UML esta compuesto por tres clases de bloques de construcción:

- ❖ Elementos. Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.).
- ❖ Relaciones. Son los vínculos que unen los elementos entre sí.
- ❖ Diagramas. Son colecciones de elementos con sus relaciones.

### 2.1.2. Objetivos

Un lenguaje proporciona un vocabulario y reglas para permitir la comunicación; en este caso, UML es ante todo un lenguaje, que se centra en la representación gráfica de un sistema y además indica como crear y leer los modelos. Los objetivos de UML son muchos, pero se pueden sintetizar a través de sus funciones:

- ❖ Visualizar. UML permite expresar de una forma gráfica un sistema, de manera que otro lo pueda entender.
- ❖ Especificar. UML permite definir cuáles son las características de un sistema antes de su construcción.
- ❖ Construir. A partir de los modelos especificados, se pueden desarrollar los sistemas diseñados.
- ❖ Documentar. Los propios elementos gráficos, sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.
- ❖ Proporcionar a los usuarios un lenguaje de modelado visual expresivo y utilizable para el desarrollo e intercambio de modelos significativos.
- ❖ Proporcionar mecanismos de extensión y especialización.
- ❖ Ser independiente del proceso de desarrollo y de los lenguajes de programación.
- ❖ Proporcionar una base formal para entender el lenguaje de modelado.
- ❖ Fomentar el crecimiento del mercado de las herramientas orientadas a objetos.
- ❖ Soportar conceptos de desarrollo de alto nivel, como pueden ser colaboraciones y componentes.

### 2.1.3. Diagramas

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En UML, un diagrama ofrece una vista del sistema a modelar; para poder representar correctamente un sistema, ofrece una amplia variedad de diagramas para visualizar desde varias perspectivas. Los diagramas son los siguientes:

- ❖ Diagrama de casos de uso.
- ❖ Diagrama de clases.
- ❖ Diagrama de objetos.
- ❖ Diagrama de secuencia.
- ❖ Diagrama de colaboración.

- ❖ Diagrama de estados.
- ❖ Diagrama de actividades.
- ❖ Diagrama de componentes.
- ❖ Diagrama de despliegue.

Cada diagrama usa la anotación pertinente y la suma de estos diagramas crean las diferentes vistas. Las vistas existentes en UML son:

- ❖ Vista casos de uso. Se forma con los diagramas de casos de uso, colaboración, estados y actividades.
- ❖ Vista de diseño. Se forma con los diagramas de clases, objetos, colaboración, estados y actividades.
- ❖ Vista de procesos. Se forma con los diagramas de la vista de diseño. Recalcando las clases y objetos referentes a procesos.
- ❖ Vista de implementación. Se forma con los diagramas de componentes, colaboración, estados y actividades.
- ❖ Vista de despliegue. Se forma con los diagramas de despliegue, interacción, estados y actividades.

Se dispone de dos tipos diferentes de diagramas, los que dan una vista estática del sistema y los que dan una visión dinámica. Los diagramas estáticos son:

- ❖ Diagrama de clases. Define las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.
- ❖ Diagrama de objetos. Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Representa las instancias y como se relacionan entre ellas. Se da una visión de casos reales.
- ❖ Diagrama de componentes. Muestra la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.
- ❖ Diagrama de despliegue. Representa los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un sistema complejo. Sirve como resumen e índice.
- ❖ Diagrama de casos de uso. Representa los casos de uso, actores y sus relaciones. Muestra quién puede hacer qué y las relaciones que existen entre acciones (casos de uso). Son muy importantes para modelar y organizar el comportamiento del sistema.

Los diagramas dinámicos son:

- ❖ Diagrama de secuencia y Diagrama de colaboración. Muestran a los diferentes objetos y las relaciones que pueden tener entre ellos así como los mensajes que se envían. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.



- ❖ Diagrama de estados. Define los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.
- ❖ Diagrama de actividades. Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

Como se puede ver, el número de diagramas es muy alto, en la mayoría de los casos, excesivo. UML permite definir sólo los necesarios, ya que no todos son indispensables en los proyectos.

Así se tiene que para una aplicación sencilla se debe realizar entre tres y seis tipos de diagramas, y para una aplicación compleja utilizar todos los tipos. Hay que notar que UML esta pensado para el modelado tanto de pequeños sistemas como de sistemas complejos, y debemos tener en cuenta que los sistemas complejos pueden estar compuestos por millones de líneas de código y ser realizados por equipos de centenares de programadores. Desde el punto de vista de UML no deja de ser un proyecto mediano tirando a pequeño.

A continuación se describirán los diagramas a utilizar en este proyecto.

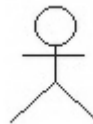
### 2.1.3.1. Diagrama de casos de uso

El diagrama de casos de uso representa la forma en cómo un cliente (actor) opera con el sistema, además de la forma, tipo y orden en cómo los elementos interactúan (operaciones o casos de uso).

Se emplean para visualizar el comportamiento del sistema, una parte de él o de una sola clase; de forma que se pueda conocer como responde esa parte del sistema. El diagrama de casos de uso es muy útil para definir cómo debería ser el comportamiento del sistema, ya que sólo especifica cómo deben comportarse y no cómo están implementadas las partes que define. Por ello es muy buena técnica el documentar partes del código que deben ser reutilizables por otros desarrolladores. Un caso de uso especifica un requerimiento funcional, es decir, indica esta parte debe hacer esto cuando pase esto.

#### 2.1.3.1.1. Elementos

- ❖ Actor. Es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra *rol*, pues con esto se especifica que un *Actor* no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema. La misma persona física puede interpretar varios papeles como actores distintos, el nombre del actor describe el papel desempeñado. Los actores pueden ser: personas que usan el sistema, personas que administran o proporcionan mantenimiento al sistema y dispositivos o sistemas con los que el sistema interactúa. Los actores se representan gráficamente con la siguiente figura:



- ❖ Caso de uso. Es una operación o tarea específica, que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación de otro caso de uso. Es decir, es una descripción de la secuencia de interacciones o pasos que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Se representa en el diagrama mediante una elipse con el nombre del caso de uso en su interior,

denota un requerimiento solucionado por el sistema. El nombre del caso de uso debe reflejar la tarea que el actor desea llevar a cabo usando el sistema.



❖ Relaciones. Existen tres tipos de relaciones en los diagramas de casos de uso:

- Asociación. Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.



- Dependencia o Instancia. Es una forma muy particular de relación entre casos de uso, en la cual un caso depende de otro, es decir, se crea o instancia. La relación se representa con una flecha punteada.



- Generalización. Este tipo de relación es una de las más utilizadas, cumple una doble función dependiendo de su estereotipo, que puede ser:

\* Herencia (<<extends>>). Se recomienda utilizar cuando un caso de uso es similar a otro, en cuanto a características.

\* Uso (<<uses>>). Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso, y no se desea mantener duplicada la descripción de la característica.

Se representa con la siguiente figura:



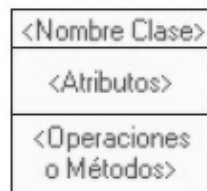
### 2.1.3.2. Diagrama de clases

Forma parte de la vista estática del sistema. El diagrama de clases, es el diagrama principal para el análisis y diseño. Representa las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye la especificación de atributos y operaciones.

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso, etc.

#### 2.1.3.2.1. Elementos


- ❖ Clase. Es la unidad básica que encapsula toda la información de un objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (sistema). Las clases se representan por un rectángulo que posee tres divisiones:





En donde:

- Superior. Contiene el nombre de la clase.


- Intermedio. Contiene los atributos o variables de instancia que caracterizan a la clase. Un atributo representa alguna propiedad de la clase que se encuentra en todas las instancias de la misma. Los atributos pueden representarse sólo mostrando su nombre y su tipo, e incluso su valor por defecto. Los atributos o características de una clase son los que definen el grado de comunicación y visibilidad de ellos con el entorno, pueden ser de tres tipos, estos son:


\* Public ( +,  ). Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, puede ser utilizado por cualquier método.


\* Private ( -,  ). Indica que el atributo sólo será visible desde dentro de la clase, sólo sus métodos lo pueden utilizar.

\* Protected ( #,  ). Indica que el atributo no será visible desde fuera de la clase, pero si podrá ser utilizado por métodos de la clase, además de las subclases que se deriven.

- Inferior. Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno, dependiendo del nivel de visibilidad:

\* Public ( +,  ). Indica que el método será visible dentro o fuera de la clase, es decir, puede ser utilizado por cualquier atributo.

\* Private ( -,  ). Indica que el método sólo será visible desde dentro de la clase.

\* Protected ( #,  ). Indica que el método no será visible desde fuera de la clase, pero si podrá ser utilizado por atributos de la misma clase, además de las subclases que se deriven.

- ❖ Relaciones. Ya definido el concepto de clase, es necesario explicar cómo se pueden interrelacionar dos o más clases, cada una con características y objetivos diferentes. Antes, es necesario definir el concepto de cardinalidad de relación, se refiere al grado y nivel de dependencia entre clases. Se anotan en cada extremo de la relación y éstas pueden ser:

\* Uno o muchos. 1...\*(1...n).

\* Cero o muchos. 0...\*(0...n).

\* Número fijo. N (n denota el número).

- Herencia (Especialización/Generalización). Indica que una subclase hereda los métodos y atributos especificados por una superclase, por ende, la subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la superclase (public y protected). UML nos permite modificar la relación con un estereotipo y cuatro restricciones:

Estereotipo.

\* Implementation. El hijo hereda la implementación del padre, sin publicar ni soportar sus interfaces.

Restricciones.

\* Complete. La herencia ya no permite más hijos.

\* Incomplete. Podemos incorporar más hijos a la generalización.

\* Disjoint. Sólo puede tener un tipo en tiempo de ejecución, una instancia del padre sólo podrá ser de un tipo de hijo.

\* Overlapping. Puede cambiar de tipo durante su vida, una instancia del padre puede ir cambiando de tipo entre los de sus hijos.

Dicha relación se denota así:



- Agregación. Para modelar objetos complejos, no bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, se tienen dos posibilidades:

\* Por valor. Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido, está condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada *Composición*; es decir, el objeto base se construye a partir del objeto incluido, "parte/todo". Se representa de la siguiente forma:



\* Por referencia. Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada *Agregación*; es decir, el objeto base utiliza al incluido para su funcionamiento. Se utiliza la siguiente figura para representar este tipo de relación:



- Asociación. La relación entre clases conocida como asociación, permite asociar o unir objetos que colaboran entre sí. Cabe destacar que no es una relación fuerte; es decir, el tiempo de vida de un objeto no depende del otro. Esta dada por:



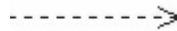
- Dependencia o Instancia. Es una relación de uso, es decir, una clase usa a otra que la necesita para su cometido. Se representa con una flecha discontinua, va desde la clase utilizadora a la clase utilizada. Con la dependencia se muestra que un cambio en la clase utilizada puede afectar el funcionamiento de la clase utilizadora, pero no al contrario. Aunque las dependencias se pueden crear tal cual, es decir, sin ningún estereotipo; UML permite dar más significado a las dependencias mediante el uso de estereotipos.

\* Bind. La clase utilizada es una plantilla y necesita de parámetros para ser utilizada.

\* Derive. Se utiliza al indicar relaciones entre dos atributos, es decir, que el valor de un atributo depende directamente del valor de otro.

\* InstanceOF. Indica que el objeto origen es una instancia del objeto destino.

La relación se simboliza así:



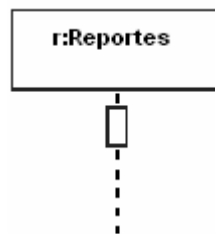
### 2.1.3.3. Diagrama de secuencia

El diagrama de secuencia forma parte del modelado dinámico del sistema. Se modelan las llamadas entre clases desde un punto concreto del sistema. Es útil para observar la vida de los objetos en el sistema, identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de llamadas entre los componentes del sistema.

En el diagrama de secuencia se muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Estos representan a un usuario o a un actor, y los objetos y componentes con los que interactúa durante la ejecución de un caso de uso; en otras palabras, representa un único escenario de caso de uso o flujo de eventos. También permiten observar el flujo de mensajes de un objeto a otro, y como tal, representan los métodos y los eventos soportados por un objeto o una clase.

#### 2.1.3.3.1. Elementos

- ❖ Línea de vida de un objeto. Se refiere al tiempo de ejecución o vida del objeto durante la interacción. Se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal, que denotan la ejecución de métodos (activación). El rectángulo de encabezado contiene el nombre del objeto y el de su clase, con el siguiente formato: nombreObjeto:nombreClase.



- ❖ Activación. Muestra el período de tiempo, en el cual el objeto se encuentra desarrollando alguna operación, bien sea por sí mismo o por medio de delegar a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto.



- ❖ Mensaje. El envío de mensajes entre objetos, se denota mediante una línea sólida, dirigida desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.



- ❖ Tiempos de transición. En un entorno de objetos concurrentes o de demoras en la recepción de mensajes, es útil agregar nombres a los tiempos de salida y llegada de mensajes.

## 2.2. Bases de datos

Para comprender lo que es una base de datos, hay que ir conociendo cada uno de los elementos que la conforman; por lo que a continuación se describirán:

### 2.2.1. Dato

Un dato es una representación simbólica (numérica, alfabética, etc.) de un atributo o característica de una entidad. El dato no tiene valor semántico (sentido) en sí mismo, pero convenientemente procesado se puede utilizar en la realización de cálculos o toma de decisiones. Esto se puede observar en la siguiente figura:

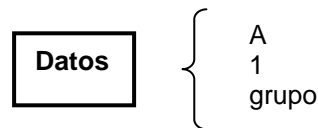


Figura 2.1. Dato.

### 2.2.2. Información

Es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario, para que un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a información, primero se debe guardar lógicamente en archivos. En la figura 2.2 se puede observar los pasos para obtener información:

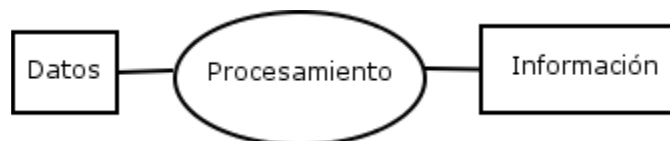


Figura 2.2. Información.

### 2.2.3. Campo

Es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del usuario representa una característica de un individuo u objeto.

### 2.2.4. Registro

Es una colección de campos de iguales o de diferentes tipos.

### 2.2.5. Archivo

Se puede definir como un conjunto de registros almacenados siguiendo una estructura homogénea.

### 2.2.6. Base de datos

Es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. En una base de datos, además de datos, también se almacena su descripción.

Es decir; el contenido de una base de datos engloba a la información concerniente (almacenados en archivos) de una organización, de tal manera que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del manejo del sistema.

### 2.2.7. Sistema Manejador de Base de Datos (DBMS)

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.

El objetivo primordial de un sistema manejador base de datos es proporcionar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfase entre los usuarios y la base de datos.

Las funciones principales de un sistema manejador de bases de datos, son:

- ❖ Crear y organizar la base de datos.
- ❖ Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedados rápidamente.
- ❖ Manejar los datos de acuerdo a las peticiones de los usuarios.
- ❖ Registrar el uso de las bases de datos.
- ❖ Interacción con el manejador de archivos. Esto a través de las sentencias en DML (*Lenguaje de Manipulación de Datos*) al comando del sistema de archivos. Así el DBMS es el responsable del verdadero almacenamiento de los datos.
- ❖ Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en la base de datos.
- ❖ Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- ❖ Seguridad e integridad. Consiste en contar con mecanismos que permitan el control de la consistencia de los datos, evitando que estos se vean perjudicados por cambios no autorizados o previstos.

El DBMS es conocido también como Gestor de Base de Datos. La figura 2.3 muestra el DBMS como interfase 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.

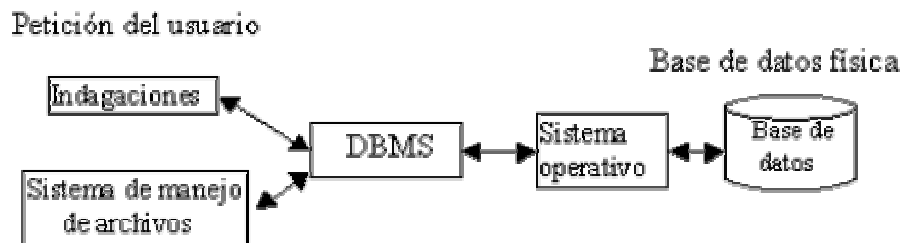


Figura 2.3. Sistema Manejador de Base de Datos.

En sí, un sistema manejador de base de datos es el corazón de la base de datos, ya que se encarga del control total de los posibles aspectos que la puedan afectar.

### 2.2.8. Administrador de base de datos (DBA)

Es la persona o equipo de personas profesionales responsables del control y manejo del sistema de base de datos, generalmente tiene(n) experiencia en DBMS, diseño de base de datos, sistemas operativos, comunicación de datos, hardware y programación. Las funciones principales de un DBA son:

- ❖ Definición de esquema. Es el esquema original de la base de datos, se crea escribiendo un conjunto de definiciones que son traducidas por el compilador de DDL (*Lenguaje de Definición de Datos*) a un conjunto de tablas que son almacenadas permanentemente en el diccionario de datos.
- ❖ Definición de la estructura de almacenamiento y del método de acceso. Estructuras de almacenamiento y de acceso adecuados, se crean escribiendo un conjunto de definiciones que son traducidas por el compilador del lenguaje de almacenamiento y definición de datos.
- ❖ Concesión de permisos para el acceso a los datos. Permite al administrador de la base de datos, regular las partes de la misma que van a ser accedidas por varios usuarios.
- ❖ Especificación de limitantes de integridad. Es una serie de restricciones que se encuentran almacenadas en una estructura especial del sistema, que es consultada por el DBMS cada vez que se realice una actualización del sistema.

### 2.2.9. Objetivos de las bases de datos

Los sistemas de base de datos se diseñan para manejar grandes cantidades de información, la manipulación de los datos involucra tanto la definición de estructuras para el almacenamiento de la información, como la provisión de mecanismos para la manipulación de la información, además, un sistema de base de datos debe tener implementados mecanismos de seguridad que garanticen la integridad de la información, a pesar de caídas del sistema o intentos de accesos no autorizados.

Un objetivo principal de un sistema de base de datos es proporcionar a los usuarios finales una visión abstracta de los datos, esto se logra escondiendo ciertos detalles de cómo se almacenan y mantienen los datos.

Los objetivos principales de un sistema de base de datos es disminuir los siguientes aspectos:

- ❖ Redundancia e inconsistencia de datos. Puesto que los archivos que mantienen almacenada la información son creados por diferentes tipos de programas de aplicación, existe la posibilidad de que si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información; es decir, que la misma información sea guardada más de una vez en un dispositivo de almacenamiento. Esto aumenta los costos de almacenamiento y acceso a los datos, además de que puede originar la inconsistencia de los datos; es decir, diversas copias de un mismo dato no concuerdan entre si, por ejemplo: que se actualiza la dirección de un cliente en un archivo y que en otros archivos permanezca la anterior.
- ❖ Dificultad para tener acceso a los datos. Un sistema de base de datos debe contemplar un entorno de datos que le facilite al usuario el manejo de los mismos. Supóngase un banco, y que uno de los gerentes necesita averiguar los nombres de todos los clientes que viven dentro del código postal 09700 de la ciudad. El gerente pide al departamento de procesamiento de datos que genere la lista correspondiente. Puesto que esta situación no fue prevista en el diseño del sistema, no existe alguna aplicación de consulta que permita este tipo de solicitud, lo que ocasiona una deficiencia del sistema.



- ❖ Aislamiento de los datos. Puesto que los datos están repartidos en varios archivos, y estos no pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los datos apropiados.
- ❖ Anomalías del acceso concurrente. Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los datos simultáneamente. En un entorno así, la interacción de actualizaciones concurrentes pueda dar por resultado datos inconsistentes. Para prevenir esta posibilidad debe mantenerse alguna forma de supervisión en el sistema.
- ❖ Problemas de seguridad. La información de toda empresa es importante, aunque unos datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de base de datos sea confiable debe mantener un grado de seguridad que garantice la autenticación y protección de los datos.
- ❖ Problemas de integridad. Los valores de datos almacenados en la base de datos deben satisfacer cierto tipo de restricciones de consistencia. Estas restricciones se hacen cumplir en el sistema añadiendo códigos apropiados en los diversos programas de aplicación.

### **2.2.10. Abstracción de la información**

Un objetivo importante de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo para que el sistema sea manejable, los datos se deben extraer eficientemente. Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema; interno, conceptual y externo, específicamente, el de almacenamiento físico; el del usuario y el del programador.

#### **2.2.10.1. Nivel físico**

Es la representación del nivel más bajo de abstracción, en éste se describe en detalle la forma en que se almacenan los datos en los dispositivos de almacenamiento (por ejemplo, mediante señales o índices para el acceso aleatorio a los datos).

#### **2.2.10.2. Nivel conceptual**

El nivel más alto de abstracción, describe que 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. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos. Consta de las siguientes definiciones:

- ❖ Definición de los datos. Se describe el tipo de datos y la longitud de campo, todos los elementos direccionables en la base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos y registros conceptuales (entidades).
- ❖ Relaciones entre datos. Se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

En el nivel conceptual la base de datos aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario.

#### **2.2.10.3. Nivel de visión**

Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe sólo una parte de la base de datos al usuario acreditado para verla. El

sistema puede proporcionar muchas visiones para la misma base de datos. La interrelación entre estos tres niveles de abstracción se ilustra en la siguiente figura:

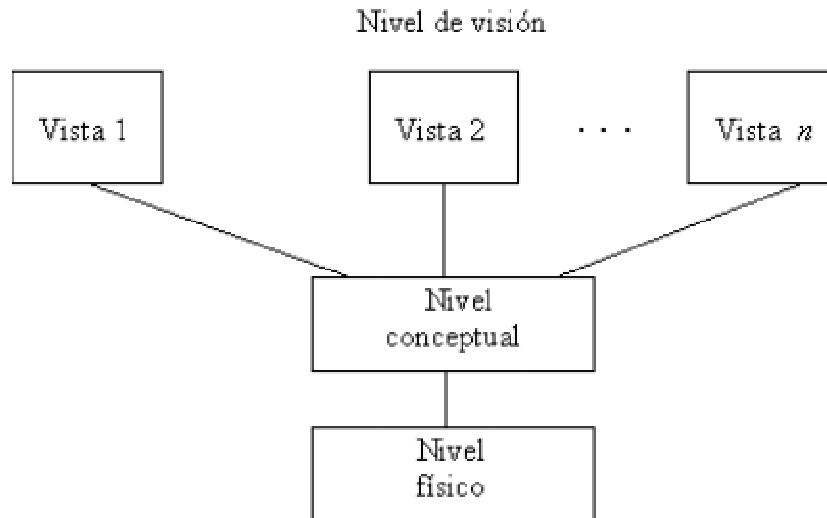


Figura 2.4. Niveles de visión.

### 2.2.11. Modelos de datos

Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Los modelos de datos se dividen en tres grupos:

- ❖ Modelos lógicos basados en objetos.
- ❖ Modelos lógicos basados en registros.
- ❖ Modelos físicos de datos.

#### 2.2.11.1. Modelos lógicos basados en objetos

Se usan para describir datos en los niveles conceptual y de visión; es decir, con este modelo representamos los datos de tal forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad – Relación.

##### 2.2.11.1.1. Modelo Entidad - Relación

Denominado por sus siglas como: E-R. Este modelo representa la realidad a través de entidades, que son objetos que existen y que se distinguen de otros por sus características. Las entidades pueden ser de dos tipos:

- ❖ Tangibles. Son todos aquellos objetos físicos que se pueden ver, tocar o sentir.
- ❖ Intangibles. Todos aquellos eventos u objetos conceptuales que no se pueden ver, aún sabiendo que existen.

Las características de las entidades en base de datos se llaman atributos; por ejemplo, el número de cuenta de un alumno o la dirección. A su vez una entidad se puede asociar o relacionar con más entidades a través de relaciones.

Los modelos Entidad – Relación, se representan gráficamente utilizando los siguientes símbolos presentados en la tabla:





Símbolo	Representa
	Entidad
	Relación
	Atributos
	Ligas

Tabla 2.2. Símbolos del modelo E-R.

En la siguiente figura se presenta un ejemplo de un modelo Entidad - Relación.

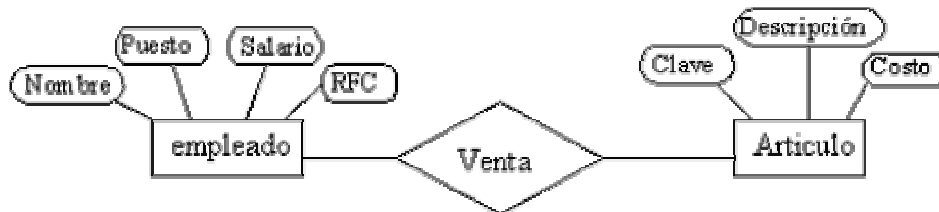


Figura 2.5. Ejemplo del modelo E-R.

### 2.2.11.2. Modelos lógicos basados en registros

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

Los tres modelos de datos más ampliamente aceptados son:

- ❖ Modelo relacional.
- ❖ Modelo de red.
- ❖ Modelo jerárquico.

#### 2.2.11.2.1. Modelo relacional

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en la cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

Un ejemplo sería como se muestra en la siguiente figura:

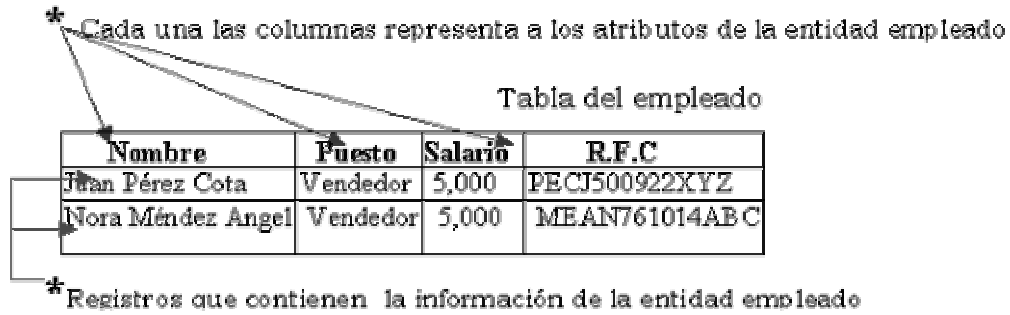


Figura 2.6. Ejemplo del modelo relacional.

Existen dos formas de representar las relaciones entre entidades, pero para esto hay que definir el concepto de llave primaria.

Una llave primaria es un atributo, el cual se define como atributo principal, es una forma única de identificar a una entidad. Por ejemplo, la CURP de una persona se distingue de otra, ya que la CURP no puede ser igual. Ahora las dos formas de representar las relaciones son:

- ❖ Haciendo una tabla que contenga cada una de las llaves primarias de las entidades involucradas en la relación.
- ❖ Incluyendo en alguna de las tablas de las entidades involucradas, la llave de la otra tabla.

### 2.2.11.2.2. Modelo de red

Este modelo representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias. En la figura 2.7 se muestra un ejemplo:

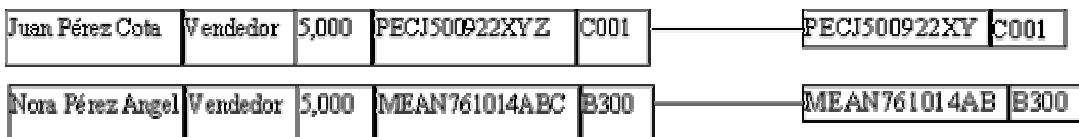


Figura 2.7. Ejemplo del modelo de red.

### 2.2.11.2.3. Modelo jerárquico

Es similar al modelo de red en cuanto a las relaciones y datos, ya que estos se representan por medio de registros y sus ligas. La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias. Se puede ver un ejemplo en la figura:

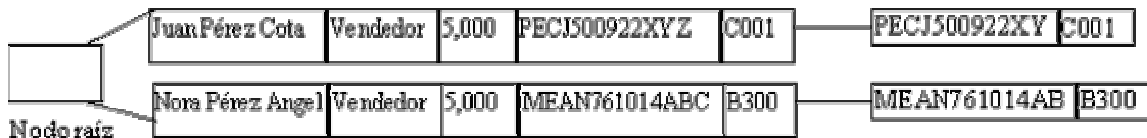


Figura 2.8. Ejemplo del modelo jerárquico.

### 2.2.11.3. Modelos físicos de datos

Se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de bases de datos. Existen dos clasificaciones de este tipo, que son:

- ❖ Modelo unificador.
- ❖ Memoria de elementos.

**2.2.12. Modelo de datos relacional**

A finales de los sesenta, Codd propone un modelo de datos basado en la teoría de las relaciones, en donde los datos se estructuran lógicamente en forma de relaciones –tablas-, siendo un objetivo fundamental del modelo mantener la independencia de esta estructura lógica respecto al modo de almacenamiento y a otras características de tipo físico. El modelo tenía los siguientes objetivos:

- ❖ Independencia lógica. Esto es, que el añadir, eliminar o modificar objetos de la base de datos no repercuta en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).
- ❖ Independencia física. Es decir, el modo en el que se almacenan los datos no influya en su manipulación lógica, y por tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico.
- ❖ Flexibilidad. En el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera.
- ❖ Uniformidad. Las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
- ❖ Sencillez. Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil de comprender y de utilizar por parte del usuario final.

Para conseguir los objetivos citados, Codd introduce el concepto de “relación” (tabla) como una estructura básica del modelo. Todos los datos de la base se representan en forma de relaciones cuyo contenido varía en el tiempo. La relación es el elemento básico en el modelo relacional y se puede representar como una tabla:

Atributo 1	Atributo 2	.....	Atributo n
Xxx	xxx	xxx	xxx
Xxx	xxx	xxx	xxx
Xxx	xxx	xxx	xxx

Tupla 1  
:  
Tupla n

*Figura 2.9. Modelo de datos relacional.*

En ella podemos distinguir un conjunto de columnas, denominadas *atributos*, que representan propiedades de la misma y que están caracterizadas por un nombre; y un conjunto de filas llamadas *tuplas* que son las ocurrencias de la relación. Existen también unos *dominios* donde los atributos toman sus valores. El número de filas de una relación se denomina *cardinalidad* de la relación y el número de columnas es el *grado* de la relación. Una relación se puede representar en forma de tabla, pero va a tener una serie de elementos característicos:

- ❖ No puede haber filas duplicadas, es decir, todas las tuplas tienen que ser distintas.
- ❖ El orden de las filas es irrelevante.
- ❖ La tabla es plana, es decir, en el cruce de una fila y una columna sólo puede haber un valor (no se admiten atributos multivaluados).

Un *dominio* D es un conjunto finito de valores homogéneos y atómicos caracterizados por un nombre; se dice homogéneo porque son todos del mismo tipo y atómicos porque son indivisibles. Todo dominio ha de tener un nombre por el cual se puede referir y un tipo de datos.

Un *atributo* es el papel que tiene un determinado dominio en una relación. Es muy usual dar el mismo nombre al atributo y al dominio. En el caso de que sean varios los atributos de una misma tabla definidos sobre el mismo dominio, habrá que darles nombres distintos, ya que una tabla no puede tener dos atributos con el mismo nombre.

Una *relación* se puede definir; matemáticamente, como un subconjunto del producto cartesiano de una lista de dominios, donde cada elemento de la relación, tupla, es una serie de n valores ordenados.

Están compuestas por dos partes; una cabecera y un cuerpo, se define la relación sobre un conjunto de dominio, la cabecera esta formada por un conjunto fijo de atributos tal que cada atributo A(j) corresponde a cada uno de los dominios de J, con j variando hasta n. El cuerpo está formado por un conjunto de tuplas, el cual varía con el tiempo, cada tupla está formada por un conjunto de pares (atributo, valor); donde (i) varía de 1 hasta n; y n es el número de tuplas del conjunto (A1, V1)(A2, V2)(A3, V3).....(An, Vn), n va a ser el grado de esa relación.

Una *llave candidata* de una relación; es un conjunto no vacío de atributos que identifican unívoca y mínimamente cada tupla. Por la propia definición de relación, siempre hay al menos una llave candidata, ya que al ser la relación un conjunto, no existen tuplas repetidas y por tanto; el conjunto de todos los atributos identificará unívocamente a las tuplas. Una relación puede tener más de una llave candidata, entre las cuales se debe distinguir:

- ❖ *Llave primaria.* Es aquella llave candidata que el usuario escogerá, por consideraciones ajenas al modelo relacional, para identificar a las tuplas de una relación.
- ❖ *Llave alternativa.* Son aquellas llaves candidatas que no han sido elegidas.

Se denomina llave *ajena o foránea* de una relación R2 a un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la llave primaria de otra relación R1. La llave foránea y la correspondiente llave primaria han de estar definidas sobre los mismos dominios.

Además de definir las llaves foráneas, hay que determinar las consecuencias que pueden tener ciertas operaciones (borrado y modificación) realizadas sobre tuplas de la relación referenciada; pudiéndose distinguir, en principio, las siguientes opciones:

- ❖ *Operación restringida.* Esto es; el borrado o la modificación de tuplas de la relación que contiene la llave primaria referenciada; sólo se permite si no existen tuplas con dicha llave en la relación que contiene la llave foránea.
- ❖ *Operación con transmisión en cascada.* Esto es, el borrado o la modificación de tuplas de la relación que contiene la llave primaria referenciada, lleva consigo el borrado o modificación en cascada de las tuplas de la relación que contiene la llave foránea.
- ❖ *Operación con puesta a nulos.* El borrado o la modificación de tuplas de la relación que contiene la llave primaria referenciada; lleva consigo poner a nulos los valores de las llaves foráneas de la relación que referencia. Esta opción, obviamente, sólo es posible cuando el atributo que es llave foránea admite el valor nulo.
- ❖ *Operación con puesta a valor por defecto.* El borrado o la modificación de tuplas de la relación que contiene la llave primaria referenciada; lleva consigo poner el valor por defecto a la llave foránea de la relación que referencia.

### 2.2.12.1. Normalización de bases de datos

Es un conjunto de reglas; que sirven para ayudar a los diseñadores 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 trataba de manipular los datos. Existen básicamente tres niveles de normalización:

- ❖ Primera forma normal. Establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.
- ❖ Segunda forma normal. Todas las dependencias parciales, se deben de 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.
- ❖ Tercera forma normal. Una tabla está normalizada en esta forma; sí todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas.

### 2.3. Red Hat Linux

En los siguientes puntos se mencionaran aspectos importantes de las herramientas utilizadas en la etapa de programación. En la figura se muestra el esquema de operación:

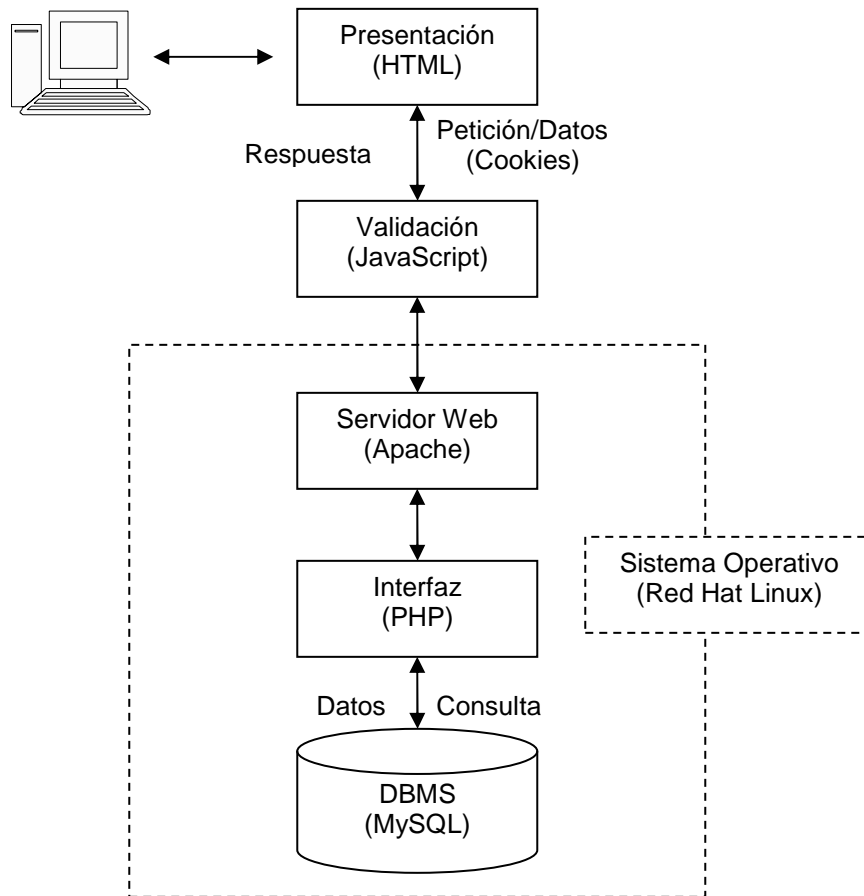


Figura 2.10. Modelo de datos relacional.

Red Hat Linux (sistema operativo) es una implementación de libre distribución Unix para computadoras personales, servidores y estaciones de trabajo. Fue desarrollado para el i386 y ahora soporta los procesadores i486, Pentium®, Pentium Pro y Pentium II, así como los clones AMD y Cyrix. También soporta máquinas basadas en SPARC, DEC Alpha, PowerPC/PowerMac, entre otras.

### 2.3.1. Historia

Linux hace su aparición a principios de la década de los noventa, en el año 1991. Linus Torvalds empezó a programar las primeras líneas de código de este sistema operativo. El comienzo estuvo inspirado en Minix, un pequeño sistema Unix desarrollado por Andy Tanenbaum. Las primeras discusiones sobre Linux fueron en el grupo de noticias comp.os.minix, en estas discusiones se hablaba sobre todo del desarrollo de un pequeño sistema Unix para usuarios de Minix que querían más.

Linus nunca anuncio la versión 0.01 de Linux (agosto 1991), esta versión no era ni siquiera ejecutable, solamente incluía los principios del núcleo del sistema, estaba escrita en lenguaje ensamblador y asumía que uno tenía acceso a un sistema Minix para su compilación.

El 5 de octubre de 1991, Linus anunció la primera versión oficial de Linux. Con esta versión se podía ejecutar Bash (*GNU Bourne Again Shell*) y gcc (el compilador GNU de C) pero no mucho más funcionaba. En este estado de desarrollo ni se pensaba en los términos soporte, documentación y distribución. Después de la versión 0.03, Linus salto en la numeración hasta la 0.10, más y más programadores a lo largo y ancho de Internet empezaron a trabajar en el proyecto y después de sucesivas revisiones, Linus incrementó el número de versión hasta la 0.95 (Marzo 1992). Más de un año después el núcleo del sistema estaba en la versión 0.99 y la versión 1.0 no llego hasta el 14 de marzo de 1994. Desde entonces no se ha parado de desarrollar.

Red Hat Software Inc. fue fundada en 1994 por Bob Young y Marc Ewing. En septiembre del 2003, Red Hat decidió concentrar sus esfuerzos de desarrollo en la versión corporativa de su distribución y delegó la versión común a Fedora Core, un proyecto abierto independiente de Red Hat.

Red Hat Linux es una distribución Linux, creada por la compañía Red Hat, que fue una de las más populares en los entornos de usuarios domésticos. La versión 1.0 fue presentada el 3 de noviembre de 1994. No es tan antigua como la distribución Slackware, pero ciertamente es más antigua que muchas otras. Fue la primera distribución que usó RPM como su formato de paquete y de cierta forma ha servido como el punto de partida para varias otras distribuciones, tales como la orientada hacia computadoras de escritorio Mandrake Linux, Yellow Dog, la cual se inicio desde Red Hat Linux, con soporte para PowerPC y ASPLinux.

Desde el 2003, Red Hat ha desplazado su enfoque hacia el mercado de los negocios con la distribución Red Hat Enterprise Linux. Red Hat Linux 9, la versión final, llegó oficialmente al final de su vida útil el pasado 30 de abril del 2004, aunque el proyecto Fedora Legacy continúa publicando actualizaciones.

### 2.3.2. Características

Red Hat Linux es instalado con un ambiente gráfico llamado *Anaconda*, diseñado para su fácil uso, por novatos. También incorpora una herramienta llamada *Lokkit* para configurar las capacidades del Firewall.

Con entornos de escritorio, ofrece una interfaz gráfica alternativa a la tradicional interfaz de línea de comandos de Unix. La creciente popularidad se debe a las ventajas que presenta ante otros tipos de software. Entre otras razones se debe a su estabilidad, al acceso a las fuentes, la independencia del proveedor, la seguridad, la rapidez con que incorpora los nuevos adelantos tecnológicos, la escalabilidad, la activa comunidad de desarrollo que hay a su alrededor, su interoperabilidad y a la abundancia de documentación relativa a los procedimientos. La colección de utilidades para la programación de GNU, es con diferencia la familia de compiladores más utilizada en Linux. Tiene capacidad para compilar C, C++, Java, Ada, entre muchos otros lenguajes.



Hay varios IDEs disponibles para Red Hat Linux incluyendo, Anjuta, KDevelop, NetBeans IDE y Eclipse. Además existen editores extensibles como puede ser Emacs. También dispone de capacidades para lenguajes de guión (*script*), aparte de los clásicos lenguajes de programación de shell, como es Python, Perl, PHP y Ruby.

Agregando que es multitarea, multiusuario, multiplataforma y multiprocesador; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; usa un sistema de memoria virtual por páginas; permite usar bibliotecas enlazadas tanto estática como dinámicamente; tiene un sistema de archivos avanzado pero puede usar los de los otros sistemas y soporta redes tanto en TCP/IP como en otros protocolos.

## 2.4. MySQL

MySQL es un sistema manejador de base de datos, creado bajo licencia GPL de la GNU. Su diseño multihilo, le permite soportar una gran carga de trabajo de forma muy eficiente. MySQL fue creado por la empresa sueca MySQL AB, que mantiene los derechos de autor del código fuente del servidor SQL (*Structured Query Language*), así como también de la marca.

Este manejador de base de datos es probablemente, el manejador más utilizado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de bibliotecas y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación; además de su fácil instalación y configuración.

### 2.4.1. Historia

El grupo comenzó usando mSQL para conectar sus tablas así como sus propias rutinas. Sin embargo; después de algunas pruebas, llegaron a la conclusión de que mSQL no cubría las necesidades en cuanto a velocidad o flexibilidad. Esto resulto en una nueva interfase SQL para su base de datos, pero casi con la misma interfase API (*Application Programming Interfaces*) de mSQL. Esta API fue diseñada para permitir el código a un tercero, que fue escrita para usar con mSQL, para ser portado fácilmente para usar con MySQL. La derivación del nombre MySQL no esta claro. El directorio base y un gran número de bibliotecas y herramientas han tenido el prefijo "my" hace como 10 años.

El logo de MySQL es un delfín llamado: *Sakila*. Sakila fue elegido por los fundadores de MySQL AB de una gran lista de nombres sugeridos por usuarios en el concurso "Nombre del Delfín". El nombre ganador fue enviado por Ambrose Twebaze, un desarrollador de software de código abierto (*open source*) de Swaziland, África. De acuerdo con Ambrose, el nombre Sakila tiene su raíz en SiSwati, la lengua local de Swaziland. Sakila es también el nombre de un pueblo en Arusha, Tanzania, cerca de la ciudad de origen de Ambrose, Uganda.

Uno de los más rápidos servidores de bases de datos SQL (*Structured Query Language*) actualmente en el mercado es el servidor MySQL, desarrollado por T.c.X. DataKonsultAB (*MySQL AB*). MySQL, disponible para descargar en <http://www.mysql.com>, ofrece al programador de bases de datos una serie de opciones y capacidades raramente vistas en otros servidores de bases de datos. Además, MySQL es gratuito para aquellos que deseen utilizarlo para uso privado y/o comercial.

Este manejador de base de datos ha obtenido enorme popularidad dentro de pequeñas y grandes empresas, incluyendo compañías como: Silicon Graphics y Siemens.

### 2.4.2. Características

Algunas características son las siguientes:

- ❖ Es un sistema manejador de base de datos relacional.
- ❖ Escrito en lenguaje C y C++.

- ❖ Trabaja en diferentes plataformas.
- ❖ Es software libre o gratuito (*open source*).
- ❖ Su principal objetivo de diseño fue la velocidad. Además de ser confiable y fácil de usar.
- ❖ Capacidad para manejar un número ilimitado de usuarios simultáneamente.
- ❖ Capacidad para manipular más de 50,000,000 registros.
- ❖ Mayor rendimiento y mayor velocidad; tanto al conectar con el servidor, como en la ejecución de comandos, quizá el más rápido que se encuentra en el mercado.
- ❖ Un sistema de privilegios y contraseñas muy flexible y seguro. Con esto se tiene un mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos. Las contraseñas son encriptadas cuando el usuario se conecta al servidor.
- ❖ Mejores utilidades de administración (*Backup, recuperación de errores, etc.*).
- ❖ Aunque se tarde en procesar algunas peticiones, no suele perder información ni corromper los datos.
- ❖ Mejor integración con PHP.
- ❖ No hay límites en el tamaño de los registros.
- ❖ Otra característica importante, es que consume muy pocos recursos, tanto de CPU como de memoria.

### 2.4.3. Instalación

Primero hay que descargar en el servidor el software, ya sea del sitio oficial, antes mencionado o de algún sitio espejo (*mirror*). Las versiones de Unix precompiladas y las de Windows, son tan simple como descomprimir y ocupar, se hace de la siguiente manera:

```
$gunzip < mysql-xxxx.tar.gz | tar -xvf
```

Donde xxxx es el número de la versión. Esto crea un directorio llamado mysql-xxxx, el cual contiene el código fuente. Después hay que cambiarse al directorio:

```
$cd mysql-xxxx
```

Aquí hay varios archivos como el README e INSTALL, es muy importante que se lean para poder proseguir con la instalación. MySQL tiene un script de configuración, para ejecutarlo es con la siguiente expresión:

```
./configure
```

Si se necesita especificar algún parámetro, por ejemplo, que el servidor tiene poca memoria, o se desea instalar en otra ruta; se escribe lo siguiente:

```
./configure --help
```

Proporciona una lista de opciones; por ejemplo, para el caso de que el servidor tiene poca memoria, se especifica con `–with-low-memory`, o para la ruta `–prefix=/usr/local/mysql`. El script de configuración puede correr e inspeccionar el sistema, y entonces crear los archivos necesarios para la instalación. Si algo falla, manda un mensaje de error en el cual menciona el porqué. Después se escribe en la línea de comandos lo siguiente:

```
$make
```

MySQL es un programa complejo y toma algún tiempo compilarlo. Después, en la línea de comandos se escribe lo siguiente y los archivos quedaran instalados:

```
$make install
```

Ahora para poder trabajar con MySQL, se necesita iniciar el servidor con el siguiente script `mysql.server.start` y para detenerlo con `mysql.server.stop`. Otra forma de inicializar el servidor manualmente, es entrando al directorio raíz en donde se instaló (`/usr/local/mysql`) y se escribe `./bin/safe_mysql &`.

Desde Linux se abre una terminal y en la línea de comandos se escribe lo siguiente:

```
$mysql –help
```

Al ejecutarlo muestra las opciones proporcionadas por `mysql`. Para conectarse al servidor, usualmente se necesita de un nombre de usuario (`login`) y de una contraseña (`password`), y si el servidor al que se desea conectar está en una máquina diferente de la local, también se necesita indicar el nombre o la dirección IP de dicho servidor. Una vez que se conocen estos tres valores, se puede conectar de la siguiente manera:

```
$mysql –h NombreDelServidor –u NombredeUsuario –p
```

Cuando se procesa este comando, se solicitará la contraseña para el nombre de usuario que se esta conectando. Si la conexión al servidor MySQL se estableció de manera satisfactoria, envía un mensaje de bienvenida y mostrará el prompt de `mysql`:

```
$mysql -h localhost -u root -p
Enter password: *****

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5563 to server version: 3.23.58

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Este prompt indica que `mysql` está listo para recibir comandos. Algunas instalaciones permiten que los usuarios se conecten de manera anónima al servidor corriendo en la máquina local. Para tal caso, se debe ser capaz de conectarse al servidor invocando a `mysql` sin ninguna opción:

```
$mysql
```

Después de que se conectó de manera satisfactoria; para desconectarse hay tres formas de hacerlo, con “quit”, “exit” o con “ctrl+d”.

#### 2.4.4. Tipos de datos en MySQL

MySQL maneja diferentes tipos de datos, algunos de los más comúnmente usados son los siguientes:

- ❖ CHAR(#). Un tipo de dato *char* es una cadena o secuencia de caracteres que puede estar en el rango de 1-255 caracteres. Por ejemplo:

```
nombre_variable CHAR(10);
```

- ❖ VARCHAR(#). Es una forma más flexible del tipo de dato *char*. También representa datos del tipo de cadenas o secuencias de caracteres. *Varchar* puede almacenar de 1-255 caracteres. Aunque el tipo *char* es mucho más rápido que un *varchar*, algunas veces hasta en un 50%. Un tipo *char* almacena la longitud total de la variable declarada, sin importar el tamaño de los datos contenidos, mientras que un *varchar* almacena solamente la longitud de los datos, así se reduce el tamaño de la base de datos. Por ejemplo:

```
nombre_variable VARCHAR(10);
```

- ❖ INT [unsigned]. El tipo de dato *int* almacena enteros en un rango de -2147483648 a 2147483647. La opción *unsigned* puede o no ser colocada en la declaración, modificando el rango de 0 a 4294967295. Por ejemplo:

```
nombre_variable INT;
Entero valido: -24567 Entero no valido: 3000000000
```

```
nombre_variable INT unsigned;
Entero valido: 3000000000 Entero no valido: -24567
```

- ❖ FLOAT[(#,#)]. Un tipo de dato *float* representa números decimales pequeños, utilizado cuando se necesita almacenar un dato más preciso. Por ejemplo:

```
nombre_variable FLOAT(4,2)
Indica que puede almacenar cuatro caracteres y dos lugares decimales.
```

- ❖ DATE. Almacena información relacionada con fechas. El formato por omisión es “YYYY-MM-DD”; y rangos de “0000-00-00” a “9999-12-31”. MySQL tiene diferentes formatos para el tipo de dato *date*. Por ejemplo:

```
nombre_variable DATE;
```

#### 2.4.5. Sintaxis y comandos

Hasta el apartado anterior ya se conoce la forma de conectarse al servidor MySQL y los tipos de datos, aún cuando no se ha seleccionado alguna base de datos para trabajar. Lo que se hará a continuación es describir algunos comandos para conocer algunas características.

```
mysql> SELECT VERSION(), CURRENT_DATE;
+-----+-----+
| VERSION() | CURRENT_DATE |
+-----+-----+
| 3.23.58 | 2005-10-01 |
+-----+-----+
1 row in set (0.03 sec)

mysql>
```

Este comando ilustra distintos aspectos acerca de mysql:

- ❖ Un comando normalmente consiste de una sentencia SQL seguida por un punto y coma.
- ❖ Cuando se emite un comando, mysql lo manda al servidor para que lo ejecute, muestra los resultados y regresa el prompt indicando que está listo para recibir más consultas.
- ❖ MySQL muestra los resultados de la consulta como una tabla (filas y columnas). La primera fila contiene etiquetas para las columnas. Las filas siguientes muestran los resultados de la consulta. Normalmente las etiquetas de las columnas, son los nombres de los campos de las tablas que se están usando en alguna consulta. Si lo que se está recuperando es el valor de una expresión (como en el ejemplo anterior) las etiquetas en las columnas son la expresión en sí.
- ❖ MySQL muestra cuántas filas fueron regresadas y cuanto tiempo tardo en ejecutarse la consulta, lo cual puede proporcionarnos una idea de la eficiencia del servidor, aunque estos valores pueden ser un tanto imprecisos ya que no se muestra la hora del CPU y porque pueden verse afectados por otros factores, tales como la carga del servidor y la velocidad de comunicación en una red.
- ❖ Las palabras clave pueden ser escritas usando mayúsculas y minúsculas.

Aunque hasta este momento se han escrito sentencias sencillas de una sola línea, es posible escribir más de una sentencia por línea, siempre y cuando estén separadas por punto y coma. Un comando no necesita ser escrito en una sola línea, así que los comandos que requieran de varias líneas no son un problema. MySQL determinará en donde finaliza la sentencia cuando encuentre el punto y coma, no cuando encuentre el fin de línea.

Los comandos multilinea comúnmente ocurren por accidente cuando se tecléa ENTER, pero se olvida escribir el punto y coma. En este caso mysql se queda esperando para que se finalice la consulta:

```
mysql> SELECT USER()
->
```

Sí esto llega a suceder, muy probablemente mysql estará esperando por un punto y coma, de manera que si se escribe el punto y coma se podrá completar la consulta y mysql podrá ejecutarla:

```
mysql> SELECT USER()
-> ;
+-----+
| USER() |
+-----+
| root@localhost |
+-----+
1 row in set (0.00 sec)
```

Para saber qué bases de datos existen en el servidor, se utiliza el comando SHOW, por ejemplo:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| mysql   |
| test    |
+-----+
2 rows in set (0.00 sec)
```

Es probable que la lista de bases de datos que se muestre sea diferente, pero seguramente las bases de datos “mysql” y “test” estarán entre ellas. En particular, la base de datos “mysql” es requerida, ya que ésta tiene la información de los privilegios de los usuarios de MySQL. La base de datos “test” es creada durante la instalación de MySQL con el propósito de servir como área de trabajo para los usuarios que inician en el aprendizaje de MySQL.

El comando USE sirve para cambiarse de base de datos, en el siguiente caso “test”:

```
mysql> USE test
Database changed
mysql>
```

Para asignar permisos a los usuarios de la base de datos así como para crear usuarios, se utiliza el siguiente comando:

```
mysql> GRANT ALL on BaseDatos.* TO NombreUsuario@MiComputadora
-> IDENTIFIED BY 'Contraseña';
```

En donde:

- ❖ BaseDatos. Es el nombre de la base de datos que utilizará.
- ❖ NombreUsuario. Es el nombre de usuario asignado dentro del contexto de MySQL.
- ❖ Computadora. Es el nombre o la dirección IP de la computadora desde la que se conecta al servidor MySQL (opcional).
- ❖ Contraseña. Es la palabra clave que se ha asignado, igualmente, dentro del ambiente de MySQL exclusivamente.

El nombre de usuario y la contraseña, no tienen nada que ver con el nombre de usuario y contraseña manejados por el sistema operativo (si es el caso).

Si la base de datos ya fue creada al momento de asignar permisos, se puede hacer uso de ella. De otro modo, se puede crear con la siguiente instrucción:

```
mysql> CREATE DATABASE NombreBaseDatos;
Query OK, 1 row affected (0.00 sec)
```

En Unix, los nombres de las bases de datos son sensibles al uso de mayúsculas y minúsculas, esto también es cierto para el nombre de las tablas. La base de datos se debe de seleccionar cada vez que se inicia sesión en mysql. Por ello es recomendable que se indique la base de datos sobre la que se va a trabajar al momento de iniciar sesión en mysql. Por ejemplo:

```
$mysql -h Servidor -u NombreUsuario -p BaseDatos

Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17 to server version: 3.23.58

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql>
```

Cabe aclarar que “BaseDatos” no es la contraseña que se está proporcionando desde la línea de comandos, sino el nombre de la base de datos a la que se desea acceder. Si se quiere proporcionar la contraseña en la línea de comandos después de la opción “-p”, se debe hacer sin dejar espacios (por ejemplo, -pgato23). Sin embargo, escribir la contraseña desde la línea de comandos no es recomendable, ya que es bastante inseguro.

Antes de poder ejecutar comandos en la base de datos, se necesitan crear las tablas; para ello se ocupa la siguiente instrucción:

```
mysql>CREATE TABLE NombreTabla(
->NombreVariable1 TipoDato(#),
->NombreVariable2 TipoDato(#),
->.....
->.....
->NombreVariableN TipoDato(#));

Query OK, 0 rows affected (0.02 sec)
mysql>
```

El mensaje enviado indica que se ha creado satisfactoriamente la tabla. Cabe aclarar que dos tablas no pueden tener el mismo nombre. Cada renglón es referido como una columna. Las características de las columnas son las siguientes:

- ❖ Un nombre no puede estar formado solamente por números.
- ❖ Un nombre puede comenzar con un número.
- ❖ Un nombre puede estar formado hasta por 64 caracteres.

Las siguientes opciones pueden ser colocadas después de cualquier tipo de dato, agregándoles otras características y capacidades.

- ❖ Primary Key. Utilizado para diferenciar un registro de otro. Dos registros no pueden tener la misma Primary Key. Es muy útil porque puede ser lo único que diferencie un registro de otro.
- ❖ Foreign key. Utilizado para relacionar una o más tablas en una base de datos. Es decir, la primary key de una tabla pasa a otra convirtiéndose en foreign key.
- ❖ Auto\_Increment. Una columna con esta función es automáticamente incrementada un valor (+1) cuando una inserción es hecha dentro del registro.

- ❖ Null. Significa que el valor no es requerido para almacenar los datos en el registro. Es decir, se le puede asignar un valor nulo.
- ❖ Not Null. Significa que el valor es requerido para almacenar los datos en el registro. Es decir, no se le puede asignar un valor nulo.

La siguiente instrucción muestra todas las tablas existentes dentro de la base de datos:

```
mysql> SHOW TABLES;
```

El comando mostrado a continuación regresa la información de las columnas que pertenecen a la tabla especificada:

```
mysql> SHOW COLUMNS FROM NombreTabla;
```

Una base de datos puede ser manipulada en diferentes formas posibles:

- ❖ Insertar registro.

La inserción de registros dentro de una tabla se realiza con la siguiente instrucción:

```
mysql> INSERT INTO NombreTabla (Campo1,Campo2,Campo3,...,CampoN) VALUES
>('Texto1','Texto2',123456,.....,'TextoN');
```

Donde:

- NombreTabla. Es el nombre de la tabla donde se guardará la información.
- Campo#. Es el nombre de la variable.
- Texto#. Es la información específica de cada variable.

Hay que tomar en cuenta algunos aspectos importantes; las comillas simples son utilizadas cuando el tipo de dato es VARCHAR, así como con todos los datos del tipo STRING (CHAR, TEXT, BLOB, VARCHAR, etc.). Los datos del tipo INT o numérico no requieren estar entre comillas simples.

Es importante recordar que el mismo número de valores debe ser insertado, como los tipos de datos declarados en la creación de la tabla. Es decir, el mismo número de valores debe ser igual al número de variables.

Una de las ventajas de MySQL, es la habilidad para convertir entre tipo de datos sin problemas. MySQL automáticamente convierte entre enteros, cadenas y fechas sin problemas.

- ❖ Selección de registros.

Una base de datos no sería muy útil sino se pudiera buscar y extraer datos. En términos de MySQL, esto se realiza con la siguiente instrucción:

```
mysql> SELECT * FROM NombreTabla;
```

Cuando MySQL lo procesa, retornará todos los registros que estén almacenados en NombreTabla. Para poder restringir la selección o búsqueda; se puede utilizar la opción WHERE, la sintaxis es la siguiente:

```
mysql> SELECT * FROM NombreTabla WHERE campo = "condición";
```



Ahora los campos que mostrará, serán los que cumplan con la condición. MySQL cuenta con soporte para los operadores lógicos básicos, como los siguientes:

- AND ( && ). Es utilizado cuando se requiere extraer de la base de datos, uno o más registros, siempre y cuando cumpla con dos o más condiciones especificadas. La sintaxis es la siguiente:

```
mysql> SELECT * FROM NombreTabla WHERE campo1 = "condicion1" AND campo2 = "condicion2";
```

- OR ( || ). Es utilizado cuando se requiere extraer de la base de datos uno o más registros, el cual cumpla una sola condición de las especificadas. La sintaxis es la siguiente:

```
mysql> SELECT * FROM NombreTabla WHERE campo1 = "condicion1" OR campo2 = "condicion2";
```

- NOT ( ! ). Muestra todos los registros que no cumplan con la condición especificada. La sintaxis es la siguiente:

```
mysql> SELECT * FROM NombreTabla WHERE campo1 != "condicion1";
```

- ORDER BY. Despliega todos los registros; dependiendo si se le agrega alguna condición, ordenados respecto al campo especificado. La sintaxis es la siguiente:

```
mysql> SELECT * FROM NombreTabla ORDER BY campo;
```

Al realizar una selección, MySQL puede mostrar solo los campos que se requieran, por ejemplo:

```
mysql> SELECT campo1, campo2 FROM NombreTabla;
```

Así como también se puede hacer uso de los operadores lógicos y de la cláusula WHERE.

❖ Eliminar registros.

MySQL permite eliminar los registros insertados en una tabla, por medio del siguiente comando:

```
mysql> DELETE FROM NombreTabla WHERE campo = "condicion";
```

Elimina los registros que cumplan con la condición especificada. Si lo que se requiere es eliminar la tabla completa se lleva a cabo con la siguiente instrucción:

```
mysql> DROP TABLE NombreTabla;
```

❖ Modificar registros.

MySQL tiene también la capacidad para poder modificar los registros que ya han sido almacenados en la tabla. Esto se realiza por medio de la siguiente instrucción:

```
mysql> UPDATE NombreTabla SET campo1 = 'nuevos datos', campo2 = 'nuevos datos'
WHERE campo = "condicion";
```

Hasta el momento sólo se ha explicado con los datos que están almacenados en las tablas. Pero MySQL tiene una muy importante función para modificar tablas previamente creadas. Esto es realizado por la instrucción ALTER, esta función permite agregar, modificar y eliminar columnas; así como también renombrar las tablas. A continuación se muestran la sintaxis:

- Renombrar la tabla:

```
mysql> ALTER TABLE NombreTabla RENAME NuevoNombreTabla;
```

- Agregar una columna:

```
mysql> ALTER TABLE NombreTabla ADD NuevaColumna TipoDato(#);
```

- Modificar una columna:

```
mysql> ALTER TABLE NombreTabla CHANGE NombreColumna NuevoNombreColumna
TipoDato(#);
```

- Eliminar una columna:

```
mysql> ALTER TABLE NombreTabla DROP NombreColumna;
```

## 2.5. Apache

Apache es el servidor Web más utilizado en todo el mundo, ya que es muy potente, puede ser ejecutado en diferentes plataformas, es flexible, entre otras características que a continuación se describirán.

### 2.5.1. Historia

El servidor Web Apache surgió a partir del servidor de HTTP más famoso y difundido en su época: NCSA. Desde entonces se convirtió en un poderoso rival de todos los servidores Unix utilizados hasta la fecha por su eficiencia, funcionalidad y rapidez. Se desarrolla de forma estable y segura gracias a la cooperación y los esfuerzos de un grupo de personas conocidas como grupo Apache (*Apache Group*), los cuales se comunican a través de Internet y del Web. Juntos se dedican a perfeccionar el servidor y su documentación regidos por la ASF (*Apache Software Foundation*).

En la actualidad Apache es el servidor más utilizado en el mundo de acuerdo con las estadísticas de <http://www.netcraft.com/Survey/> que lo colocan en más de 7 millones de sitios Web, lo cual significa más del 60% en todo el mundo. Entre las características principales del Apache se encuentran:

- ❖ Es un servidor Web potente, flexible y ajustado al HTTP/1.1.
- ❖ Es altamente configurable y extensible.

- ❖ Puede ser ajustado a través de la definición de módulos empleando su propio API (*Application Programming Interface*).
- ❖ Provee todo su código fuente de forma libre y se distribuye bajo una licencia no restrictiva.
- ❖ Se ejecuta en diversas plataformas operativas tales como: Windows 9x/NT, Macintosh, Novell NetWare, OS/2, Linux y la mayoría de los Unix existentes: IRIX, Solaris, HPUX, SCO, FreeBSD, NetBSD, AIX, Digital Unix, etc.
- ❖ Se desarrolla de forma acelerada estimulando la retroalimentación desde sus usuarios a través de nuevas ideas, reportes de errores y parches.
- ❖ Apache significa “*A PAtCHy sErver*”, o sea se basa en un código y un conjunto de archivos “parches”. Otros desarrolladores relacionan su nombre con el de las tribus nativas americanas de Apaches.
- ❖ Implementa muchas posibilidades frecuentemente demandadas, tales como:
  - Bases de datos DBM para autenticación. Permiten establecer fácilmente la protección de documentos a través de contraseñas para una gran cantidad de usuarios sin dañar el funcionamiento del servidor.
  - Respuestas adaptables a los errores o problemas. Se pueden definir archivos o scripts de tipo CGI (*Common Gateway Interface*) que respondan ante la ocurrencia de errores internos o en las solicitudes realizadas.
  - Directiva para definir múltiples índices. Se utiliza cuando se solicitan directorios por parte de los clientes a partir de lo cual se puede buscar en estos y devolver un documento índice cuyo nombre puede ser por ejemplo: index.html, index.cgi o default.html.
  - Ilimitadas y flexibles posibilidades de redireccionamiento y definición de alias para los URL's. Apache no tiene un límite establecido para definir alias y redireccionamientos que pueden ser declarados en sus archivos de configuración.
  - Negociación del contenido de las respuestas. Apache es capaz de ofrecer la mejor representación de la información accedida de acuerdo con las capacidades del cliente solicitante.
  - Soporte de hosts virtuales. Es la habilidad del servidor de distinguir entre los pedidos hechos a diferentes direcciones IP o nombres de dominio definidos en la misma máquina.
  - Configuración flexible de las trazas generales. Es posible adaptar el formato de las trazas obtenidas así como redireccionarlas a través de tuberías (Unix) en aras de filtrarlas. De esta forma se puede lograr por ejemplo dividir dinámicamente las trazas de los hosts virtuales en distintos archivos.

El paquete de la distribución Red Hat que contiene la implementación del servidor Apache para Linux se nombra apache. También se dispone de un manual del mismo en el paquete apache-manual. Existe una aplicación con interfaz gráfica en el paquete apacheconf que permite configurar al Apache con las limitaciones propias de dichas interfaces.

El Apache en Red Hat, se ejecuta a través de un daemon llamado httpd que se manipula utilizando el script de inicio del mismo nombre en /etc/rc.d/init.d/. Por tanto la forma más sencilla de

iniciar, detener, conocer el estado o indicar que relea su configuración al daemon es como se muestra en los ejemplos:

```
# service httpd stop
Stopping httpd:          [ OK ]

# service httpd start
Starting httpd:         [ OK ]

# service httpd status
httpd ( pid 6973 6972 6971 6970 6969 6968 6967 6966) is running...

# service httpd reload
Reloading httpd:       [ OK ]
```

### 2.5.2. Configuración

Una vez instalado el paquete del Apache en Red Hat, el directorio de la configuración será `/etc/httpd/conf/`. Esto puede variarse utilizando la opción `-f` al ejecutar el daemon. Los archivos de configuración agrupados en este directorio están formados por un conjunto de directivas que regulan el comportamiento del servidor. Además se pueden incluir comentarios precedidos por el carácter “#” como es tradicional.

Actualmente el archivo de configuración principal del Apache y el único que se debe modificar de los existentes, se nombra `httpd.conf`. En versiones anteriores se empleaban otros dos archivos nombrados `srvc.conf` y `access.conf`. Apache procesa estos archivos en el orden en que se mencionaron. En las versiones actuales se recomienda escribir todas las directivas de configuración en `httpd.conf` y deshabilitar la lectura de los otros archivos en la propia configuración.

Además de los archivos de configuración mencionados, el acceso a cada directorio del servidor se puede regular en un archivo, contenido en el mismo directorio o en sus antecesores, cuyo nombre por defecto es `.htaccess`, y que también puede ser variado mediante la configuración.

Como ya se explicó, de los tres archivos de configuración mencionados el más importante es `httpd.conf`. Éste se divide en tres secciones con fines puramente organizativos:

Sección 1: reúne los aspectos globales del servidor. Por ejemplo: el número máximo de clientes concurrentes, los timeouts, el directorio raíz del servidor, etc.

Sección 2: agrupa las directivas que definen la forma de responder a todos los pedidos del servidor principal, o sea aquellos que no son para los hosts virtuales, de existir alguno definido. También reúne los aspectos por defecto de todos los hosts virtuales que se configuren más adelante.

Sección 3: agrupa las directivas relacionadas con los hosts virtuales que se definan.

## 2.6. PHP

PHP es un lenguaje interpretado de alto nivel, embebido en páginas HTML y ejecutado en el lado del servidor. Es muy flexible, ya que puede interactuar con diferentes bases de datos, entre las que destacan, MySQL y PostgreSQL.

### 2.6.1. Historia

PHP fue desarrollado en otoño de 1994 por Rasmus Lerdorf. Las versiones desarrolladas inicialmente fueron utilizadas únicamente en sus páginas web.

Pero en el año de 1995 fue puesta al público con el nombre de Herramientas para páginas web personales (Personal Home Pages Tools). Cabe mencionar que esta versión era muy limitada, apenas contenía algunas macros y una serie de utilidades comunes en las páginas web.

Durante el año de 1995 Rasmus incorporó a la versión denominada PHP/FI versión 2.FI un intérprete para formularios, así como soporte para mSQL. Posteriormente en el año de 1996 PHP se estaba usando en cerca de 15,000 páginas web en el mundo.

A mediados de 1997 el desarrollo del proyecto sufrió un profundo cambio, dejó de ser un proyecto personal de Rasmus, al cual habían ayudado un grupo de usuarios y se convirtió en un proyecto de grupo mucho más organizado. El analizador sintáctico se rescribió desde el principio por Zeev Suraski y Andi Gutmans y este nuevo analizador estableció las bases para PHP versión 3. Gran cantidad de código de PHP/FI fue portado a PHP3 y otra gran cantidad fue escrita completamente de nuevo.

PHP es un acrónimo de Preprocesador de Hipertexto, es decir, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. La mayoría de su sintaxis es similar a C, Java y Perl. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

Con PHP se puede procesar la información de formularios, generar páginas con contenidos dinámicos, mandar o recibir cookies, entre otras. Sin duda alguna la capacidad más importante es el soporte para una gran cantidad de bases de datos. El desarrollo de una interfaz para poder explotar todos los datos es una forma sencilla en PHP.

Entre algunas bases de datos que soporta el lenguaje PHP tenemos: Adabas, DBase, Empress, FilePro, InterBase, Oracle, IBM DB2, FrontBase, PostgreSQL, Informix, MSOL, Solid, Velocis, MySQL y Sybase

PHP también soporta el uso de otros servicios que utilicen protocolos como: IMAP, POP3, http.

PHP es un lenguaje que se ejecuta del lado del servidor. Algunos servidores son: Roxen, Apache, IIS (*Internet Information Server*), Zeus, TomCat y Xitami.

Los requerimientos para que se ejecuten los programas de PHP son los siguientes: Servidor web e intérprete PHP.

Las figuras 2.11 y 2.12 ayudan a entender mejor la diferencia entre lenguaje interpretado y compilado:

Lenguajes interpretados. Programa PHP

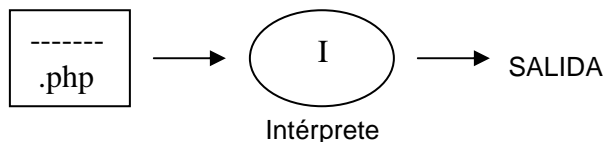


Figura 2.11. Representación del Intérprete.

Algunos lenguajes que necesitan intérprete: PHP, Perl, JavaScript.

Lenguajes compilados.

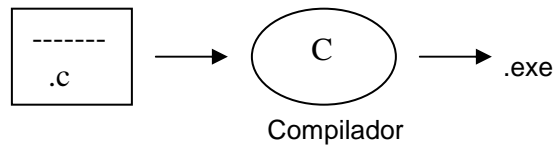


Figura 2.12. Representación del Compilador.

### 2.6.2. Sintaxis y estructura

El código de un script PHP es embebido dentro de una página HTML y es ejecutado en el servidor antes de ser enviado a un navegador. Para que el código embebido sea ejecutado, es necesario que se encuentre dentro de un archivo de texto plano cuyo nombre tenga la extensión php y además este encerrado entre un par de etiquetas especiales. Existen cuatro maneras en las cuales se puede incorporar el código PHP en una página HTML:

- 1.- `<? ...código de php... ?>`
- 2.- `<?php ...código de php... ?>`
- 3.- `<SCRIPT LANGUAGE="php"> ...código de php... </SCRIPT>`
- 4.- `<% ...código de php... %>`

PHP corresponde a las iniciales de *Personal Home Page*, Procesador de Hipertexto. PHP es un lenguaje de programación, con una sintaxis similar a los lenguajes C y Perl, que se interpreta por un servidor web Apache y genera código HTML dinámico. Es decir, nos permite crear un programa que se pueda ejecutar en el servidor desde un programa visualizador de páginas web y dar respuestas en función de los datos que introduzca el usuario. El cliente nunca verá el código del programa PHP, sólo le llegarán las páginas HTML que genere el programa. A diferencia de JavaScript, que se ejecuta en las máquinas clientes, un programa PHP se ejecuta en el servidor web.

En la figura 2.13 se puede apreciar el esquema de funcionamiento de un programa en PHP: el cliente realiza una petición de un programa a un servidor web como si se tratara de cualquier otra página; el cliente no sabrá distinguirlo. El servidor web, por la extensión de la página (.php por ejemplo) se la envía al intérprete PHP y éste, una vez ejecutado el programa, le devuelve los resultados al navegador cliente.

Para poder usar PHP necesitamos unos mínimos conocimientos de HTML. Aunque siempre existe la posibilidad de generar mediante un programa la página HTML (como por ejemplo Macromedia Dreamweaver) y después, mediante un editor de textos añadir el código PHP. A esta característica de PHP se le llama código embebido.

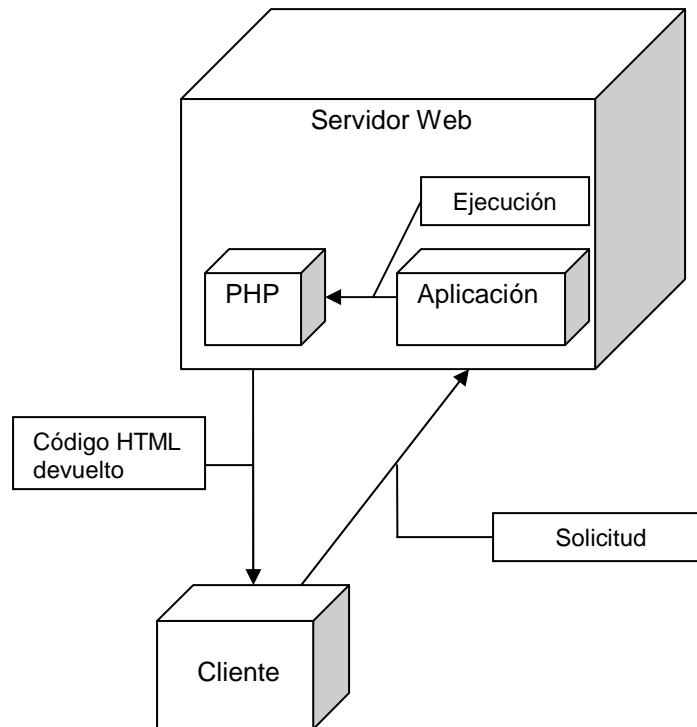


Figura 2.13. Esquema de funcionamiento de PHP.

### 2.6.3. Características

El lenguaje de programación PHP dispone de funciones para realizar las operaciones habituales de los lenguajes de programación, usar archivos, tratamiento de cadenas de texto, etc. Quizá, unas de las características más interesantes que incorpora, es la facilidad para consultar bases de datos y generar páginas en función de los resultados obtenidos en la correspondiente consulta. Las consultas se pueden realizar, bien en modo nativo o, bien, mediante ODBC. En modo nativo, es compatible con Oracle, PostgreSQL, Informix, MySQL y otras mencionadas anteriormente.

Además, en este lenguaje podemos definir clases y usar ciertas características de la programación orientada a objetos.

PHP presenta múltiples ventajas frente a otros lenguajes de programación que necesariamente harán que este lenguaje se imponga como una alternativa para el desarrollo de todo tipo de aplicaciones.

En primer lugar se ejecuta a través de una interfaz que le resulta familiar al usuario: el cliente web. No es necesario que el usuario aprenda nuevas combinaciones de teclas, ni nada parecido, para aprender a usar el programa.

Tampoco es necesario tener que instalar un software adicional en la estación cliente para usar un programa PHP aparte del propio navegador web.

De la misma forma, la ejecución de un programa PHP se puede realizar desde un cliente web de cualquier plataforma; el usuario puede escoger su sistema operativo y su cliente web preferidos.

El propio diseño de PHP lleva incorporada esta virtud. El programa se ejecuta en un servidor al cual se puede acceder desde cualquier nodo de una red. El servidor siempre podría limitar el

acceso a determinados nodos y además obligar a la autenticación de un usuario para poder acceder a ciertas partes de un programa.

Al tener el código ejecutable albergado en el servidor, este código está protegido tanto de la manipulación de los usuarios como de la presencia de virus. Cuando se configura el servidor se tiene que asegurar de que se han tomado las medidas de seguridad necesarias para impedir accesos indebidos al sistema.

El código incluido o embebido pretende, sobre todo, aclarar los mecanismos de programación, más que ser código eficiente y depurado. En muchas ocasiones se sacrifica la eficiencia del código para ganar en claridad didáctica. En otras ocasiones lo que se pretende es dar ideas en el código para facilitar la resolución de problemas habituales de programación. El inconveniente que presenta PHP es que la inclusión de los controles de HTML le resta legibilidad al código.

Hay múltiples razones para escoger PHP como lenguaje de programación para entornos web en comparación con otros existentes en el mercado. Algunas de ellas son las siguientes:

- ❖ Conociendo los lenguajes C, Perl, Java o programación en Shell de Unix puede decir que se tienen los fundamentos de PHP.
- ❖ Está ampliamente probado como herramienta. Un millón y medio de servidores, por el momento, en todo el mundo lo avalan como una plataforma para desarrollar aplicaciones de portales, comercio electrónico, aplicaciones en intranets, etc.
- ❖ Se puede usar prácticamente en cualquier plataforma.
- ❖ Aunque no dispone de soporte comercial, existen numerosas listas de correo en las que se pueden obtener soluciones a los problemas que se puedan presentar.
- ❖ Existen recursos en la web que pueden facilitar el desarrollo de estas aplicaciones. Existen bibliotecas de clases que resuelven los problemas más frecuentes con los que se puede encontrar el programador.
- ❖ No requiere unos recursos desmesurados para funcionar. Es posible tener PHP funcionando en una computadora portátil 486 con 16M de RAM para desarrollo.

Para poder utilizar los programas PHP evidentemente, lo primero que necesitamos es un servidor web Apache. Además, este servidor web Apache tiene que estar compilado para incorporar las características PHP o admitir módulos dinámicos. PHP funciona asociado al servidor web Apache y se distribuyen los programas fuentes en C. Así es fácil compilarlo en una u otra plataforma (bajo Win32 o bajo Unix/Linux) que disponga de un compilador de C realizando las modificaciones necesarias.

Respecto a las bases de datos, PHP es capaz de consultar una base de datos con una potencia excepcional a la hora de generar páginas web dinámicas. En la figura 2.14 podemos ver la relación que se puede establecer entre PHP y un sistema manejador de bases de datos: nuestro programa puede realizar consultas y obtener los resultados para procesarlos. Este manejador de bases de datos puede estar ubicado en la misma máquina que ejecuta PHP o puede estar en otra máquina distinta y realizar las consultas a través de la red.



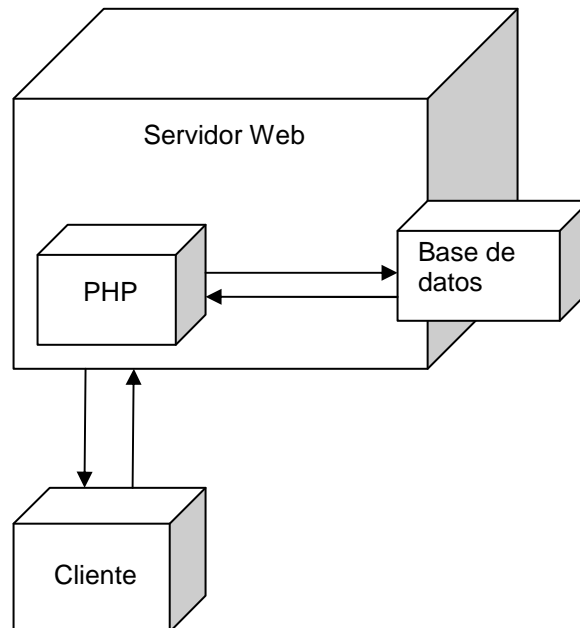


Figura 2.14. Esquema de funcionamiento PHP – BD.

#### 2.6.4. Instalación y configuración

En primer lugar, se necesita tener instalado un servidor de páginas web (*Apache*). Después se tiene que obtener la distribución fuente de PHP (Por lo general viene incluido en cualquier distribución de Linux). La última versión se puede descargar libremente de su sitio oficial, o en cualquiera de sus réplicas.

Ahora se tendrá un archivo llamado `php-4.x.x.tar.gz` y se tiene que descomprimir y desagruparlo para crear el árbol de directorio con los programas fuente de PHP. Se elige un directorio adecuado para crear el árbol de archivos de PHP, se copia en él el archivo `php-4.x.x.tar.gz` y se ejecuta lo siguiente:

```
$tar -xvfz php-4.x.x.tar.gz
```

Ahora, se tiene un directorio llamado `php-4.x.x` que contiene diversos archivos. Siempre es aconsejable leer los archivos `README` e `INSTALL` existentes por si hubiera alguna nota especial que tener en cuenta en el proceso de compilación e instalación.

El siguiente paso es configurar PHP para adaptarlo al sistema operativo y con las características que se necesiten. El programa *configure* realiza este proceso. Ejecute:

```
./configure --help
```

Y se obtendrá una lista de las características de configuración de PHP. Por lo que se puede ejecutar este programa con las opciones que se requieran.

Después se compila el programa, por lo tanto se debe ejecutar:

```
$make
```

Si la compilación ha sido correcta ahora se tiene que instalar el resultado obtenido:

```
$make install
```

Una vez instalado PHP en la máquina se tiene que configurar Apache para que sepa interpretar los programas PHP; se tiene que incluir en el archivo httpd.conf unas líneas similares a las siguientes. Puede que sólo se tenga que quitar el comentario a las líneas que vienen en el archivo de configuración preestablecido.

Primero, para identificar las extensiones de los archivos que lee el servidor:

```
AddType application/x-httpd-php .php
```

```
AddType application/x-httpd-php-source .phps
```

Evidentemente las extensiones también pueden ser php3, php4, asp o cualquier otra cosa que interese. Y también donde está el código PHP:

```
LoadModule php4_module ruta/libphp4.so
```

Sí se tiene en la configuración del Apache una directiva ClearModuleList, se tiene que añadir:

```
AddModule mod_php4.c
```

Ahora se reinicia el servidor Apache y se tendrá PHP disponible. Para verificar que esta funcionando PHP se crea un programa llamado phpinfo.php con las siguientes líneas:

```
<? phpinfo(); ?>
```

Y desde el programa navegador se teclea la siguiente dirección <http://localhost/phpinfo.php> y se comprobará el funcionamiento.

PHP dispone de un archivo de configuración donde se puede determinar ciertas características del comportamiento de PHP. El archivo de configuración se llama php3.ini en el caso de PHP 3 y php.ini en el caso de PHP4.

El archivo de configuración se lee cuando se inicia PHP. Si éste se usa como módulo del servidor Apache, el archivo de configuración se leerá al iniciar el servidor web, aunque algunas características se pueden modificar usando directivas de configuración de Apache y archivos .htaccess. En la versión CGI la lectura se realiza en cada llamada.

### 2.6.5. Funciones para el manejo de bases de datos (MySQL)

PHP provee de funciones para el manejo de bases de datos MySQL. A continuación se describirán las más importantes:

- ❖ `mysql_close`
  - `int mysql_close(int [link_identifier] );`

Devuelve: TRUE si se ha cerrado correctamente, FALSE en caso de error.

Cierra la conexión a la base de datos MySQL asociada al identificador de conexión especificado. Si no se especifica un identificador de conexión, se asume la de la última conexión abierta.

- ❖ `mysql_connect`
  - `int mysql_connect(string [hostname], string [username], string [password] );`

Devuelve: un identificador de conexión, o FALSE en caso de error.

Establece una conexión a un servidor de MySQL. Todos los argumentos son optativos, y si no se especifican, los valores por defecto son: el localhost, nombre del usuario que posee el proceso en el servidor, la contraseña vacía. La cadena hostname también puede incluir un número del puerto, "hostname:port".

En caso de realizar una segunda llamada a `mysql_connect` con los mismos argumentos, no se establecerá una nueva conexión, si no, se devolverá el identificador de conexión de la ya existente. La conexión al servidor se cerrará en cuanto la ejecución del script acabe, a menos que la cerremos antes con la función `mysql_close`. Cabe aclarar que se recomienda cerrar la conexión con la función anterior.

- ❖ `mysql_create_db`
  - `int mysql_create_db(string database_name, int [link_identifier] );`

Intenta crear una nueva base de datos en el servidor asociado con el identificador de conexión especificado.

- ❖ `mysql_drop_db`
  - `int mysql_drop_db(string database_name, int [link_identifier] );`

Devuelve: TRUE si todo ha ido bien, y FALSE en caso de error.  
Elimina una base de datos del servidor asociado al identificador de conexión.

- ❖ `mysql_fetch_array`
  - `array mysql_fetch_array(int result);`

Devuelve un array con la información correspondiente al resultado de una consulta, especificado por su identificador o FALSE si ya no hay más filas.

Es una versión extendida de `mysql_fetch_row()`. Además de almacenar los datos a través de índices numéricos del array, también lo hace a través de índices asociativos, utilizando los nombres de los campos como claves. Si dos o más columnas del resultado tienen el mismo nombre de campo, la última es la que tiene preferencia. Para acceder a las demás, es necesario utilizar el índice numérico o construir un alias para la columna.

- ❖ `mysql_fetch_object`
  - `int mysql_fetch_object(int result);`

Devuelve: un objeto o FALSE en caso de error.

Esta función es similar a `mysql_fetch_array`, sólo que los resultados de una consulta, en lugar de una tabla, los devuelve como un objeto. En este caso, sólo se puede acceder a los datos a través de los nombres de sus campos. La velocidad de ejecución es idéntica a la de `mysql_fetch_array`. Para referenciar el valor de un campo debemos utilizar el operador típico de los objetos ( `->` ).

- ❖ `mysql_num_fields`
  - `int mysql_num_fields(int result);`

Devuelve el número de campos de una consulta.

- ❖ `mysql_num_rows`
  - `int mysql_num_rows(string result);`

Devuelve el número de filas de una consulta.

- ❖ `mysql_query`
  - `int mysql_query(string query, int [link_identifier] );`

Ejecuta una consulta a la base de datos activa en el servidor asociado al identificador de conexión. Si no se especifica, se utiliza la última conexión abierta. Si no hay conexiones abiertas, la función intenta establecer una.

Esta función devuelve TRUE o FALSE para indicar si las operaciones UPDATE, INSERT o DELETE han sido ejecutadas correctamente. Para la operación SELECT devuelve un nuevo identificador de resultado.

- ❖ `mysql_select_db`
  - `int mysql_select_db(string database_name, int [link_identifier] );`

Establece la base de datos activa en el servidor. Si no se especifica identificador de conexión, se utiliza la última conexión abierta. Si no hay conexiones abiertas, la función intenta establecer una. A partir de la llamada a `mysql_select_db` las llamadas a `mysql_query`, actúan sobre la nueva base de datos activa.

## 2.7. HTML

El HTML acrónimo inglés de *Hypertext Markup Language*, es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Utiliza etiquetas o marcas que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en el navegador, así como también las imágenes y los demás elementos. Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (<>), y algunas tienen atributos que pueden tomar algún valor. A continuación se presenta la estructura básica de un documento HTML:

```
<html>
  <head>
    <title>Estructura</title>
  </head>
  <body>
    Formato y contenido de la página
  </body>
</html>
```

`<html>`. Es la etiqueta que define el inicio del documento, le indica al navegador que todo lo que sigue a continuación debe tratarlo como una serie de códigos html.

`<head>`. Define la cabecera del documento. Esta cabecera suele contener información sobre el documento, no se muestra directamente en el navegador. Dentro de la cabecera se puede encontrar:

`<title>`. Define el título de la página.

`<link>`. Se utiliza para definir algunas características avanzadas, como por ejemplo las hojas de estilo. Éstas, consisten en la creación de estilos propios dentro de las páginas web o en un archivo separado que será invocado por el sistema, para mejorar la presentación. Su funcionamiento consiste en definir la forma de presentación de un sitio web completo, un único documento html o una parte de

la página. La principal ventaja de utilizar hojas de estilo, es el control centralizado de la presentación de un sitio web completo, con lo que se agiliza de forma considerable la actualización del mismo.

`<body>`. Define el contenido principal o cuerpo del documento, este es la parte del documento html que se muestra en el navegador. Dentro del contenido se pueden tener formularios. La declaración del formulario queda especificada por las etiquetas `<form> ... </form>` y dentro de ellas se obtienen todas las variables de entrada. A la etiqueta de apertura le acompañan estos atributos:

`action=""`. Entre comillas se indica el programa que va a tratar las variables enviadas con el formulario, un guión CGI o el URL.

`method=""`. Indica el método de transferencia de las variables. Post, si se envía a través de la entrada estándar. Get, si se envía a través del URL.

## 2.8. JavaScript

JavaScript es un lenguaje de programación interpretado y diseñado para complementar las capacidades del html. El código es enviado al cliente como parte del código de la página, y puede ser usado para la creación de acciones y procesos de automatización para las páginas web.

Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

Los programas JavaScript van incrustados en los documentos html, y se encargan de realizar acciones en el cliente, como puede ser pedir datos, confirmaciones, mostrar mensajes, comprobar campos, etc.

## 2.9. Cookies

Una cookie, es un fragmento de información que se almacena en el disco duro del visitante de una página web a través del navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas. Al ser el protocolo http incapaz de mantener información por sí mismo, para que se pueda conservar información entre una página vista y otra, ésta debe ser almacenada ya sea en el URL de la página, en el propio servidor o en una cookie en la computadora del visitante.

El uso más frecuente de las cookies; es llevar el control de usuarios, cuando una persona introduce su nombre de usuario y contraseña, se almacena una cookie para que no tenga que estar introduciendo nuevamente los datos para cada página.

Originalmente sólo podían ser almacenadas por petición de un CGI desde el servidor, pero Netscape dio al lenguaje JavaScript la capacidad de introducir las directamente desde el cliente, sin necesidad de CGIs. En un principio, debido a errores del navegador se detectaron problemas de seguridad que posteriormente fueron resueltos. Las cookies pueden ser eliminadas, aceptadas o bloqueadas según se desee, para esto se debe configurar convenientemente el navegador web.

En la siguiente tabla se presenta la infraestructura final, es decir, cada una de las actividades a realizar y el software o método para llevar a cabo dicha actividad.

<b>Aspecto</b>	<b>Herramienta</b>
<b>Metodología de diseño</b>	❖ UML
<b>Sistema operativo</b>	❖ Red Hat Linux
<b>Manejador de base de datos</b>	❖ MySQL
<b>Servidor web</b>	❖ Apache
<b>Interfaz para el manejo de la base de datos</b>	❖ PHP
<b>Creación de pantallas</b>	❖ HTML
<b>Validación de formularios y transferencia de datos</b>	❖ JavaScript y Cookies

*Tabla 2.3. Infraestructura final.*

Establecida la metodología para el diseño de la base de datos y del comportamiento del sistema; además de haber descrito las características de cada uno de los componentes necesarios para la codificación, se pondrán en práctica para el desarrollo de la aplicación.

# **CAPÍTULO 3**

## **ANÁLISIS, DISEÑO Y PROGRAMACIÓN**

El presente capítulo se divide en dos partes fundamentales: el análisis, en el cual se describe el problema, los requerimientos a ser cubiertos e información concerniente a los procesos que se llevan a cabo; el diseño y programación, donde se define la solución al problema, por medio de un modelo que sea factible implementar. Esto se realiza siguiendo las especificaciones en cuanto a metodología de análisis y diseño de UML y del modelo de datos relacional.

### 3.1. Análisis

El objetivo de este proceso, es la descripción detallada de los problemas que se tienen y los requerimientos que serán cubiertos por el sistema de información; y que sirvan de base para el posterior diseño del sistema.

#### 3.1.1. Descripción del problema

En la Universidad Nacional Autónoma de México a nivel licenciatura, existen dos sistemas de estudio, el sistema escolarizado y el sistema abierto. A su vez; en la División Sistema Universidad Abierta, específicamente en la Facultad de Economía, se divide en dos tipos de cursos:

- ❖ Cursos presenciales. El sistema abierto ha brindado una opción presencial, con una estructura menor en términos de hora-clase, destinada a facilitar los estudios a quien por diversas razones no puede acudir a los cursos normales; o bien para quien desee elegir esta modalidad. Cabe resaltar que en las sesiones presenciales de grupo no se imparte clase, sino asesorías, las cuales no son obligatorias para efectos de la acreditación, pero sí muy recomendables. En las asesorías; el docente promueve una dinámica participativa que posibilita el enriquecimiento del estudio individual de los alumnos, mediante la reflexión, la discusión, el análisis y la síntesis del conocimiento obtenido. Cada semestre está organizado para que el alumno cubra sus asignaturas asistiendo una vez entre semana y el sábado.
- ❖ Cursos a distancia. Esta es una experiencia nueva que se está perfeccionando. Como todos los proyectos nuevos, requiere un esfuerzo adicional de alumnos y asesores. Los cursos a distancia son impartidos por medio de un sistema o plataforma educativa vía web. La dinámica es de la siguiente manera: el asesor envía el primer mensaje acompañado de una guía de estudio, indicando al alumno qué, cómo y cuándo se revisarán los temas de la asignatura. En la plataforma; establece la programación de fechas para trabajar y entrega de actividades, así como los exámenes pertinentes, permitiendo al alumno planificar sus acciones y estrategias de estudio para el mejor logro de los objetivos del curso. Esta modalidad educativa está dirigida para los alumnos que teniendo la posibilidad de dedicarle al estudio de la licenciatura en Economía el tiempo suficiente, no pueden desplazarse para asistir a clases presenciales. Además de tener acceso a los medios electrónicos, fundamentalmente para establecer comunicación con el asesor y los compañeros.

Una de las principales características de los alumnos de universidad abierta, es el limitado tiempo que tienen para dedicarle a los estudios. En este caso es preferible y se logra una mejor formación académica, sí en lugar de cursar las siete asignaturas previstas en el plan de estudios para los primeros semestres con bajo rendimiento, se dedica el tiempo disponible a cursar un número menor pero con un mejor aprovechamiento.

En la División Sistema Universidad Abierta de la Facultad de Economía; se realiza una evaluación semestral a los asesores; ya sea en curso presencial o en curso a distancia, por parte de los alumnos a través de encuestas, con la finalidad de verificar el desempeño en las clases y/o asesorías en las que esta asignado. Así como una encuesta de autoevaluación, para que los alumnos externen sus inquietudes y desempeño en esta modalidad.

Existen para tal objetivo las siguientes encuestas:



- ❖ Encuesta de autoevaluación.
- ❖ Encuesta de evaluación de asesores a distancia (Semestre 2005-1).
- ❖ Encuesta de evaluación de asesores presenciales.
- ❖ Encuesta de evaluación de asesores a distancia por medio de la plataforma (Semestre 2005-2 en adelante).

Las encuestas son contestadas únicamente por los alumnos, es un requisito para poderse inscribir al siguiente semestre. Hay que aclarar que si el alumno no inscribió asignaturas el semestre anterior, no es necesario que conteste las encuestas, solamente la de autoevaluación.

En la División los rezagos son bastantes, además de que la planta docente no es lo suficientemente comprometida con el sistema. Hay algunos asesores que solo llegan a firmar y no imparten asesoría (en el caso del curso presencial), y por supuesto los más afectados son los alumnos, que al no otorgárseles mayor interés terminan por abandonar el curso. En el caso contrario, el interés que los alumnos no tienen hacia las asesorías, al pensar que sólo tienen que resolver las actividades que les solicitan, sin participar en los foros de discusión o salas de charla.

Estos son algunos motivos por los cuales las encuestas tienen un valor muy importante para el Sistema Universidad Abierta, ya que en éstas se ve reflejada la opinión de los alumnos.

Los resultados de las encuestas son necesarios para ciertos trámites por parte de los asesores, como por ejemplo, para el mérito académico. Además; con los resultados se tiene un documento en el cual apoyarse para hablar con los asesores acerca de su desempeño en las asesorías. El proceso de llenado de encuestas se realiza de la siguiente manera:

El alumno recoge la papelería correspondiente en servicios escolares, dicha papelería incluye:

- ❖ Cédula de identificación.
- ❖ Solicitud de inscripción.
- ❖ Encuesta de autoevaluación.
- ❖ Dependiendo de la modalidad en que se cursaron las asignaturas, se les entrega una cantidad de encuestas igual al número de asignaturas cursadas. Las encuestas pueden ser de evaluación de asesores a distancia y evaluación de asesores presenciales.

El alumno para poderse inscribir a las asignaturas para el siguiente semestre; como se mencionó anteriormente, primero tiene que entregar las encuestas contestadas, sí el alumno no las entrega no le permiten la inscripción.

Las encuestas contienen preguntas acerca del desempeño del asesor, del temario de la asignatura, comentarios y sugerencias para la división entre otros aspectos. Hay cinco posibles respuestas u opciones, que son: nunca, ocasionalmente, algunas veces, frecuentemente y siempre, así como también hay de dos a tres preguntas abiertas. Además de que tienen que escribir el nombre del asesor, clave y nombre de la asignatura, y grupo.

Al momento de entregar las encuestas se subraya su nombre en un listado, permitiéndole pasar a las computadoras a registrar sus asignaturas.

Después de terminar el proceso de inscripción, las encuestas son procesadas y analizadas por dos psicólogas; las cuales tienen que generar un reporte por asesor, además de tomar en cuenta la opinión de los alumnos acerca de otros aspectos referentes al sistema universidad abierta. El

objetivo principal es tener una evaluación del desempeño de los asesores, alumnos y del sistema. Un formato de encuesta se presenta en los anexos.

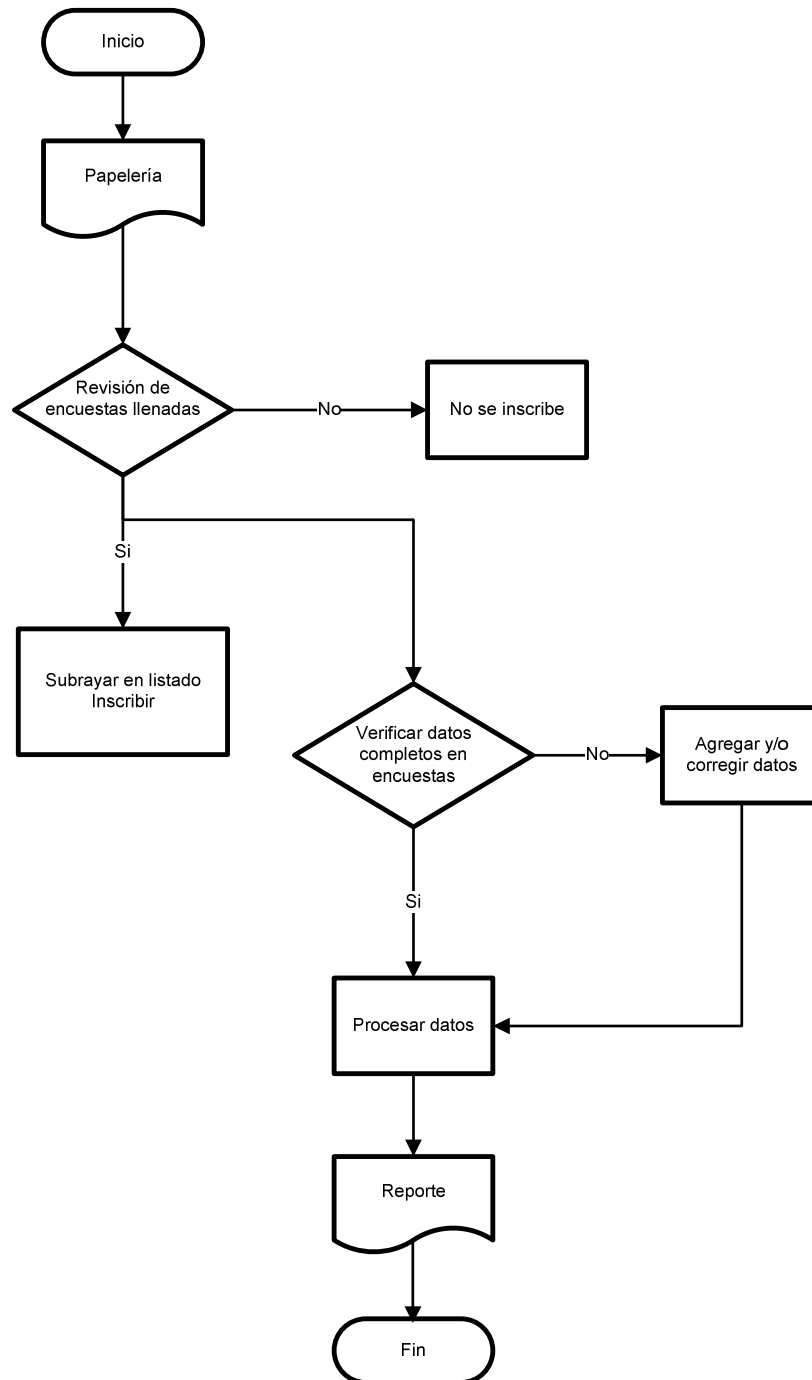


Figura 3.1. Diagrama de flujo del proceso de obtención de reportes.

Las desventajas encontradas en el proceso anteriormente descrito, son las siguientes:

- ❖ Demasiada información para ser procesada por dos personas, alrededor del 35% de una matrícula de 320 alumnos inscritos. Tomando en cuenta que las encuestas tienen todos los datos necesarios para su procesamiento.

- ❖ Alumnos que entregan encuestas en blanco.
- ❖ Datos incompletos en las encuestas; por ejemplo, no escriben el nombre del asesor o el de la asignatura, así como demás datos necesarios.
- ❖ El día de la inscripción muchos alumnos llegan a solicitar las encuestas y en ese momento las llenan, motivo por el cual no razonan las respuestas. Principalmente con los alumnos procedentes de otros estados.
- ❖ No se capta la opinión de todos los alumnos inscritos en la licenciatura, esto se debe a que muchos alumnos viven fuera de la Ciudad de México y no tienen la posibilidad de trasladarse a la Facultad. Alrededor del 30% del total de alumnos inscritos. Por lo que optan por enviar documentos por fax o llaman a servicios escolares para dar de alta sus asignaturas.
- ❖ Para poder obtener los reportes necesarios, tienen que introducir los datos en hojas de cálculo de Microsoft Excel®.
- ❖ Es un proceso repetitivo, tardado y tedioso. Motivos por los cuales, se pueden perder datos al momento de capturarlos.
- ❖ El proceso puede tardar de dos a tres meses para obtener los reportes.

### 3.1.2. Requerimientos generales del sistema de información

Debido a la problemática planteada anteriormente; se tiene la necesidad de un sistema de información que automatice el proceso, para obtener los resultados en menor tiempo, con mayor eficacia y veracidad. Por lo que se requiere que el nuevo sistema cuente con las siguientes características:

- ❖ Las encuestas sean llenadas vía Internet, sin necesidad de proporcionar los datos necesarios de sus asignaturas.
- ❖ Restringir el acceso a los usuarios a fin de que sólo las personas autorizadas puedan utilizar el sistema.
- ❖ Permita contestar todas las encuestas en una sola sesión o en varias sesiones.
- ❖ Automatice el proceso de emisión de reportes.
- ❖ Genere una lista de los alumnos que contestaron las encuestas.
- ❖ Permita crear nuevas encuestas de evaluación.

## 3.2. Diseño y Programación

En la etapa de diseño y programación se identifican los elementos que conforman el sistema, es decir, el modelo de la base de datos, las interfaces, diagramas, etc.

### 3.2.1. Descripción general del sistema

Para cubrir los requerimientos mencionados en la etapa de análisis, se propone un Sistema de Información vía Web, con el objetivo de ser utilizado sin importar la plataforma o el sistema operativo desde el cual se ingrese. Esto en relación a la compatibilidad y portabilidad del sistema. Además de que los alumnos que residen fuera de la Ciudad de México puedan contestar las encuestas.

La información se almacena en una base de datos relacional, para obtener un formato uniforme, mayor integridad y seguridad de datos.

Con el objetivo de que el usuario no interactúe directamente con el sistema manejador de base de datos, se crean interfaces gráficas. Con lo cual se pueden realizar las operaciones necesarias, como contestar encuestas, emisión de reportes, etc.

### 3.2.2. Diagrama de casos de uso

En el capítulo anterior se detallaron los componentes de un diagrama de casos de uso. A continuación se va a realizar el diagrama de casos de uso para visualizar el comportamiento del sistema propuesto.

Como una primera aproximación, se identifican a los usuarios que interactúan con el sistema.

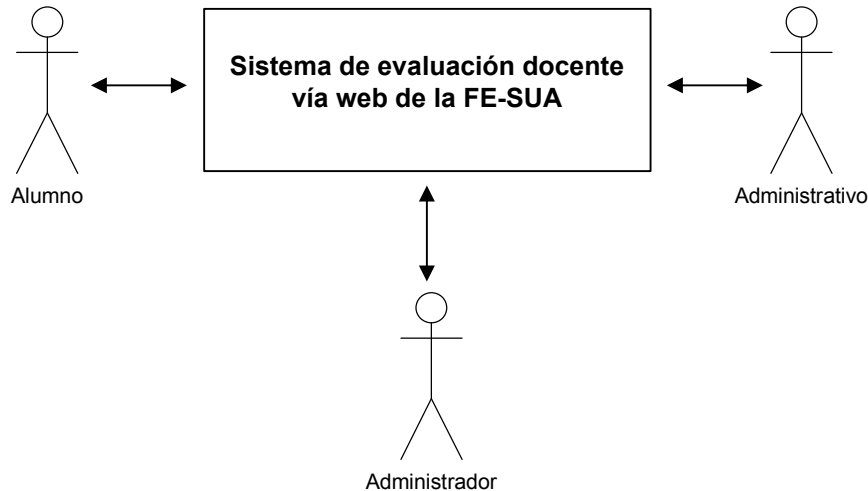


Figura 3.2. Usuarios del sistema.

En el siguiente paso, se desglosan las operaciones que los actores pueden realizar en el sistema.

- ❖ Alumno. El alumno puede contestar las encuestas.

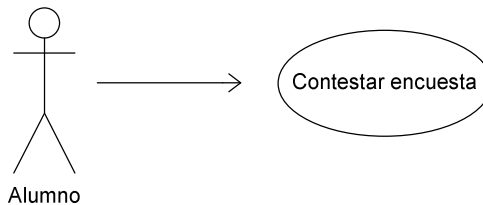


Figura 3.3. Usuario alumno.

Hay que notar que encuesta puede ser de asesores a distancia (2005-1 y 2005-2 en adelante), asesores presenciales o de autoevaluación. Así como también las nuevas encuestas.

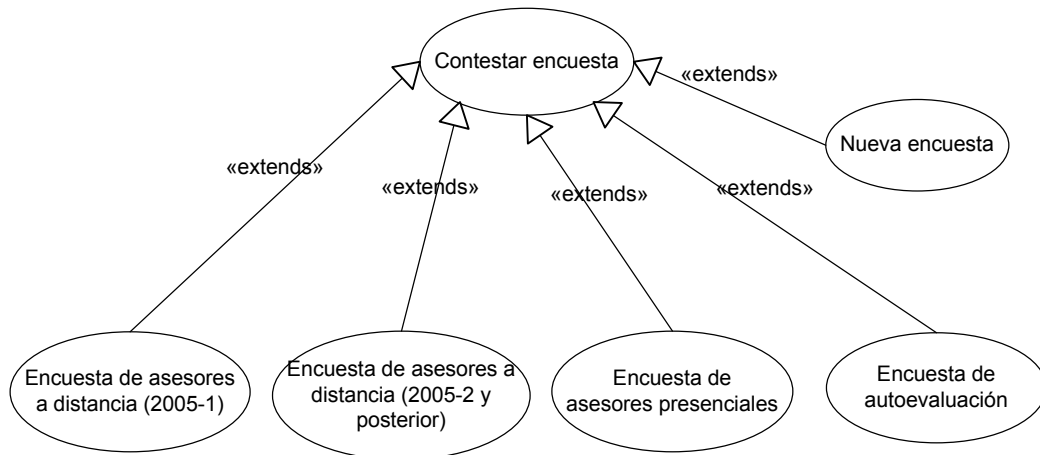


Figura 3.4. Operaciones del usuario alumno con el sistema.

- ❖ **Administrativo.** El personal administrativo puede generar el reporte de cada uno de los asesores y el de autoevaluación. Así como también, generar el reporte de los alumnos que contestaron las encuestas y crear nuevas encuestas de evaluación.

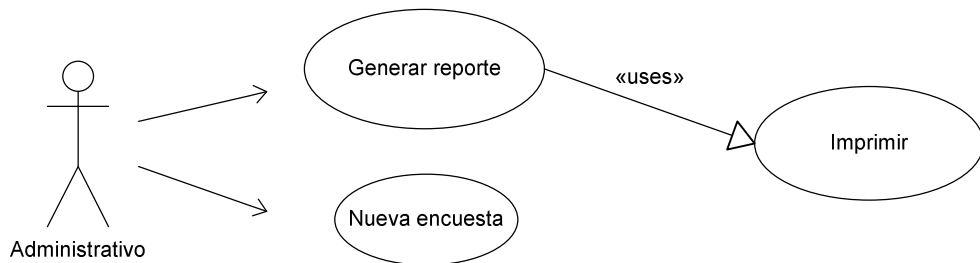


Figura 3.5. Usuario administrativo.

El reporte puede ser:

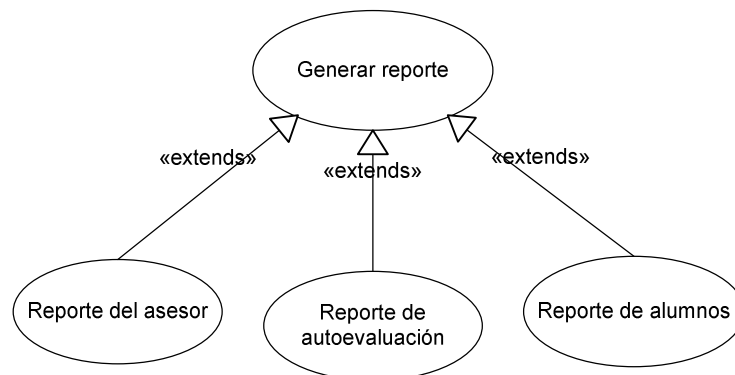


Figura 3.6. Operaciones del usuario administrativo con el sistema.

- ❖ **Administrador.** El administrador puede hacer altas, bajas y cambios de alumnos, asesores, asignaturas e inscripciones. También puede eliminar las respuestas almacenadas de semestres muy antiguos.

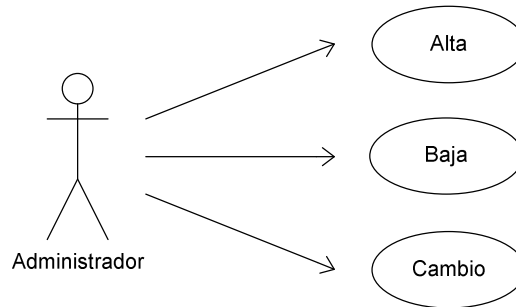


Figura 3.7. Usuario administrador.

Las operaciones pueden ser (Figuras 3.8, 3.9 y 3.10):

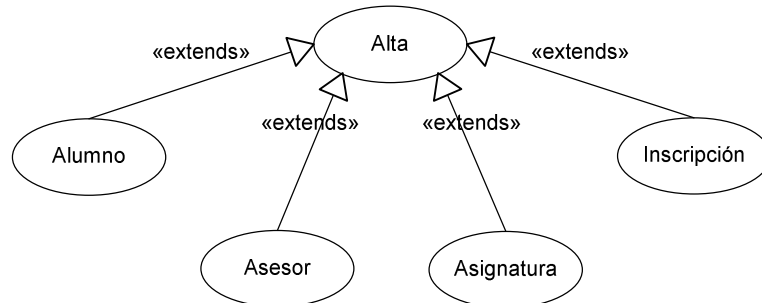


Figura 3.8. Operaciones del usuario administrador con el sistema (alta).

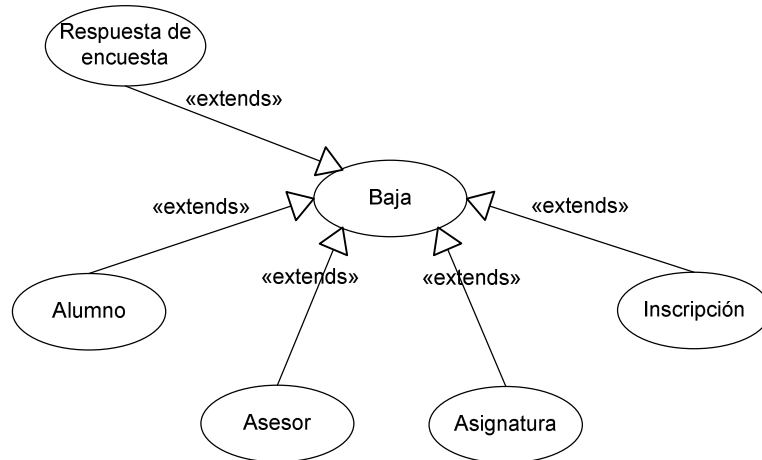


Figura 3.9. Operaciones del usuario administrador con el sistema (baja).

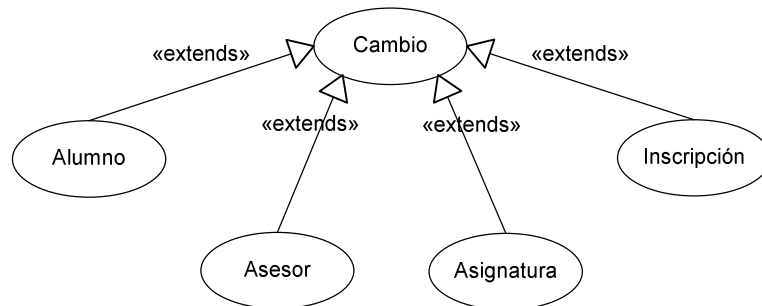


Figura 3.10. Operaciones del usuario administrador con el sistema (cambio).

El diseño completo del diagrama de casos de uso es el mostrado en la siguiente figura:

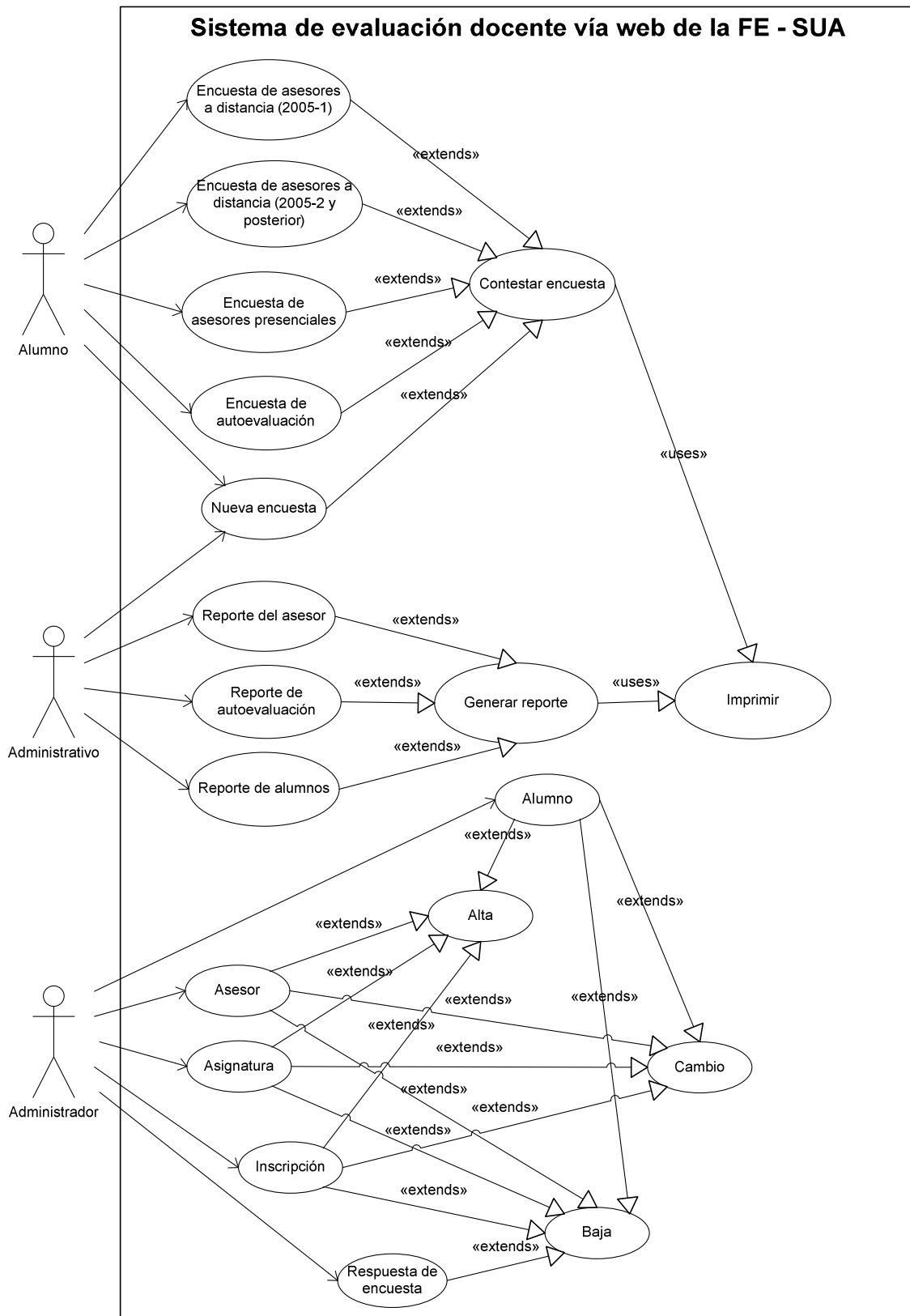


Figura 3.11. Diagrama de casos de uso del sistema.

### 3.2.3. Diagrama de clases

Este diagrama muestra el conjunto de clases y objetos importantes que conforman el sistema. Estos diagramas se caracterizan por presentar de una manera estática la estructura de información del sistema y la visibilidad que tiene cada una de las clases.

Primero se identifican las clases y sus características; obteniendo lo siguiente:

- ❖ Las clases principales en todo el sistema son las de *Alumno*, *Asignatura* y *Asesor*. Estas superclases, son las que definen las subclases; es decir, son las superclases que heredan tres atributos importantes o fundamentales para las subclases. Con esto tenemos la primera aproximación para el diagrama final:

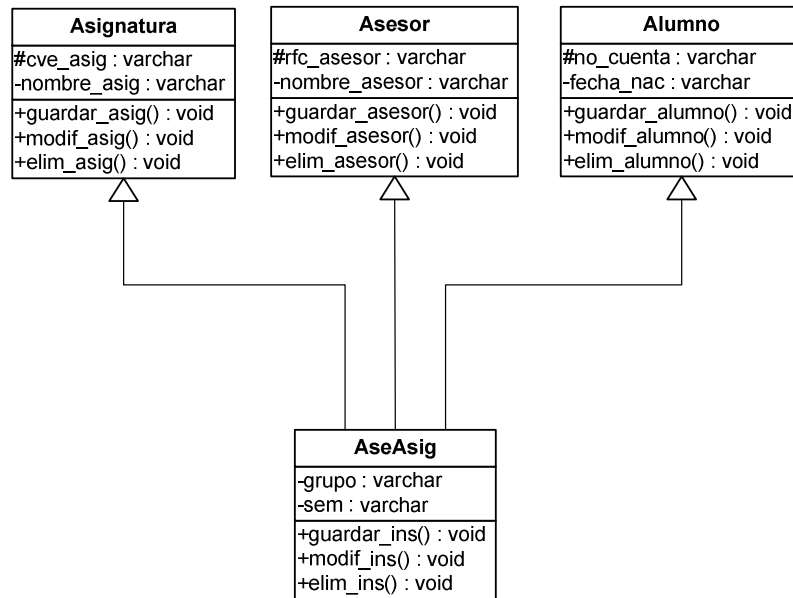


Figura 3.12. Clases principales.

En la figura anterior, se tiene que los atributos; *cve\_asig*, *rfc\_asesor* y *no\_cuenta* son los atributos heredados a la subclase *AseAsig*.

- ❖ La superclase o clase padre *Alumno*, hereda su atributo *no\_cuenta* a la subclase o clase hija *Autoevaluacion*, dicha relación se muestra en la siguiente figura:



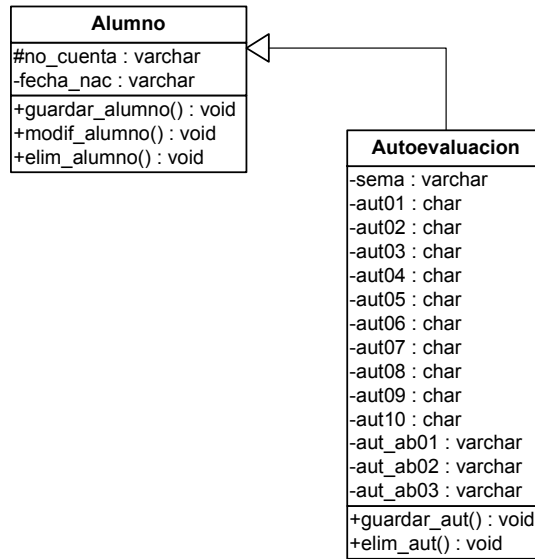


Figura 3.13. Relación de clases de Alumno - Autoevaluacion.

- ❖ La subclase AseAsig, se convierte en superclase para las subclases AseDistancia, AseDistancianva, AsePresencial y Respuesta, figura 3.14.

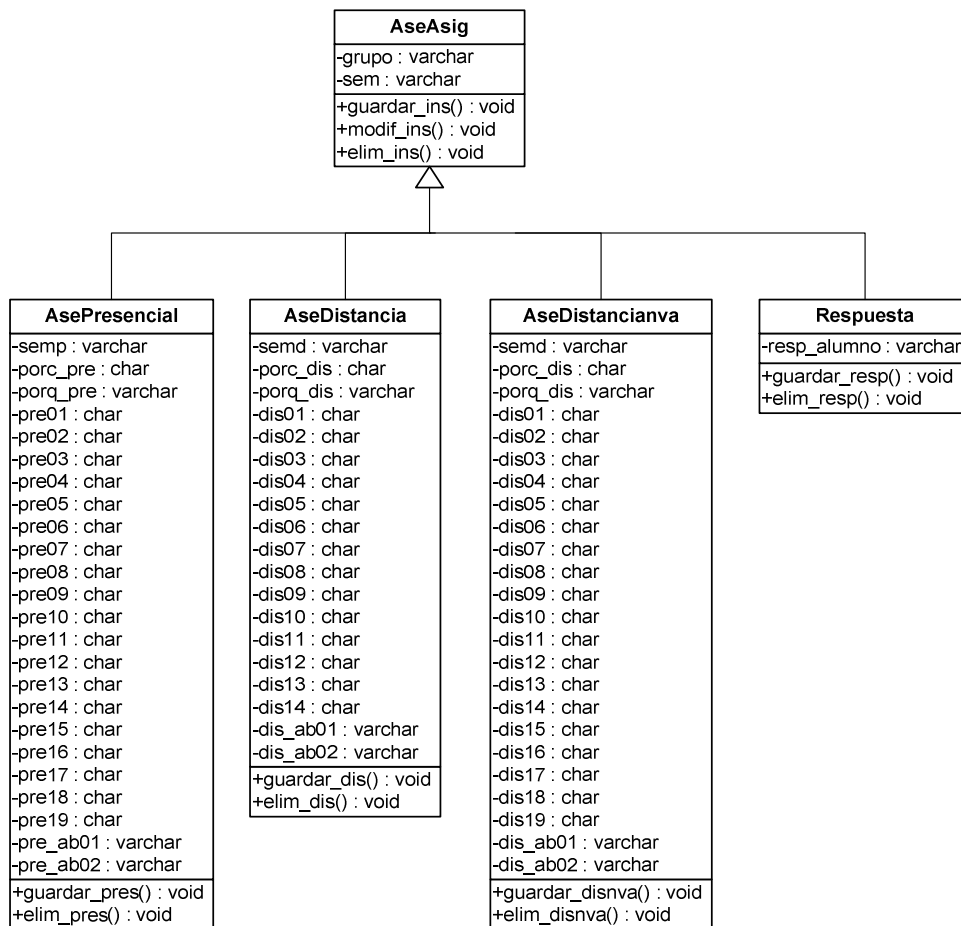


Figura 3.14. Relación de la clase AseAsig con otras clases.

- ❖ Para la creación de nuevas encuestas; tenemos que la clase *Evaluacion* se define como la clase padre para *Pregunta*, que es la clase hija. Heredando el atributo *id\_eval*.

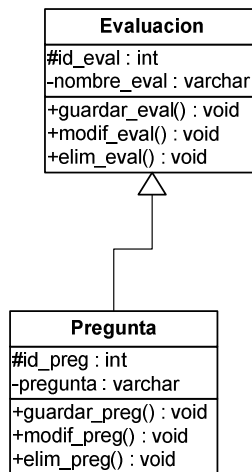


Figura 3.15. Clase principal para la creación de nuevas encuestas.

- ❖ La subclase *Respuesta* se define de esta forma; por el hecho de heredar de las superclases *Pregunta*, *Opcion* y *AseAsig* los atributos *id\_eval* e *id\_preg*; *id\_op*; y *cve\_asig*, *rfc\_asesor* y *no\_cuenta* respectivamente.

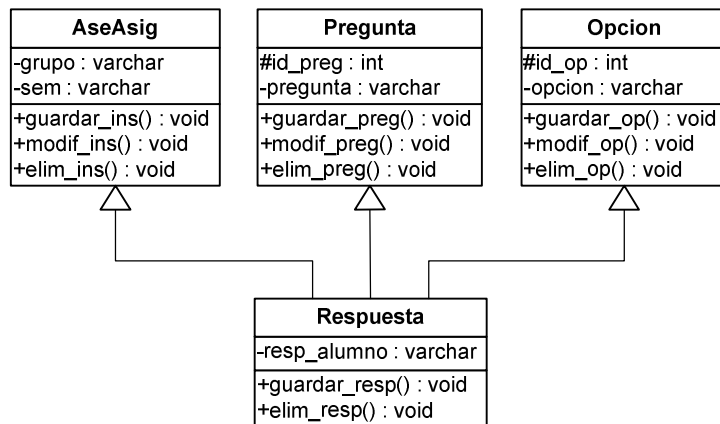


Figura 3.16. Relación de la subclase *Respuesta* con otras clases.

Después de identificar las clases que forman parte del sistema, sus características, atributos y relaciones, el diagrama final es mostrado en la siguiente figura:

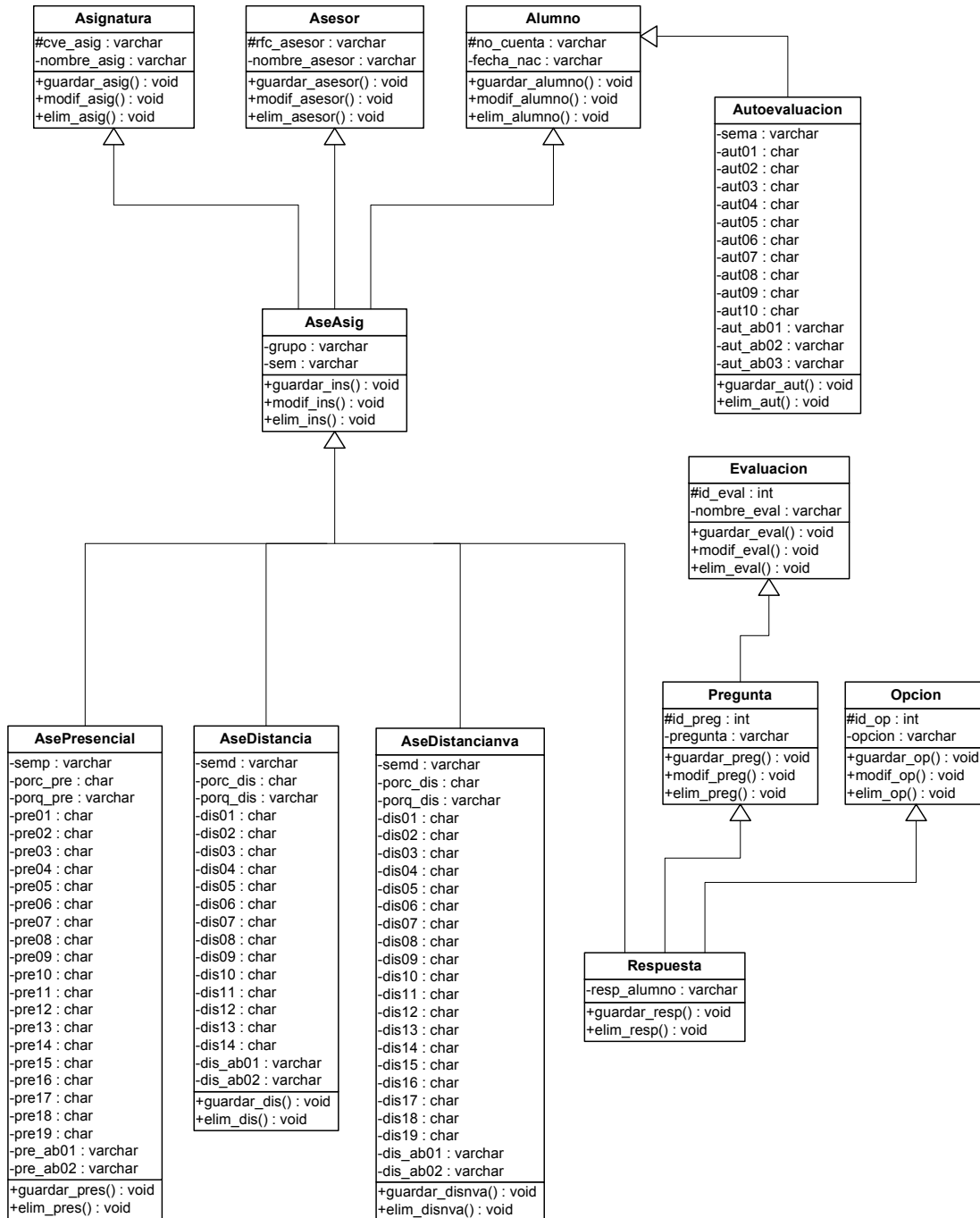


Figura 3.17. Diagrama de clases del sistema.

### 3.2.4. Diagrama de secuencia

En el diagrama de secuencia se muestra el orden de las llamadas en el sistema. A continuación se describe el diagrama correspondiente a cada uno de los usuarios o actores que interactúan con el sistema.

- ❖ Alumno. El alumno inicia sesión en el sistema con su número de cuenta y fecha de nacimiento, si los datos son incorrectos, el sistema los vuelve a solicitar. Como es requisito indispensable contestar la encuesta de autoevaluación, es la primera que tiene que contestar. Después se llama a un método para validar que los campos

hayan sido llenados, sino el sistema manda un mensaje para que los datos faltantes sean contestados. Si el sistema almacenó los datos, manda un mensaje de confirmación. El proceso se repite para las encuestas de asesores a distancia (ambos tipos), asesores presenciales y nuevas encuestas. Al terminar de contestar, se imprime el comprobante y se cierra sesión. Gráficamente se representa en la figura 3.18:

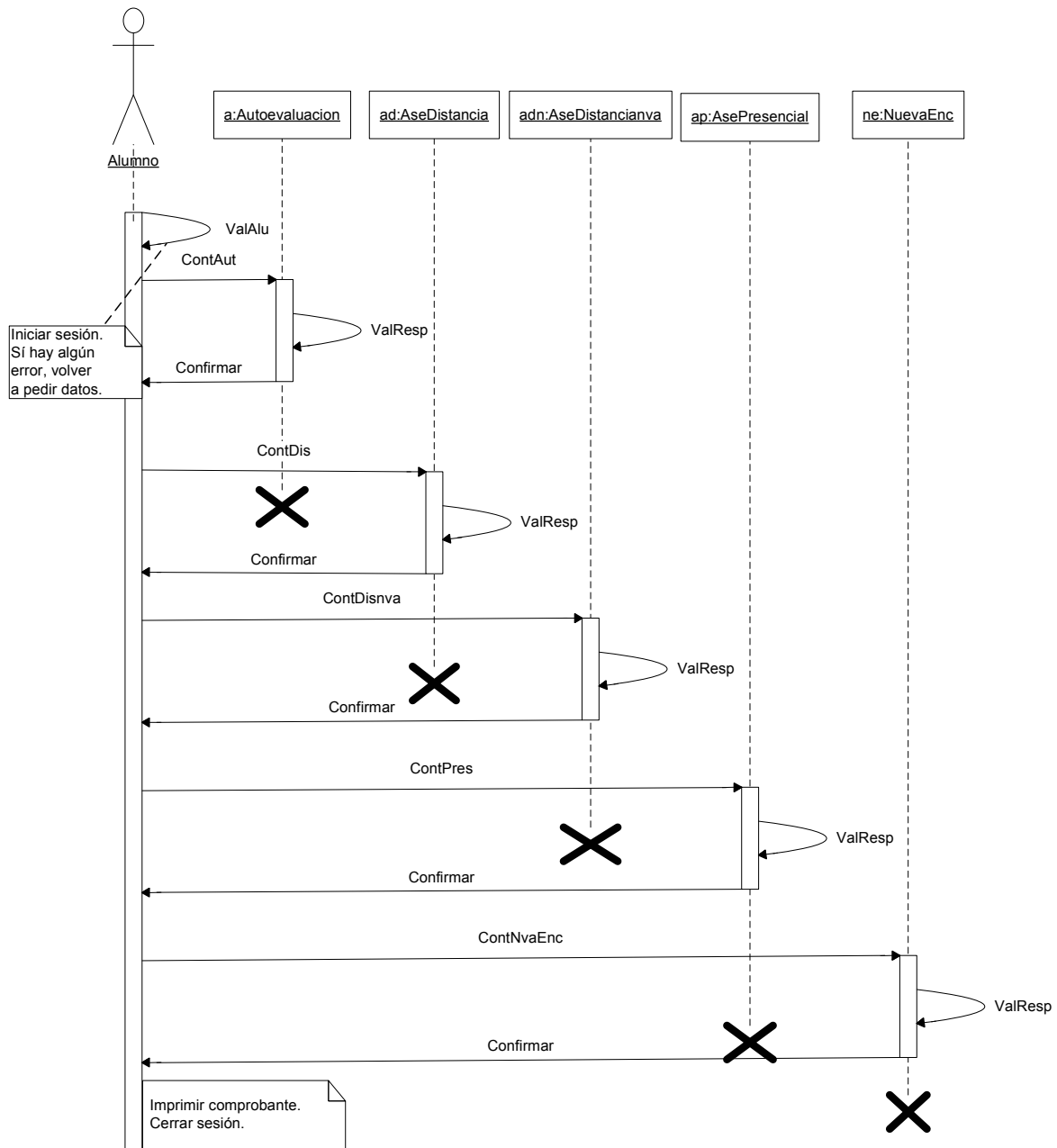


Figura 3.18. Diagrama de secuencia del alumno.

- ❖ **Administrativo.** Inicia sesión con una clave especial. El método valida la clave y si es correcta permite el acceso al sistema; en caso contrario, vuelve a solicitar la clave. El actor consulta o genera el reporte de alumnos que contestaron encuestas, el sistema valida el semestre y le permite imprimirla. Los pasos se repiten para los reportes de asesores y de autoevaluación. Para el objeto *NuevaEnc*, hace la petición e introduce

los datos solicitados; como es: el nombre de la encuesta, preguntas y opciones. El método *ValDatos*, los revisa y manda el mensaje de confirmación, por último cierra la sesión. El diagrama correspondiente es mostrado en la figura 3.19:

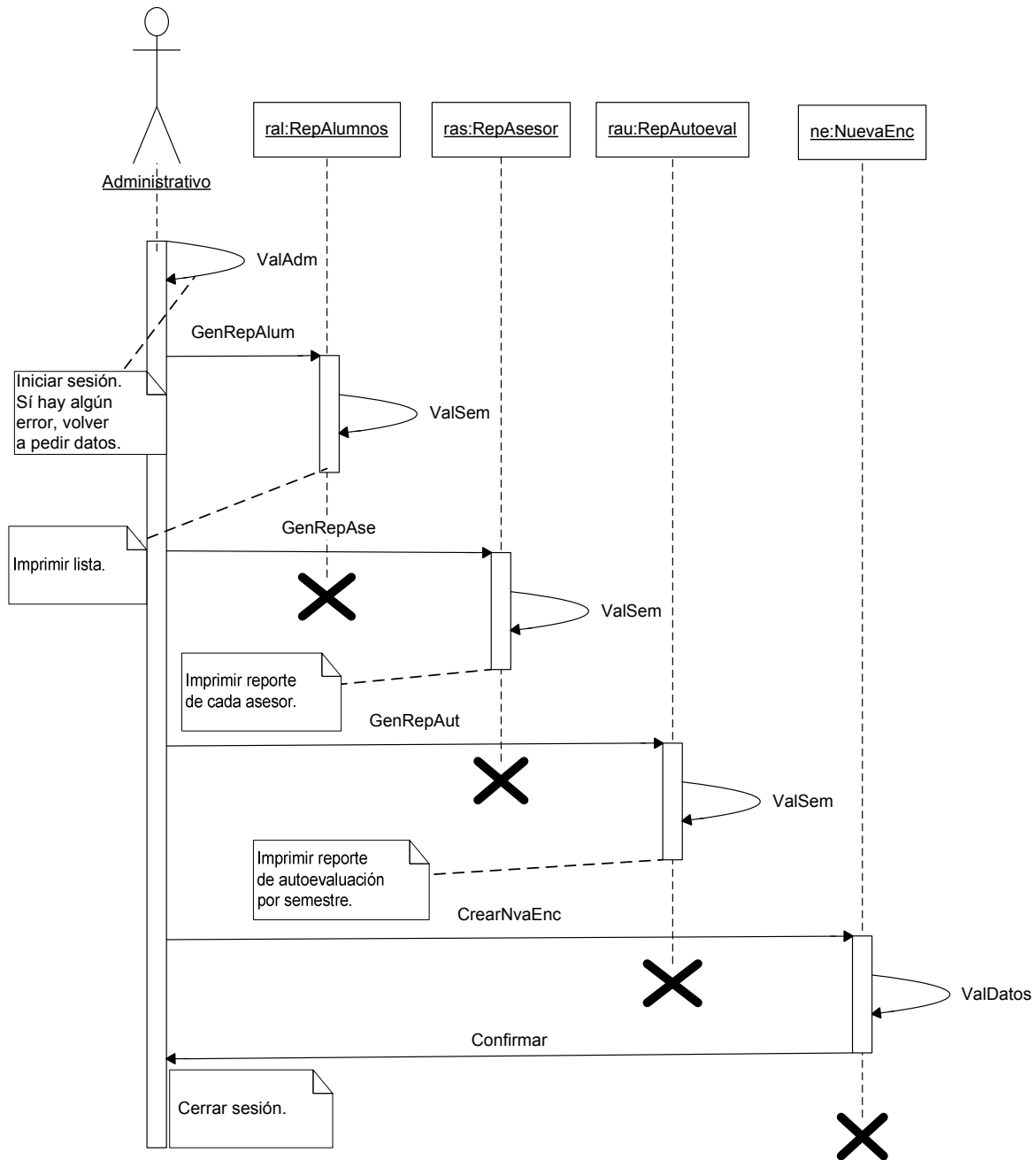


Figura 3.19. Diagrama de secuencia del administrativo.

- ❖ Administrador. El rol de administrador inicia sesión con una clave especial, si es incorrecta se vuelve a pedir la clave. El administrador puede hacer la petición de las siguientes operaciones: alta, baja y cambio de alumnos, asesores, asignaturas e inscripción. Los datos son validados por sus respectivos métodos. Al terminar de ejecutarse las solicitudes, manda un mensaje de confirmación. Para el caso del objeto *RespEnc*, sólo puede realizar una operación; que es eliminar. Al igual que las

anteriores, el dato es validado y después se manda el mensaje de confirmación. Permitiendo cerrar la sesión. El diagrama final para éste rol, es el siguiente:

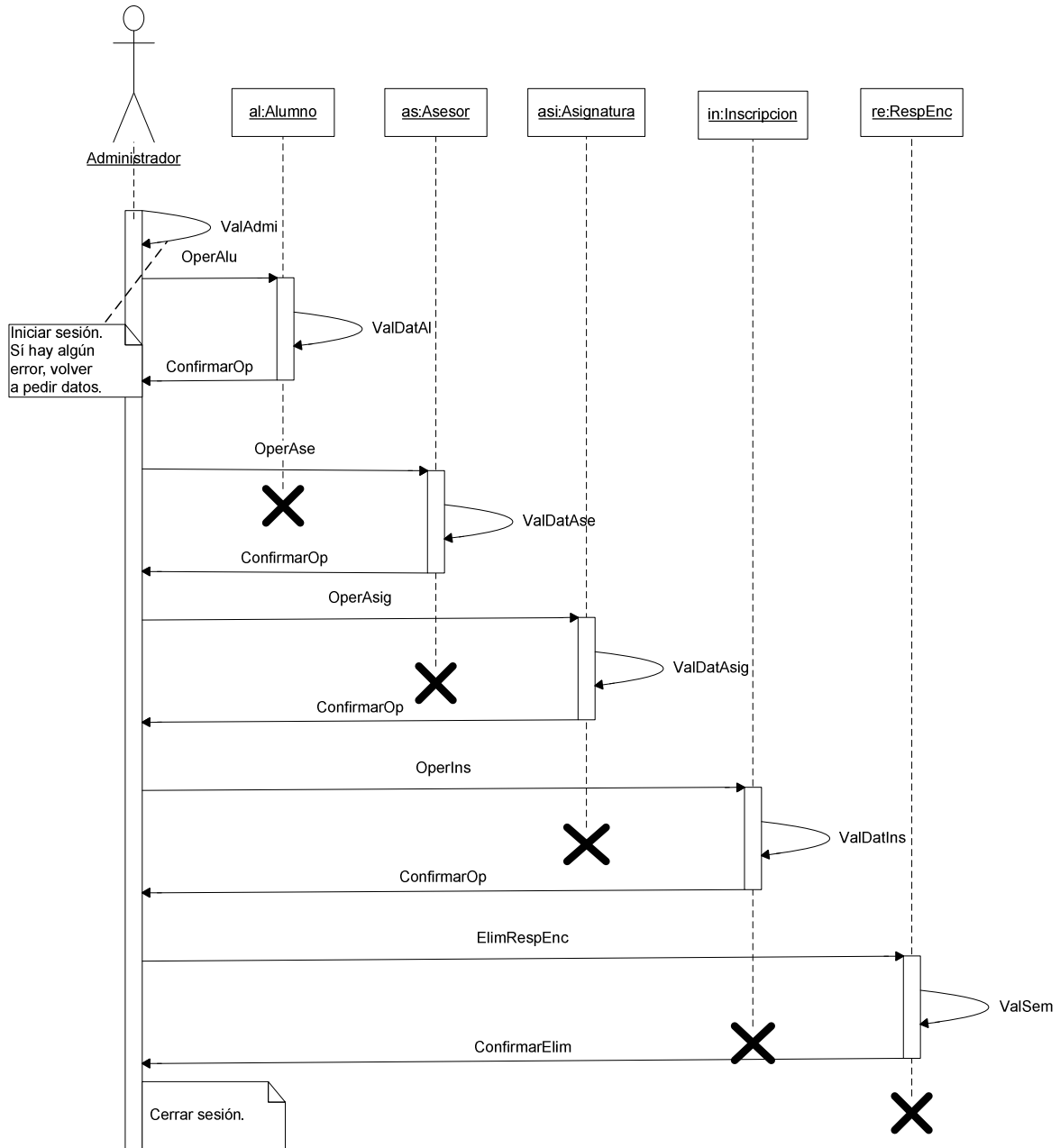


Figura 3.20. Diagrama de secuencia del administrador.

### 3.2.5. Base de datos

En el capítulo anterior se describieron las características del modelo de datos relacional, en base a ello se eligió este modelo para diseñar la base de datos.

#### 3.2.5.1. Diseño de la base de datos

El diseñar una base de datos no es tarea sencilla, motivo por el cual si se siguen las reglas de normalización se obtiene un diseño eficiente y factible de implementar.

Primero hay que identificar los elementos que pueden ser entidades; en este caso, los diferentes tipos de encuestas son posibles entidades. Así como también los alumnos, que son los que van a contestar las encuestas.

Para poder crear nuevas encuestas, se necesitan entidades para el tipo de encuesta, las preguntas y sus posibles respuestas.

Como el principal objetivo del sistema es automatizar los procesos descritos en el apartado anterior, se tienen que almacenar los datos de los asesores y de las asignaturas; esto para evitar perder datos fundamentales.

Con las entidades ya definidas, hay que comenzar a detectar los atributos de cada una de ellas. Para después aplicar la primera forma normal y así poder obtener la llave primaria, es decir, el atributo que identifica la entidad.

Teniendo la llave primaria, se revisa que todos los demás atributos dependan de ella o que tengan relación. Cumpliendo así la segunda forma normal.

Al identificar las entidades, hay que analizar las posibles relaciones entre las abstracciones. Para ello hay que evitar las relaciones muchos a muchos, con una entidad o tabla transitiva.

**3.2.5.2. Diagrama Entidad – Relación de la base de datos**

En la siguiente figura se presenta el diagrama final:

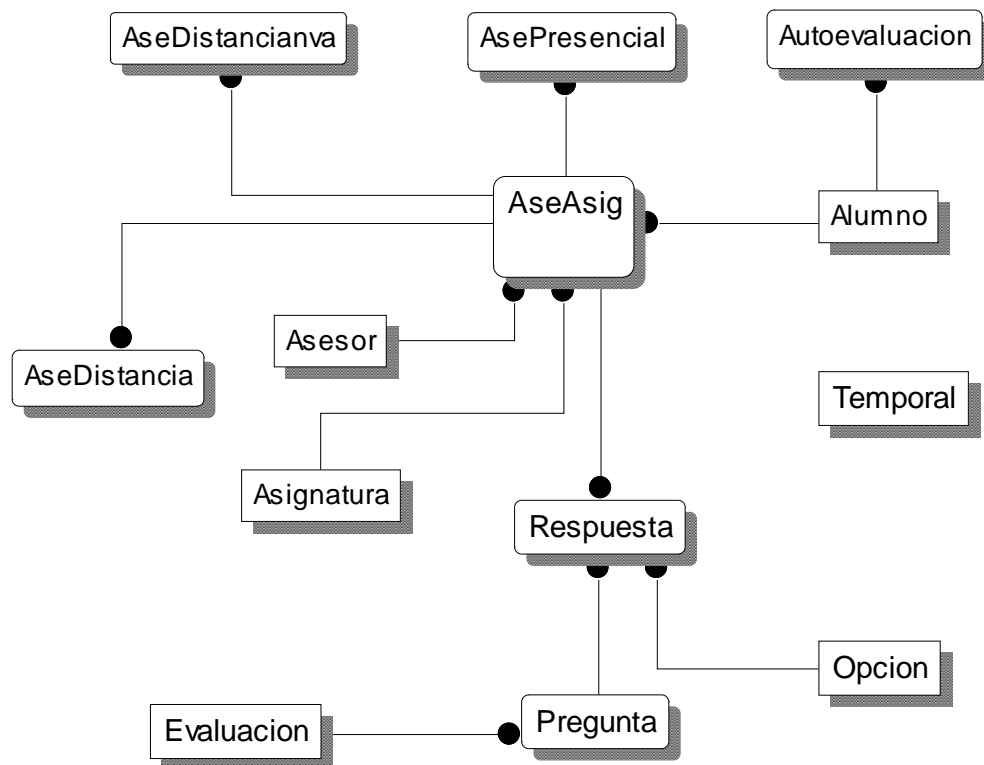


Figura 3.21. Diagrama Entidad – Relación de la base de datos (simplificado).

### 3.2.5.3. Diccionario de datos

El resultado de la etapa anterior es el esquema de la base de datos; que haciendo uso del DDL o Lenguaje de Definición de Datos, se generan las instrucciones necesarias para crear la base de datos. A continuación se describe el diccionario de datos en conjunto con el sistema manejador de base de datos, en este caso, MySQL.

Entidad: Alumno				
Llave	Atributo	Tipo	Tamaño	Null
Primary	no_cuenta	varchar	9	No
	fecha_nac	varchar	8	No

Entidad: Asesor				
Llave	Campo	Tipo	Tamaño	Null
Primary	rfc_asesor	varchar	10	No
	nombre_asesor	varchar	100	No

Entidad: Asignatura				
Llave	Campo	Tipo	Tamaño	Null
Primary	cve_asig	varchar	4	No
	nombre_asig	varchar	100	No

Entidad: AseAsig				
Llave	Campo	Tipo	Tamaño	Null
Primary	cve_asig	varchar	4	No
Primary	rfc_asesor	varchar	10	No
Primary	no_cuenta	varchar	9	No
Primary	sem	varchar	6	No
	grupo	varchar	4	Si

Entidad: AseDistancia				
Llave	Campo	Tipo	Tamaño	Null
Primary	cve_asig	varchar	4	No
Primary	rfc_asesor	varchar	10	No
Primary	no_cuenta	varchar	9	No
Primary	semd	varchar	6	No
	porc_dis	char	1	No
	porq_dis	varchar	100	Si
	dis01	char	1	Si
	dis02	char	1	Si
	dis03	char	1	Si
	dis04	char	1	Si
	dis05	char	1	Si
	dis06	char	1	Si
	dis07	char	1	Si
	dis08	char	1	Si
	dis09	char	1	Si
	dis10	char	1	Si
	dis11	char	1	Si
	dis12	char	1	Si
	dis13	char	1	Si
	dis14	char	1	Si
	dis_ab01	varchar	200	Si
	dis_ab02	varchar	200	Si



<b>Entidad: AseDistancianva</b>				
Llave	Campo	Tipo	Tamaño	Null
Primary	cve_asig	varchar	4	No
Primary	rfc_asesor	varchar	10	No
Primary	no_cuenta	varchar	9	No
Primary	semd	varchar	6	No
	porc_dis	char	1	No
	porq_dis	varchar	100	Si
	dis01	char	1	Si
	dis02	char	1	Si
	dis03	char	1	Si
	dis04	char	1	Si
	dis05	char	1	Si
	dis06	char	1	Si
	dis07	char	1	Si
	dis08	char	1	Si
	dis09	char	1	Si
	dis10	char	1	Si
	dis11	char	1	Si
	dis12	char	1	Si
	dis13	char	1	Si
	dis14	char	1	Si
	dis15	char	1	Si
	dis16	char	1	Si
	dis17	char	1	Si
	dis18	char	1	Si
	dis19	char	1	Si
	dis_ab01	varchar	200	Si
	dis_ab02	varchar	200	Si

<b>Entidad: AsePresencial</b>				
Llave	Campo	Tipo	Tamaño	Null
Primary	cve_asig	varchar	4	No
Primary	rfc_asesor	varchar	10	No
Primary	no_cuenta	varchar	9	No
Primary	semp	varchar	6	No
	porc_pre	char	1	No
	porq_pre	varchar	100	Si
	pre01	char	1	Si
	pre02	char	1	Si
	pre03	char	1	Si
	pre04	char	1	Si
	pre05	char	1	Si
	pre06	char	1	Si
	pre07	char	1	Si
	pre08	char	1	Si
	pre09	char	1	Si
	pre10	char	1	Si
	pre11	char	1	Si
	pre12	char	1	Si
	pre13	char	1	Si
	pre14	char	1	Si
	pre15	char	1	Si
	pre16	char	1	Si

	pre17	char	1	Si
	pre18	char	1	Si
	pre19	char	1	Si
	pre_ab01	varchar	200	Si
	pre_ab02	varchar	200	Si

Entidad: Autoevaluacion				
Llave	Campo	Tipo	Tamaño	Null
Primary	no_cuenta	varchar	9	No
Primary	sema	varchar	6	No
	aut01	char	1	Si
	aut02	char	1	Si
	aut03	char	1	Si
	aut04	char	1	Si
	aut05	char	1	Si
	aut06	char	1	Si
	aut07	char	1	Si
	aut08	char	1	Si
	aut09	char	1	Si
	aut10	char	1	Si
	aut_ab01	varchar	200	Si
	aut_ab02	varchar	200	Si
	aut_ab03	varchar	200	Si

Entidad: Temporal				
Llave	Campo	Tipo	Tamaño	Null
Primary	cuenta	varchar	9	No
	nombre	varchar	100	Si

Entidad: Evaluacion				
Llave	Campo	Tipo	Tamaño	Null
Primary	id_eval	int	11	No
	nombre_eval	varchar	50	No

Entidad: Pregunta				
Llave	Campo	Tipo	Tamaño	Null
Primary	id_preg	int	11	No
Primary	id_eval	int	11	No
	pregunta	varchar	200	No

Entidad: Respuesta				
Llave	Campo	Tipo	Tamaño	Null
Primary	cve_asig	varchar	4	No
Primary	rfc_asesor	varchar	10	No
Primary	no_cuenta	varchar	9	No
Primary	id_preg	int	11	No
Primary	id_eval	int	11	No
Primary	id_op	int	11	No
Primary	sem	varchar	6	No
	resp_alumno	varchar	50	Si

Entidad: Opcion				
Llave	Campo	Tipo	Tamaño	Null
Primary	Id_op	int	11	No
	Opcion	varchar	50	No

### 3.2.6. Especificación de los módulos del sistema

En los siguientes apartados, se describen los tres módulos que conforman el sistema.

#### 3.2.6.1. Módulo de usuarios

Este módulo es el encargado de interactuar con los usuarios, que en este caso son los alumnos y administrativos. Los alumnos pueden realizar las siguientes acciones:

- ❖ Contestar la encuesta de autoevaluación. Esta encuesta es obligatoria y consta de diez preguntas de opción múltiple y tres preguntas abiertas. El alumno después de contestar todas las preguntas; envía las respuestas, el sistema comprueba que todos los campos hayan sido llenados y después almacena las respuestas en la base de datos.
- ❖ Contestar la encuesta de asesores a distancia (2005-1). Este tipo de encuesta es para las asignaturas que se cursaron a distancia, antes del semestre 2005-1. Se estructura de catorce preguntas de opción múltiple, dos preguntas abiertas, así como del porcentaje de trabajo en el curso y los motivos de dicho porcentaje. El alumno envía las respuestas y el sistema revisa que todos los campos hayan sido contestados y después las respuestas las guarda en la base de datos. El número de encuestas depende del número de asignaturas cursadas en el semestre.
- ❖ Contestar la encuesta de asesores a distancia (2005-2 y posteriores). La encuesta de asesores a distancia se aplica a alumnos que cursaron la asignatura a distancia, pero a través de la plataforma educativa en el semestre 2005-2 y posteriores. Dicha encuesta esta formada por diecinueve preguntas de opción múltiple, dos preguntas abiertas, también por el porcentaje de trabajo en el curso y los motivos del porque de dicho porcentaje. Al enviar la encuesta contestada, el sistema corrobora que todos los campos hayan sido llenados, para después almacenar en la base de datos esa información.
- ❖ Contestar la encuesta de asesores presenciales. La encuesta tiene el objetivo de evaluar a los asesores de cursos presenciales, tiene diecinueve preguntas de opción múltiple, dos preguntas abiertas, el porcentaje de asistencia a asesorías y las razones de dicho porcentaje. Envía las respuestas, el sistema valida los campos y después almacena las respuestas en la base de datos.
- ❖ Contestar las nuevas encuestas. Este punto se refiere a las nuevas encuestas que se van a ir creando, su estructura dependerá del aspecto a evaluar.
- ❖ Imprimir comprobante. Después de que contestó todas las encuestas y sus respuestas fueron almacenadas, el alumno puede imprimir su comprobante de que lleno encuestas.

Para la encuesta de autoevaluación, sólo se aplica una por alumno. En el caso de encuestas de asesores a distancia (ambos tipos) y de asesores presenciales, el número de encuestas depende del número de asignaturas que cursó durante ese semestre. Para las nuevas encuestas, depende de las políticas de la división.

Los administrativos lo que pueden hacer es:

- ❖ Crear nuevas encuestas. El sistema permite al administrativo generar nuevas encuestas, para ello primero tiene que almacenar el nombre de la nueva encuesta y el número de preguntas que tendrá. Los datos se validan y el sistema permite ingresar el número de preguntas especificadas. En esta parte define si la pregunta es de opción múltiple o abierta. El sistema verifica los datos y los almacena. El administrador es el encargado de activar las nuevas encuestas, en base a las políticas de la división.

### 3.2.6.2. Módulo de administración

El módulo es utilizado por el administrador, que es el encargado de proporcionar mantenimiento al sistema, llevando a cabo las siguientes acciones:

- ❖ El administrador puede dar de alta un nuevo alumno, almacenando en la base de datos el número de cuenta y la fecha de nacimiento. Así como también puede modificar o eliminar los datos del alumno. Las tres operaciones son validadas por el sistema, después los datos son almacenados en la base de datos.
- ❖ El encargado del módulo puede dar de alta, modificar e incluso eliminar un asesor. Para agregar un nuevo asesor, el sistema solicita los datos, que son: el registro federal de contribuyentes sin los dígitos verificadores y el nombre completo del asesor. En el caso de modificar o eliminar, se solicita solamente el registro federal de contribuyentes, campo que es validado por el sistema.
- ❖ En la parte de asignaturas, el administrador puede ingresar una nueva asignatura con la clave y el nombre, y al igual que las secciones antes descritas, el sistema valida los datos y si son correctos, los almacena en la base de datos. También se puede modificar o eliminar asignaturas ya almacenadas, solamente con la clave de esta.
- ❖ Los registros de inscripción se dan de alta con los datos de la clave de asignatura, el registro federal de contribuyentes del asesor, el semestre, el grupo y el número de cuenta del alumno que cursó dicha asignatura. También se pueden modificar o eliminar registros, para ser almacenados o eliminados, los datos son validados.
- ❖ Los registros que almacenan las respuestas de los alumnos, pueden ser depurados por el administrador, para evitar tener información no necesaria. Es decir, las respuestas que se almacenan en la base de datos de cada uno de los tipos de encuestas, deben ser eliminados cada cierto tiempo.

### 3.2.6.3. Módulo de reportes

Esta parte se encarga de generar los reportes requeridos por la división. Los cuales son:

- ❖ Reporte de asesores a distancia (2005-1) y presenciales. El reporte refleja el nombre y clave de la asignatura, grupo y nombre del asesor; el número de alumnos inscritos, el número de alumnos que contestaron y que no contestaron, y el porcentaje de alumnos que llenaron la encuesta. Así como también el número de alumnos que trabajaron en el curso, agrupados por el porcentaje. Para poder obtener un porcentaje global del asesor en cada una de las asignaturas impartidas, a las opciones se les asigna un valor que va de 0 a 10; sumando el total de cada una de las preguntas. Tomando como puntaje máximo el resultado de la multiplicación del número de alumnos que trabajaron más del 1% por 10. El puntaje global se obtiene, sumando el puntaje obtenido entre la suma del puntaje máximo y se multiplica por 100. Las respuestas a las preguntas abiertas, sólo se muestran sin ningún procesamiento.
- ❖ Reporte de asesores a distancia (2005-2 y posteriores). El reporte presenta el nombre y clave de la asignatura, grupo y nombre del asesor; número de alumnos inscritos, número de alumnos que contestaron y que no contestaron, y porcentaje de alumnos que llenaron la encuesta. Así como también el número de alumnos que trabajaron en

el curso, agrupados por el porcentaje. En este tipo de reporte las opciones tienen un valor de 0 a 10, sumando el total de cada una de las preguntas se conoce el puntaje obtenido de esa pregunta. El puntaje máximo es el resultado de multiplicar el número de alumnos que trabajaron en el curso más del 1% por 10. Las preguntas están agrupadas por aspectos a evaluar, como son: responsabilidad que tiene un valor del 15%, evaluación con valor de 16%, motivación con valor de 32% y aprendizaje con 37%, dando como resultado el 100%. Para conocer el puntaje global, se suma el resultado del puntaje obtenido por el valor asignado al rubro, dividido entre el puntaje máximo, esto con cada aspecto antes mencionado. Aquí también se incluye una pequeña autoevaluación, la cual se procesa como los reportes anteriores, pero con la diferencia de que el valor obtenido se multiplica por 100 y se divide entre el valor máximo. Las respuestas a las preguntas abiertas no se procesan, sólo se despliegan.

- ❖ Reporte de autoevaluación. El reporte esta formado por el número de alumnos inscritos y el número de alumnos que contestaron la encuesta. En este caso; en cada una de las opciones se muestra el número y el porcentaje de alumnos que eligieron la opción, esto en cada una de las preguntas y opciones. En el caso de las preguntas abiertas, éstas no se procesan, sólo se muestran.
- ❖ Reporte de alumnos. Muestra el número de cuenta y el nombre completo de los alumnos que contestaron todas sus encuestas, ordenados alfabéticamente por apellido paterno.

### 3.2.7. Reglas de negocio

Las reglas de negocio, son las condiciones de validación que se incluyen en los formularios (interfases), para que se puedan almacenar los datos o permitir el acceso. A continuación se presentan las condiciones para el sistema.

- ❖ Alumno.
  - Para ingresar al sistema, considerar:  
El número de cuenta debe ser de nueve dígitos, sin el guión que separa al dígito verificador.  
Los números de cuenta de generaciones anteriores al 2000, hay que agregar un cero al inicio.  
La fecha de nacimiento debe estar formada por dos dígitos para el día, dos dígitos para el mes y cuatro dígitos para el año.
  - Para contestar las encuestas y sean almacenadas las respuestas:  
Debe seleccionar el porcentaje de asistencia. Sí el porcentaje de asistencia es nunca, forzosamente tiene que explicar el porqué de dicho porcentaje y no contestar las preguntas de la encuesta, ya que no tiene los fundamentos para emitir su opinión.  
Sí el porcentaje de asistencia corresponde a cualquiera de las otras tres opciones, tiene que contestar todas las preguntas de opción múltiple.  
No es obligatorio contestar las preguntas abiertas. Sí se elige contestarlas, la respuesta debe ser de 200 caracteres como máximo.
- ❖ Administrativo.
  - Al ingresar a la sección de reportes, el semestre debe estar formado por cuatro dígitos para el año, el guión medio (-) y el semestre al que corresponde (1 ó 2). El total de dígitos son 6.
  - Para ingresar a la sección de nuevas encuestas, el nombre de usuario esta formado por 6 letras y la contraseña por 8 caracteres alfanuméricos.  
El nombre de la nueva encuesta debe tener como máximo 50 caracteres.  
El número de preguntas debe ser como máximo 20, tanto para opción múltiple como abiertas.  
El número de opciones máximo son 5.

Las preguntas deberán formularse con máximo 200 caracteres y las opciones con máximo 50 caracteres. Así como también las respuestas a las preguntas abiertas con 50 caracteres máximo.

❖ Administrador.

- Alta de alumnos. El número de cuenta de 9 dígitos sin el guión verificador y anteponiendo un cero al inicio del número a generaciones anteriores al 2000. El nombre empezando por apellido paterno, apellido materno y nombre(s). La fecha de nacimiento de 8 dígitos, dos para el día, dos para el mes y cuatro para el año. Sí es por archivo, debe tener extensión .csv y campos separados por punto y coma, el primer campo debe corresponder al número de cuenta, el segundo debe ser la fecha de nacimiento y el último campo debe ser el nombre completo. Los datos anteriores se forman con las características antes mencionadas.
- Alta de asesores. El RFC debe estar compuesto por 10 dígitos, 4 letras y 6 números. El nombre debe comenzar con el apellido paterno, después el apellido materno y por último nombre(s).  
Al dar de alta por archivo, éste tendrá extensión .csv y campos separados por punto y coma. El primer campo será el RFC y el segundo el nombre completo, estos con las características antes mencionadas.
- Alta de asignaturas. La clave de la asignatura estará formada por 4 dígitos numéricos.  
Alta de asignaturas por archivo, tendrá la extensión .csv y campos separados por punto y coma. El primer campo, clave de asignatura y el segundo el nombre de la asignatura.
- Alta de administrativo o administrador. Asignar nombre de usuario de máximo 9 caracteres alfanuméricos y contraseña de 8 caracteres alfanuméricos.
- Inscripciones. La clave de asignatura, el RFC del asesor, el número de cuenta y el semestre; tendrán el formato antes mencionado. El grupo serán 4 dígitos.  
Inscripción por archivo. El archivo tendrá la extensión .csv y campos separados por punto y coma. Deberán estar en el orden y con el formato antes mencionado.
- Sección bajas y cambios. Buscar y seleccionar sí se da de baja o se hace un cambio. En el caso de cambio, tomar en cuenta los criterios antes mencionados. Lo anterior para todos los rubros de esta sección.
- Sección depurar. El semestre estará formado por 6 dígitos, cuatro dígitos numéricos para el año, el guión medio (-) y el semestre 1 o 2, según corresponda. Después seleccionar encuestas y/o inscripciones.

La base de datos normalizada con las dependencias claramente marcadas; así como también la especificación de usuarios y los procesos que llevarán a cabo en el sistema (comportamiento), definición de clases, módulos y validaciones, son los resultados de las etapas de análisis, diseño y programación.

Teniendo lo anterior, es factible realizar la implementación del sistema propuesto.

# **CAPÍTULO 4**

## **IMPLEMENTACIÓN**

Identificado el comportamiento del sistema, a continuación se describe el mapa de navegación y el de archivos, las características del servidor, el plan de implementación y el diseño de pantallas que interactúan con el usuario de forma sencilla y amigable.

#### 4.1. Mapa de navegación

El mapa de navegación define cada una de las pantallas y la relación que existe entre ellas; a través de hipervínculos. Al igual que los organigramas de funciones, los mapas se construyen a partir de las secciones del sistema.

A través del mapa de navegación, se puede recorrer cada una de las pantallas del sistema de manera conceptual. El mapa tiene que ser lo suficientemente explicativo como para indicar claramente un posible trayecto hasta el objetivo que se persiga.

A continuación se muestra el mapa de navegación para el sistema, se divide en tres secciones:

- ❖ La primera sección muestra el mapa de navegación para contestar encuestas; enfocada principalmente a los alumnos.

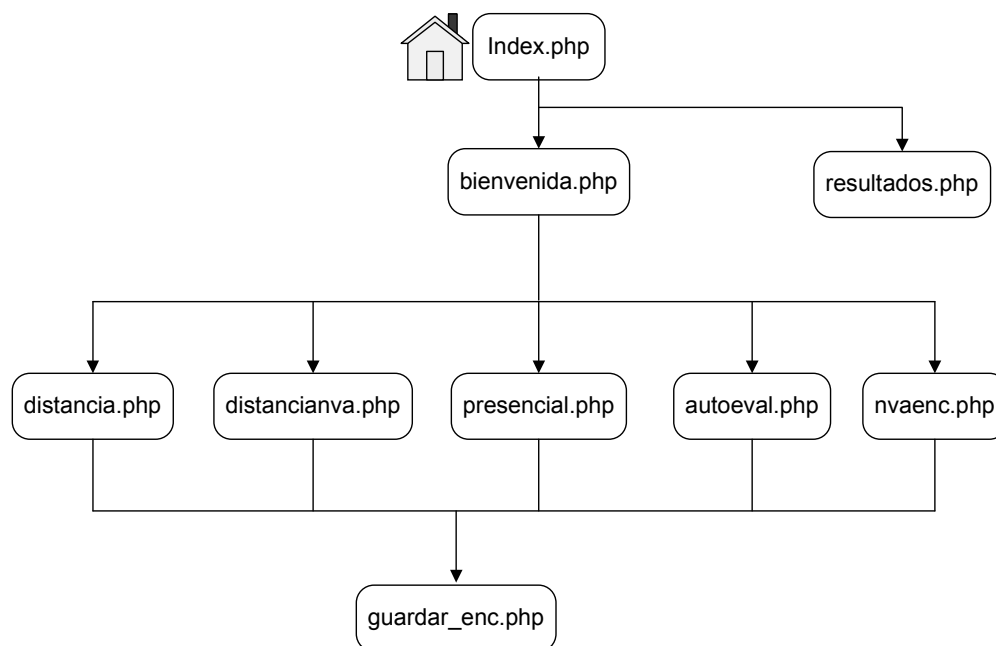


Figura 4.1. Mapa de navegación para contestar encuestas.

- ❖ La segunda sección se refiere a los reportes. El mapa está dirigido principalmente para los administrativos, ya que la legislación universitaria no permite que los alumnos conozcan los resultados de las evaluaciones.



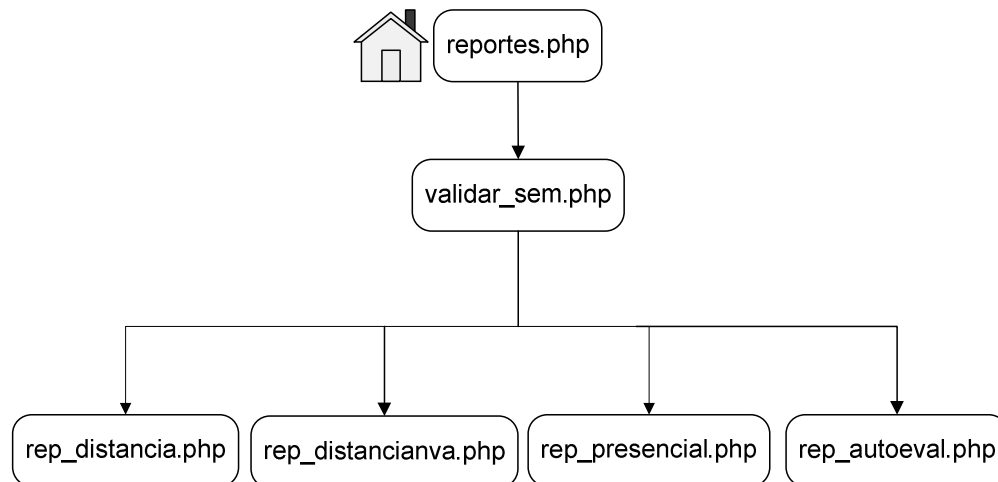


Figura 4.2. Mapa de navegación para reportes.

- ❖ La tercera sección despliega el orden de navegación para el administrador y otra parte para los administrativos. Básicamente el administrador proporciona mantenimiento al sistema y la parte del administrativo se refiere a la creación de nuevas encuestas.

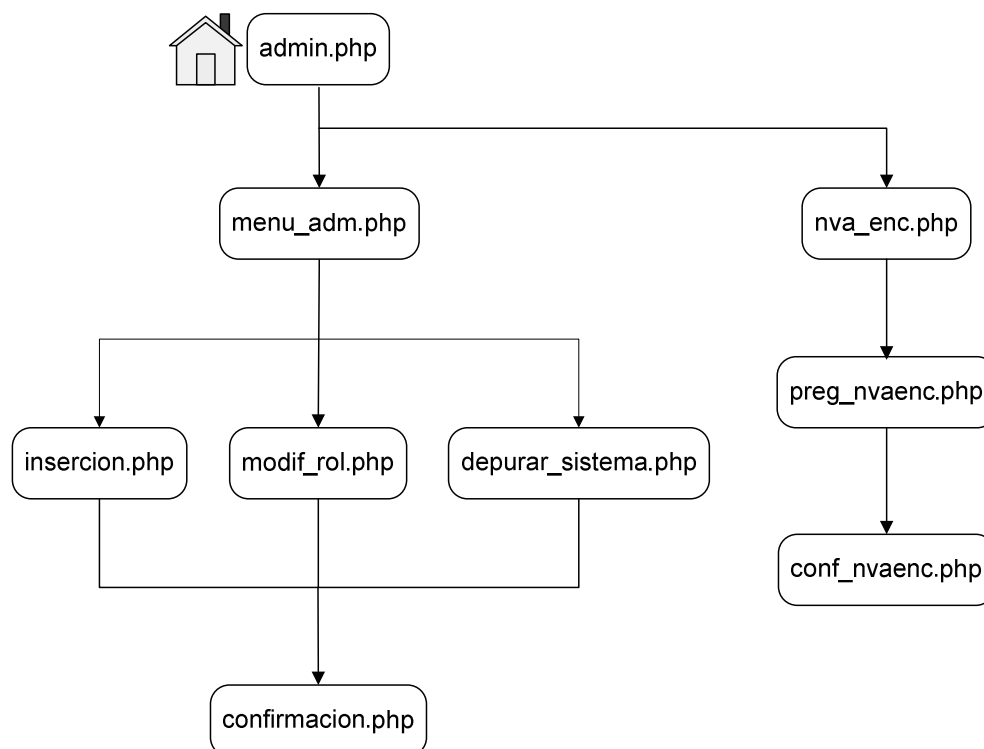


Figura 4.3. Mapa de navegación para administrador y administrativo.

## 4.2. Mapa de archivos

El mapa de archivos despliega la estructura y componentes del sistema. Es decir; nombre de los archivos, archivos o scripts de validación y hojas de estilo que conforman el sistema. En el mapa se observan diferentes figuras: los rectángulos con esquinas redondeadas son páginas y los rectángulos son scripts de validación. Al igual que en el mapa de navegación, el mapa de archivos se divide en tres secciones, las cuales corresponden a las mencionadas en el punto anterior.

- ❖ En la primera sección; el archivo principal es el Index.php, se adjunta un archivo de encabezado y una hoja de estilo, los cuales se utilizan en el todo el mapa.

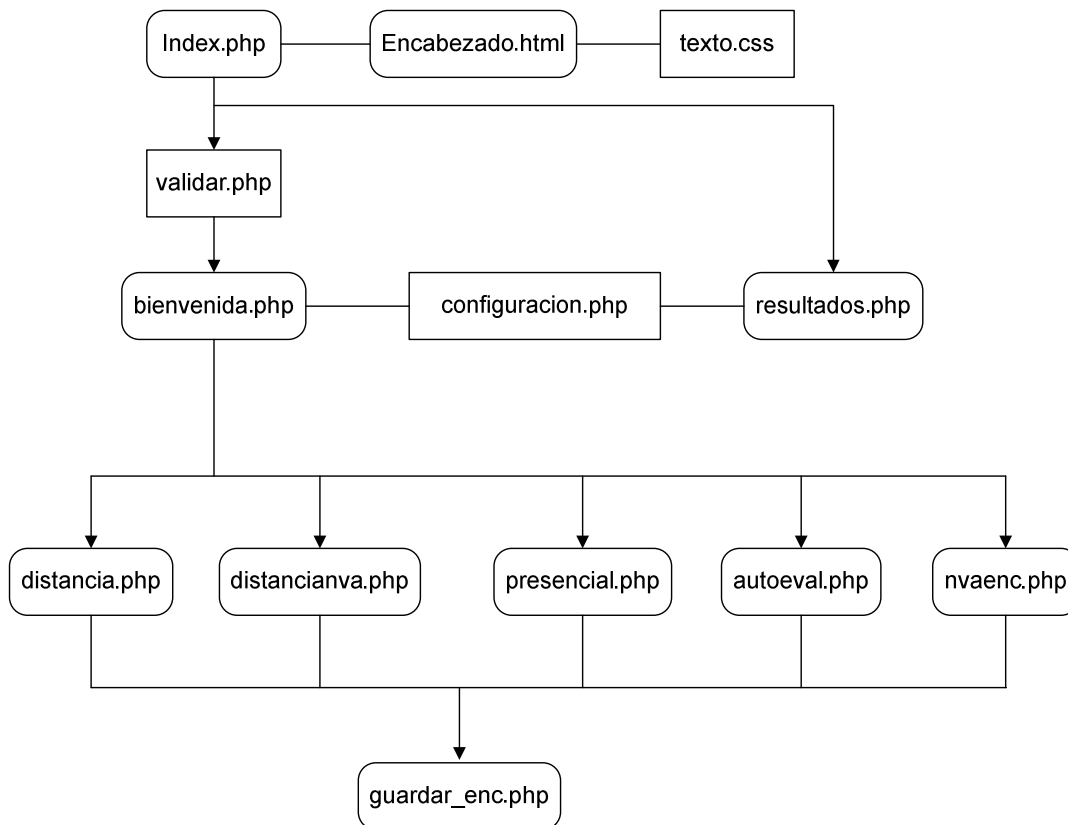


Figura 4.4. Mapa de archivos para contestar encuestas.

- ❖ En la segunda sección, el archivo principal es reportes.php, se adjunta el archivo de encabezado y la hoja de estilo, utilizados en todo el mapa.

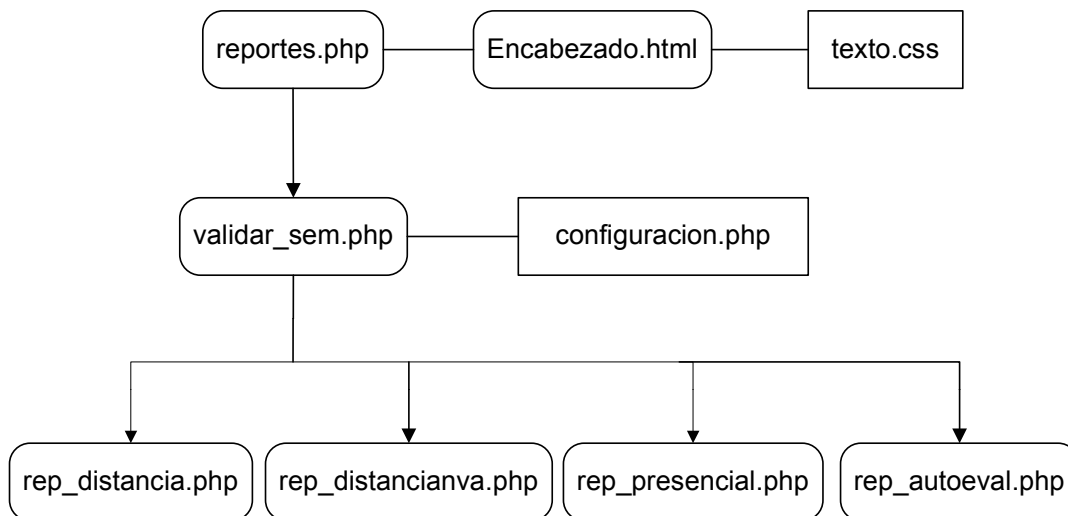


Figura 4.5. Mapa de archivos para reportes.

- ❖ En la tercera sección, admin.php es el archivo principal, se adjunta el archivo de encabezado y una hoja de estilo. Ambos archivos se utilizan en todo el mapa.

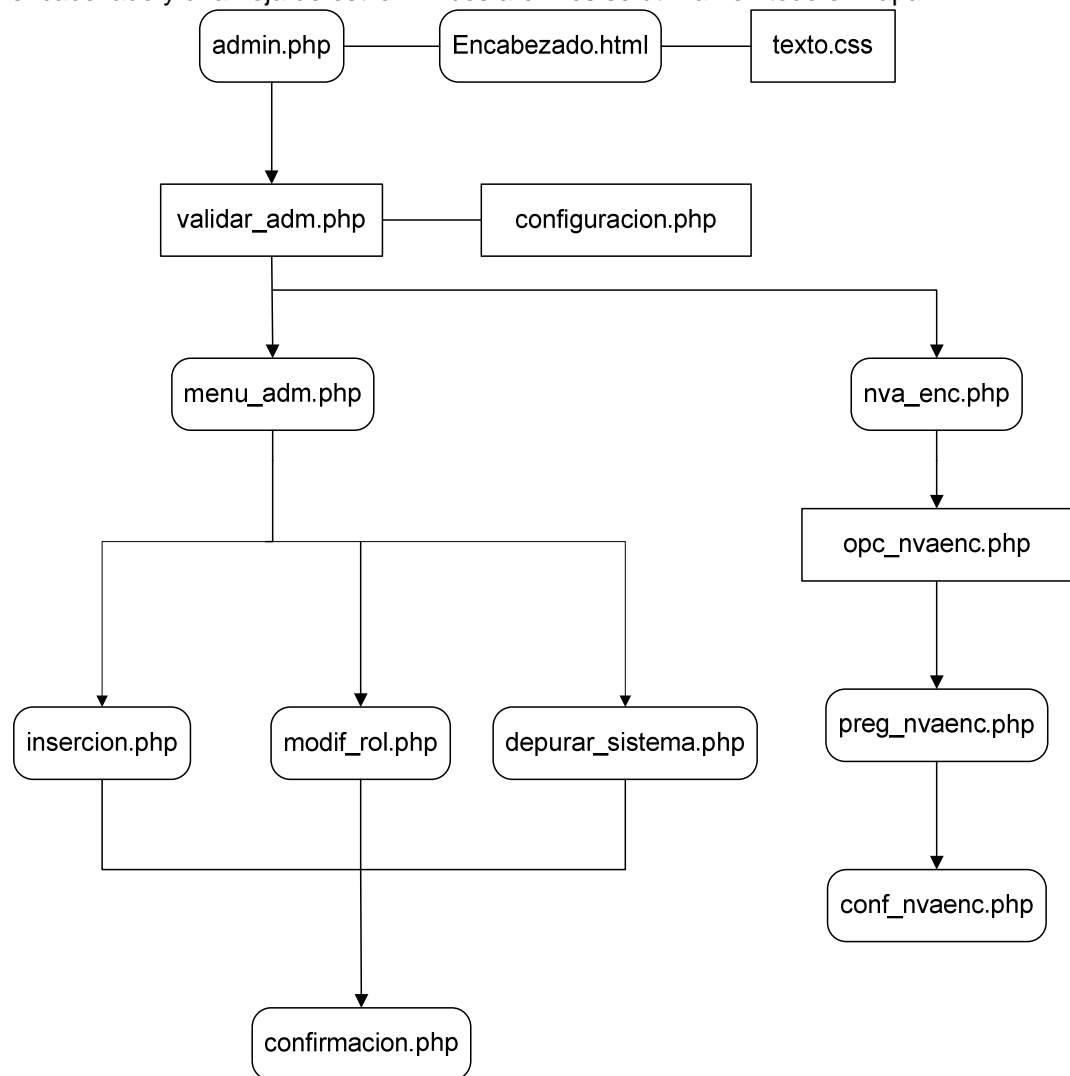


Figura 4.6. Mapa de archivos para administrador y administrativo.

### 4.3. Diseño de pantallas

El objetivo de este punto, es la especificación de las interfaces entre el sistema y el usuario: formatos de pantallas, diálogos e informes, principalmente. Se realiza un análisis de los procesos del sistema de información en los que se requiere una interacción del usuario, con el fin de crear una interfaz o pantalla que satisfaga todos los requisitos establecidos, teniendo en cuenta los diferentes perfiles a quienes va dirigido.

Al comienzo del análisis, es necesario seleccionar el entorno en el que es operativa la interfaz, considerando estándares en los procesos de diseño y construcción. El propósito es construir una interfaz de usuario acorde a las necesidades, flexible, coherente, eficiente y sencilla de utilizar; teniendo en cuenta la facilidad de cambio a otras plataformas, sí fuera necesario.

A continuación se presentan las pantallas que conforman el sistema:

- ❖ En la pantalla principal se solicita al usuario que escriba su número de cuenta y su fecha de nacimiento, con el objetivo de ingresar a las encuestas.

Figura 4.7. Pantalla principal.

- ❖ En la siguiente pantalla, se muestran las asignaturas, el asesor, el grupo y el estado; este ultimo se refiere a sí ya fue contestada o no. Además es el comprobante de que contestó las encuestas.

Asignatura	Asesor	Grupo	Estado
Contabilidad Gral. y de Costos	Quiroz García Mercedes	9213	Contestada
Historia Económica General I	Cordera Campos Rafael	9214	Contestada
Autoevaluación			Contestada

Figura 4.8. Muestra datos de las asignaturas inscritas.

- ❖ La interfaz para la encuesta de asesores a distancia semestre 2005-1, en la cual se presentan las preguntas y las opciones, se muestra en la figura 4.9:



## SISTEMA UNIVERSIDAD ABIERTA



**Asesores a distancia**  
**Encuesta sobre el desempeño del semestre 2005-1**

**Asignatura:** Economía Política I  
**Clave:** 0102  
**Grupo:** 9213  
**Asesor:** Moreno Suárez Agustín

Con el objeto de mejorar la calidad académica de nuestra Facultad, se realiza una evaluación al desempeño de cada asesor. Agradeceremos que conteste honestamente las preguntas, seleccionando sólo una opción.

La información que nos proporcione es anónima y muy importante, por lo que le pedimos que conteste todo el cuestionario sólo si tiene elementos suficientes para ello. Si no asistió a las asesorías marque la opción correspondiente y envíe el cuestionario.

¿En que porcentaje trabajó en el curso a distancia?

100% a 76%     
  De 75% a 50%     
  Menos de 50%     
  Nunca

Explique por qué:

- 1.- ¿El asesor se comunicó con usted alguna vez por correo electrónico?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 2.- ¿El asesor orienta acerca de cómo estudiar la asignatura?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 3.- ¿El asesor muestra dominio de la asignatura?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 4.- ¿La asesoría se desarrolla con base a las lecturas del curso?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 5.- ¿Relaciona su explicación con otras asignaturas del plan de estudios?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 6.- ¿El asesor estableció una buena comunicación para responder a dudas o ampliar el tema?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 7.- ¿El asesor responde oportunamente en forma precisa y explícita a dudas?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 8.- ¿El asesor genera actividades que estimulen la comunicación a través de correo electrónico?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 9.- ¿El asesor sugiere lecturas o actividades complementarias para mejorar la comprensión de textos?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 10.- ¿El asesor fomenta la participación con el planteamiento de problemas y la búsqueda de soluciones?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 11.- ¿El asesor explica a detalle la forma de evaluar?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 12.- ¿El asesor considera los temas y las actividades de aprendizaje para realizar las evaluaciones del curso?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 13.- ¿Las formas de evaluación son adecuadas para medir los conocimientos adquiridos?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 14.- ¿El asesor entrega oportunamente calificaciones?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre


En el siguiente espacio escriba de manera responsable, otros elementos sobre el desempeño del asesor que no hayan sido contemplados en la preguntas anteriores. Haga hincapié en las cosas que permitan realizar mejor nuestro trabajo.

1.- Mencione cuáles fueron los elementos valiosos que consideró del curso:


2.- Añote las sugerencias que mejoren el desempeño del asesor y del curso:

Figura 4.9. Encuesta de asesores a distancia 2005-1.

- ❖ La figura 4.10 es la interfaz para la encuesta de asesores a distancia, pero del semestre 2005-2 en adelante (preguntas y opciones).



SISTEMA UNIVERSIDAD ABIERTA



**Asesores a distancia**  
Encuesta sobre el desempeño del semestre 2006-1

**Asignatura:** Introducción a la Teoría Económica  
**Clave:** 0103  
**Grupo:** 9214  
**Asesor:** Ramírez Hernández Guillermo

Con el objeto de mejorar la calidad académica de nuestra Facultad, se realiza una evaluación al desempeño de cada asesor. Agradeceremos que conteste honestamente las preguntas, seleccionando sólo una opción.

La información que nos proporcione es anónima y muy importante, por lo que le pedimos que conteste todo el cuestionario sólo si tiene elementos suficientes para ello. Si no asistió a las asesorías marque la opción correspondiente y envíe el cuestionario.

¿En que porcentaje trabajo en el curso a distancia?

100% a 76%    
  De 75% a 50%    
  Menos de 50%    
  Nunca

Explique por qué:

**Responsabilidad**

1.- ¿El curso contó con la bibliografía montada en la plataforma, o existe paquete de aprendizaje?

Nunca    
  Ocasionalmente    
  Frecuentemente    
  Siempre

2.- ¿El asesor montó oportunamente las actividades a desarrollar en el curso?

Nunca    
  Ocasionalmente    
  Frecuentemente    
  Siempre

3.- ¿El asesor calificó los trabajos y exámenes en plazo razonable?

Nunca    
  Ocasionalmente    
  Frecuentemente    
  Siempre

4.- ¿El asesor dio respuesta oportuna a las preguntas y dudas planteadas en la plataforma?

Nunca    
  Ocasionalmente    
  Frecuentemente    
  Siempre

5.- En caso de que no estuviera montada la bibliografía en la plataforma ¿fueron claras las indicaciones de los textos para estudiar?

Nunca    
  Ocasionalmente    
  Frecuentemente    
  Siempre

**Evaluación**

6.- ¿Las actividades de evaluación reflejan el contenido del curso?

Nunca    
  Ocasionalmente    
  Frecuentemente    
  Siempre

7.- Las actividades a desarrollar en el curso fueron:

Muy cargadas    
  Poco cargadas    
  Adecuadas    
  Ligeras

8.- ¿Los comentarios a los exámenes y actividades del asesor permitieron identificar los errores?

Nunca    
  Ocasionalmente    
  Frecuentemente    
  Siempre

**Motivación**

9.- El interés mostrado por el asesor durante el curso fue:

Nulo       Escaso       Grande       Muy grande

10.- Si al asesor se le propone en otro curso ¿volvería a tomar con él?

No       Probablemente       Lo pensaría       Sí

**Aprendizaje**

11.- ¿El asesor orienta y apoya el estudio de la materia?

Nunca       Ocasionalmente       Frecuentemente       Siempre

12.- ¿Las actividades planteadas por el asesor contribuyen al aprendizaje?

Nunca       Ocasionalmente       Frecuentemente       Siempre

13.- ¿Ante algún tema difícil planteado por el grupo, el asesor proporcionó material adicional?

Nunca       Ocasionalmente       Frecuentemente       No fue el caso

**Autoevaluación del Alumno**

14.- ¿Cuándo se incorporó al curso a distancia?

Muy tarde       Al mes       Primeras semanas       Al inicio

15.- ¿Ingresaba a la plataforma?

Ocasionalmente       Una vez por semana       Casi diario       Diario

16.- ¿Las actividades y trabajos los entregó a tiempo?

Nunca       Ocasionalmente       Casi siempre       Siempre

17.- Evalúe la Plataforma SUAFE:

Deficiente       Regular       Buena       Muy buena

18.- ¿Participó en los mensajes y foros planteando dudas y haciendo contribuciones?

Nunca       Algunas veces       Con frecuencia       Siempre

19.- ¿Piensa tomar otros cursos a distancia?

No       Depende del profesor       Si no hay de otra       Sí

En el siguiente espacio escriba de manera responsable, otros elementos sobre el desempeño del asesor que no hayan sido contemplados en la preguntas anteriores. Haga hincapié en las cosas que permitan realizar mejor nuestro trabajo.

1.- Comentarios sobre el asesor:

2.- Comentarios sobre la plataforma y el curso (materiales, actividades y otros):

Figura 4.10. Encuesta de asesores a distancia 2005-2 y posteriores.

❖ En la figura 4.11 se despliegan las preguntas de la encuesta de asesores presenciales.



## SISTEMA UNIVERSIDAD ABIERTA



**Asesores presenciales**  
**Encuesta sobre el desempeño del semestre 2006-1**

<b>Asignatura:</b>	Economía Política III
<b>Clave:</b>	0302
<b>Grupo:</b>	9231
<b>Asesor:</b>	Lizarraga Gómez Antonio

Con el objeto de mejorar la calidad académica de nuestra Facultad, se realiza una evaluación al desempeño de cada asesor. Agradeceremos que conteste honestamente las preguntas, seleccionando sólo una opción.

La información que nos proporcione es anónima y muy importante, por lo que le pedimos que conteste todo el cuestionario sólo si tiene elementos suficientes para ello. Si no asistió a las asesorías marque la opción correspondiente y envíe el cuestionario.

¿A qué porcentaje de asesorías presenciales asistió?

100% a 76%     
  De 75% a 50%     
  Menos de 50%     
  Nunca

Explique por qué:

- 1.- ¿El asesor asiste a las asesorías grupales programadas?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 2.- ¿El asesor se presenta puntualmente en horarios establecidos para la asesoría grupal?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 3.- ¿El asesor muestra dominio de la asignatura?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 4.- ¿La asesoría se desarrolla sin perder el punto central del tema?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 5.- ¿Relaciona su explicación con otras asignaturas del plan de estudios?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 6.- ¿El asesor utiliza el material didáctico diseñado para la asignatura?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 7.- ¿Utiliza el asesor el material didáctico para orientar su trabajo durante el curso?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 8.- ¿El asesor se muestra accesible para atender dudas, a través de comunicación verbal o escrita fuera de la asesoría grupal?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 9.- ¿El asesor dedica el tiempo necesario para resolver las dudas en su totalidad?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 10.- ¿El asesor responde de manera clara y precisa a las dudas que le plantean?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 11.- ¿El asesor realiza actividades que estimulan la participación del grupo?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 12.- ¿El asesor lleva a cabo actividades de aprendizaje y ejercicios que refuercen los contenidos de la asesoría?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 13.- ¿El asesor resume o da conclusiones al término de cada sesión de asesoría grupal?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 14.- ¿El asesor recomienda lecturas o actividades complementarias al material?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 15.- ¿El asesor explica claramente la forma de evaluar?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 16.- ¿El asesor se basa en el programa de la asignatura para realizar sus evaluaciones?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre
- 17.- ¿Considera que las formas de evaluación son adecuadas para medir los conocimientos adquiridos?
 

Nunca   
  Ocasionalmente   
  Algunas veces   
  Frecuentemente   
  Siempre



18.- ¿El asesor entrega oportunamente calificaciones?

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

19.- ¿Se inscribiría o recomendaría al asesor en otros cursos?

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

En el siguiente espacio escriba de manera responsable, otros elementos sobre el desempeño del asesor que no hayan sido contemplados en la preguntas anteriores. Haga hincapié en las cosas que permitan realizar mejor nuestro trabajo.


1.- Mencione cuáles fueron los elementos valiosos que consideró del curso.

2.- Anote las sugerencias que mejoren el desempeño del asesor y del curso.


 

Figura 4.11. Encuesta de asesores presenciales.

❖ En esta pantalla, se presentan las preguntas de la encuesta de autoevaluación.



SISTEMA UNIVERSIDAD ABIERTA



**Autoevaluación**  
**Encuesta sobre el desempeño del semestre 2006-1**

Con el objeto de mejorar la calidad académica de nuestra Facultad, periódicamente se realiza una evaluación al desempeño de los alumnos. Agradeceremos que conteste honestamente las preguntas, seleccionando sólo una opción.

La información que nos proporcione es anónima y muy importante, por lo que le pedimos conteste todo el cuestionario.

1.- Considero las características de las modalidades (abierta y a distancia), antes de inscribirme a las asignaturas.

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

2.- Me informo adecuadamente de los contenidos de las asignaturas.

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

3.- Realizo las actividades de aprendizaje y preparo mis lecturas previo a la asesoría.

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

4.- Organizo mi tiempo para preparar los temas de la siguiente asesoría.

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

5.- Participo en la asesoría de grupo, expresando mis dudas con respecto a la lectura (o por escrito en la modalidad a distancia).

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

6.- Participo en la asesoría de grupo aportando elementos comprendidos (o participando en las listas de distribución en la modalidad a distancia).

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

7.- Asisto a las asesorías grupales (presenciales o a distancia) programadas para la asignatura.

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

8.- Establezco comunicación verbal o escrita con el asesor (o con mis compañeros) sobre dudas o para exponer mi comprensión del tema.

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

9.- Solicito asesorías individuales con el asesor titular o con otros profesores.

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

10.- El aprendizaje obtenido con las asesorías y mi estudio independiente, son las adecuadas para presentar una evaluación final.

Nunca    Ocasionalmente    Algunas veces    Frecuentemente    Siempre

A continuación responda estas tres preguntas para que exprese algunos elementos que no fueron contemplados en las anteriores. Esta información permitirá identificar los problemas y las formas de implementar acciones para mejorar los servicios educativos.

1.- Mencione las características valiosas que permitirán un mejor desempeño a los alumnos SUA – FE.

2.- ¿Qué considera que no deberían realizar los alumnos del SUA?

3.- ¿Qué propone a la División del SUA – FE sobre las actividades en que, de manera conjunta, se podrían atender ambos elementos anteriores?

Figura 4.12. Encuesta de autoevaluación.

- ❖ La siguiente pantalla indica si las respuestas fueron almacenadas o no, se despliega después de que contestó la encuesta y oprimió el botón Enviar.

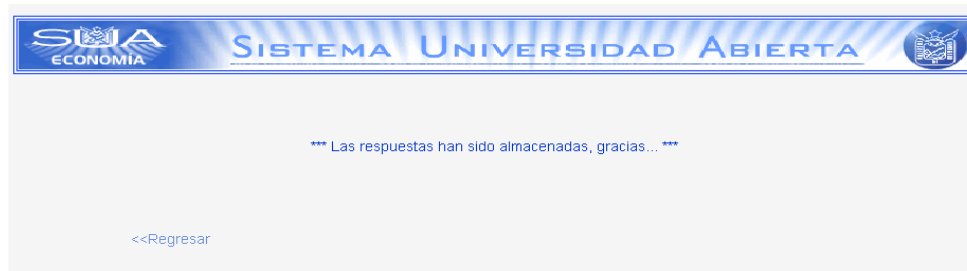


Figura 4.13. Pantalla de confirmación.

- ❖ A continuación la interfaz principal para los diferentes reportes, la cual solicita el semestre que se quiere consultar.

Figura 4.14. Pantalla principal de reportes.

- ❖ La pantalla muestra una lista con el nombre de la asignatura, nombre del asesor, grupo y tipo de curso (distancia o presencial). Además del hipervínculo al reporte de autoevaluación. Sólo se presenta el inicio y fin de la interfaz.

Semestre 2006-1			
Asignatura	Asesor	Grupo	Tipo
Contabilidad Gral. y de Costos	Alcántara Barrera Juan Carlos	9214	Distancia
Introducción a la Teoría Económica	Anaya Díaz Alfonso Miguel	9213	Distancia
Introducción a la Teoría Económica	Anaya Díaz Alfonso Miguel	9215	Presencial
Economía Industrial	Anaya Díaz Alfonso Miguel	9252	Presencial
Taller de Economía Cuantitativa I	Alcántara Guerrero Cristina	9213	Distancia
Taller de Economía Cuantitativa I	Alcántara Guerrero Cristina	9214	Distancia
Teoría Microeconómica I	Acevedo Fernández Ernesto	9222	Presencial
Introducción a la Econometría	Acevedo Fernández Ernesto	9251	Presencial
Econometría I	Acevedo Fernández Ernesto	9271	Presencial
Política Social en México	Ávila Martínez Ángel	9291	Presencial
Investigación y Análisis Económico III	Vargas Cruz Natividad	9231	Presencial
Teoría Microeconómica I	Vega Yáñez Alejandro	9221	Presencial
Teoría Macroeconómica I	Vega Yáñez Alejandro	9241	Presencial
Teoría Microeconómica II	Vega Yáñez Alejandro	9231	Presencial
Contabilidad Social	Villegas Ramírez Ma. Clemencia	9221	Presencial
Autoevaluación			

<<Salir

Figura 4.15. Asignaturas impartidas en el semestre especificado.

- ❖ Muestra el nombre y clave de la asignatura, el grupo, el nombre del asesor, el número de alumnos y el porcentaje de alumnos que contestaron la encuesta de asesores a distancia (2005-1). Así como también el puntaje obtenido y el máximo de cada pregunta, el puntaje global y las opiniones de los alumnos.

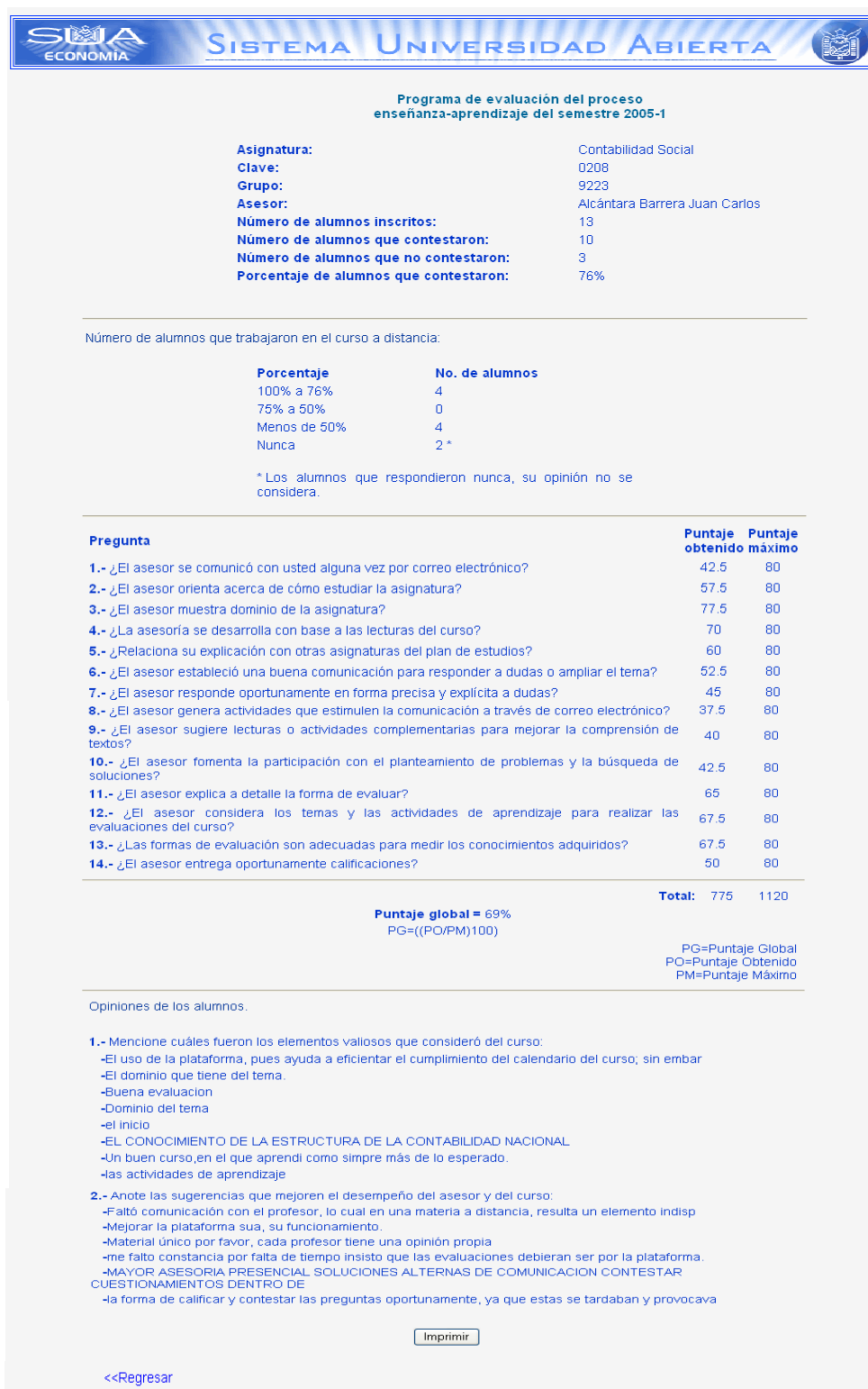




Figura 4.16. Reporte de asesores a distancia, semestre 2005-1.

- ❖ El reporte de asesores a distancia (2005-2 y posteriores); contiene el nombre y clave de la asignatura, el grupo, el nombre del asesor, número y porcentaje de alumnos que contestaron la encuesta. Este tipo de reporte está dividido en secciones, las cuales también tienen un puntaje obtenido y puntaje máximo para cada pregunta, al final se

obtiene un puntaje global. También se enlistan los comentarios de los alumnos, lo anterior se muestra en la siguiente figura.



## SISTEMA UNIVERSIDAD ABIERTA



**Programa de evaluación del proceso enseñanza-aprendizaje del semestre 2006-1**

<b>Asignatura:</b>	Introducción a la Teoría Económica
<b>Clave:</b>	0103
<b>Grupo:</b>	9213
<b>Asesor:</b>	Anaya Díaz Alfonso Miguel
<b>Número de alumnos inscritos:</b>	15
<b>Número de alumnos que contestaron:</b>	8
<b>Número de alumnos que no contestaron:</b>	7
<b>Porcentaje de alumnos que contestaron:</b>	53%

---

Número de alumnos que trabajaron en el curso a distancia:

Porcentaje	No. de alumnos
100% a 76%	5
75% a 50%	1
Menos de 50%	2
Nunca	0 *

\* Los alumnos que respondieron nunca, su opinión no se considera.

---

		Puntaje obtenido	Puntaje máximo
<b>*Responsabilidad</b>			
1.- ¿El curso contó con la bibliografía montada en la plataforma, o existe paquete de aprendizaje?	49	80	
2.- ¿El asesor montó oportunamente las actividades a desarrollar en el curso?	49	80	
3.- ¿El asesor calificó los trabajos y exámenes en plazo razonable?	41	80	
4.- ¿El asesor dio respuesta oportuna a las preguntas y dudas planteadas en la plataforma?	41	80	
5.- En caso de que no estuviera montada la bibliografía en la plataforma ¿fueron claras las indicaciones de los textos para estudiar?	49	80	
	Total:	229	400
		Total:	(229*15)/400
		<b>Total:</b>	<b>8%</b>
<b>*Evaluación</b>			
6.- ¿Las actividades de evaluación reflejan el contenido del curso?	55	80	
7.- Las actividades a desarrollar en el curso fueron:	36	80	
8.- ¿Los comentarios a los exámenes y actividades del asesor permitieron identificar los errores?	35	80	
	Total:	126	240
		Total:	(126*16)/240
		<b>Total:</b>	<b>8%</b>
<b>*Motivación</b>			
9.- El interés mostrado por el asesor durante el curso fue:	45	80	
10.- Si al asesor se le propone en otro curso ¿volvería a tomar con él?	55	80	
	Total:	100	160
		Total:	(100*32)/160
		<b>Total:</b>	<b>20%</b>
<b>*Aprendizaje</b>			
11.- ¿El asesor orienta y apoya el estudio de la materia?	48	80	
12.- ¿Las actividades planteadas por el asesor contribuyen al aprendizaje?	51	80	
13.- ¿Ante algún tema difícil planteado por el grupo, el asesor proporcionó material adicional?	41	80	
	Total:	140	240
		Total:	(140*37)/240
		<b>Total:</b>	<b>21%</b>
Puntaje global= 8%+8%+20%+21% <b>Puntaje global = 57%</b>			
		Puntaje obtenido	Puntaje máximo
<b>*Autoevaluación del Alumno</b>			
1.- ¿Cuándo se incorporó al curso a distancia?	56	80	
2.- ¿Ingresaba a la plataforma?	40	80	
3.- ¿Las actividades y trabajos los entregó a tiempo?	37	80	
4.- ¿Participó en los mensajes y foros planteando dudas y haciendo contribuciones?	25	80	
	Total:	158	320
		Puntaje total = (158*100)/320	
		<b>Puntaje total = 49%</b>	
Opiniones de los alumnos.			
1.- Comentarios sobre el asesor:			
-Muy buena idea de pasar las sesiones teóricas por videoconferencia. Se podría mejorar todavía mas la interacción, usando el chat que se abrió en cada sesión.			
-Es el único del SUAFE que he conocido que tiene bien estructurado su plan de trabajo, con fechas exactas y horarios bien establecidos.			

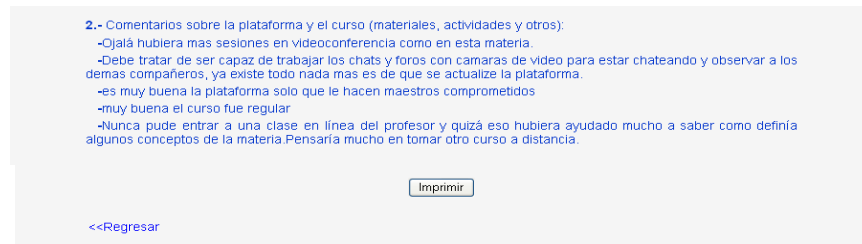


Figura 4.17. Reporte de asesor a distancia, semestre 2005-2 y posteriores.

- ❖ La pantalla contiene el nombre y clave de la asignatura, grupo, nombre del asesor, número y porcentaje de alumnos que contestaron la encuesta de asesores presenciales. También el puntaje obtenido y máximo de cada pregunta, para obtener el porcentaje global; y al último los comentarios de los alumnos, la figura 4.18 muestra lo anterior:

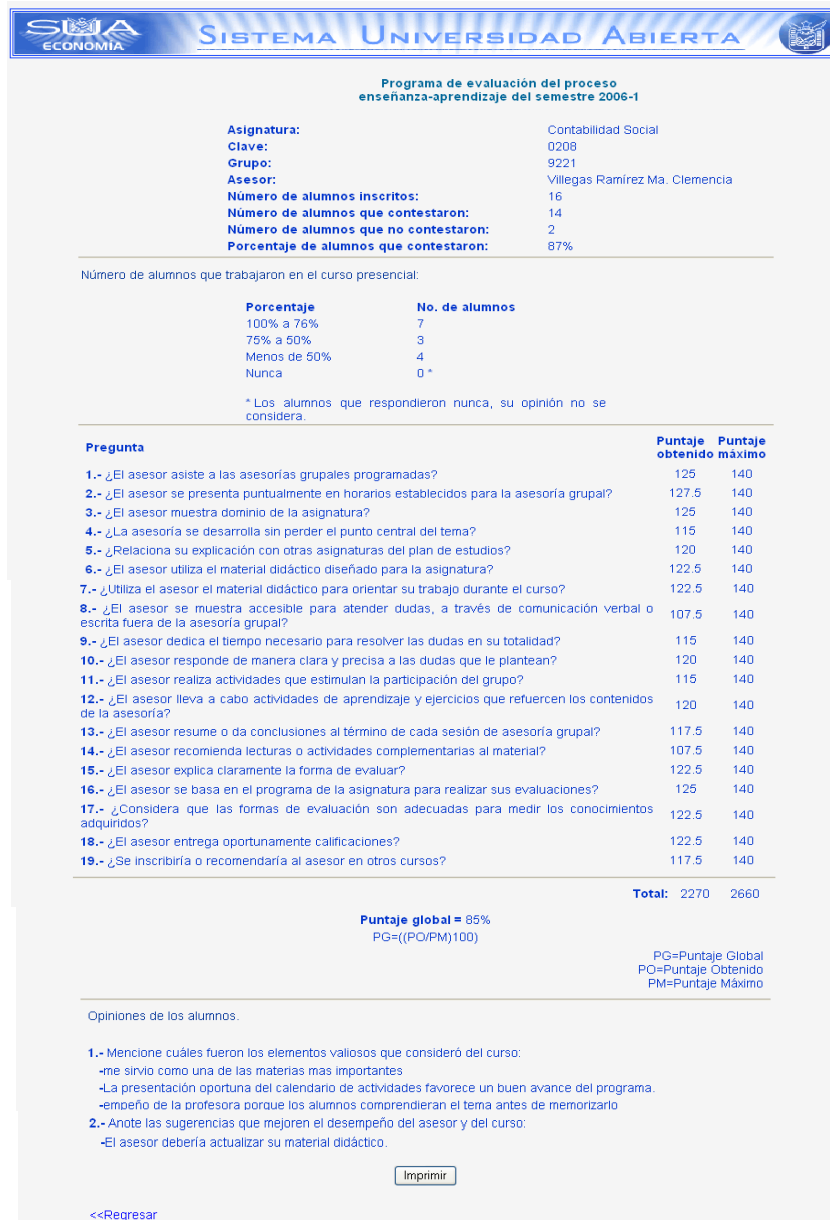


Figura 4.18. Reporte de asesor presencial.

- ❖ El reporte de la encuesta de autoevaluación, despliega el número de alumnos inscritos y el número de alumnos que contestaron. En cada pregunta; se muestra el número y porcentaje de alumnos que eligieron la opción. Al final se enlistan las respuestas a las preguntas abiertas. En la interfaz siguiente no se muestran todas las respuestas.



Figura 4.19. Reporte de autoevaluación.

- ❖ En la interfaz que muestra la figura 4.20, se enlistan los alumnos que contestaron todas sus encuestas, se muestra el número de cuenta y el nombre, al final el total de alumnos que contestaron. Aquí no se despliegan todos los alumnos (inicio y fin de la lista).

Alumnos que contestaron encuestas	
406015297	ACOSTA MENA AARON ALI
406036854	AGUILAR DEL ROSARIO MARICELA
081293661	AGUILAR PACHECO ALFREDO
405103380	AGUILAR PICAZO SARA
099533926	AGUILERA RODRIGUEZ LUIS
405057977	AGUIRRE HERNÁNDEZ MISHEL PAOLA
099541592	AHEDO GARCIA MARIO RICARDO
402044527	ALANIZ ESTRADA LUIS ALBERTO
406041669	ALPIZAR MEJIA DIEGO
097125923	ALTAMIRANO JIMENEZ JESUS ARMANDO
403110805	ALVA MARTINEZ ALEJANDRO FEDERICO
401017973	ALVAREZ BARRON JOSE LUIS
098066072	ALVAREZ PAZ CESAR
402055242	ANDRADE CRUZ MA DEL ROSARIO VERO
406019491	ANGULO LARA ALBERTO TOMAS
400072791	ANZUREZ RODRIGUEZ JESUS
403045620	AQUINO ILLESCAS JUAN CARLOS
402118592	ARCOS OLVERA JORAM PABLO
095321624	ARELLANO CORTEZ JOSE MANUEL
093308102	ARJONA NUÑEZ HECTOR
406032034	AVILES EUSEBIO DAVID
090364804	AYALA BERRA ADRIAN FEDERICO
401038103	AZAMAR ROMERO ALEJANDRO
097172587	BAEZ DIAZ IVAN ALONSO
406009948	BALTAZAR LEON JOSUE NEFTALI
406049263	BALTIERRA URIBE CHRISTIAN DIEGO
404048387	VALDES SALMON MARIA DEL PILAR
404056513	VALENCIA JIMENEZ MARCO ANTONIO
088165866	VARGAS GRAJALES CITLALI ALINA
402027230	VASCONCELOS BARRERA SILVIA
401007602	VAZQUEZ TORRES BEATRIZ
093325334	VEGA LOPEZ CARLOS
097201438	VELAZQUEZ RUBIO SUSAN LILLIAN
075416038	VILLEGAS HERNANDEZ DANIEL
406037936	YLESCAS MENDOZA HECTOR
400039262	ZARATE MARTINEZ SALVADOR
402119991	ZARATE ROJAS RICARDA KARINA
098096068	ZUBIGA MARTINEZ JORGE

Total de alumnos que contestaron: 232

Imprimir

Figura 4.20. Reporte de alumnos que contestaron encuestas.

- ❖ Pantalla principal para el administrador y el administrativo, en la cual se solicita el nombre de usuario y la contraseña. También se puede acceder a las encuestas y reportes, por medio de un hipervínculo.

Administración

Acceso a encuestas

Reportes

Nombre de usuario:

Contraseña:

Entrar

**SISTEMA DE EVALUACIÓN DOCENTE**

UNAM. Facultad de Economía. División Sistema Universidad Abierta.

Este sitio se ve mejor a 1024 x 768  
2004 D.R. © UNAM

Figura 4.21. Pantalla principal para administrador y administrativo.



- ❖ La pantalla del administrador se divide en tres secciones; altas, bajas y cambios, y depurar.



Figura 4.22. Secciones para el administrador.

- ❖ La pantalla (Figura 4.23) cambia dependiendo de la acción a realizar; las acciones pueden ser las mostradas en la interfaz anterior (sección altas); en este caso se presenta la de alta de alumno. Tiene dos opciones, agregar un sólo registro con los datos solicitados, o agregar más de uno por medio de un archivo.



Figura 4.23. Alta alumno.

- ❖ Esta pantalla presenta diferentes características para realizar bajas o cambios, debido a la opción elegida. Las opciones para hacer cambios o bajas del sistema, son: alumno, asesor, asignatura, administrativo o administrador e inscripciones. Se muestra la interfaz para la opción de inscripciones, en la cual se puede buscar el registro o si se conocen todos los datos, hacer el cambio o modificación solicitado.

Figura 4.24. Baja o cambio de inscripción del alumno.

- ❖ Interfaz para eliminar los registros almacenados de los diferentes tipos de encuestas, así como los registros de inscripción; dependiendo del semestre elegido.

Figura 4.25. Eliminar registros.

- ❖ Pantalla que indica si las peticiones del administrador fueron realizadas correctamente.

Figura 4.26. Pantalla de confirmación.

- ❖ La figura 4.27 es la interfaz utilizada por el administrativo; en la cual se solicita el nombre para la nueva encuesta, número de preguntas y tipo. Así como también presenta una lista con las encuestas almacenadas, por sí se requiere modificar alguna.

Figura 4.27. Acciones del administrativo.

- ❖ Se muestra la siguiente pantalla para crear una nueva encuesta, con campos para escribir las preguntas y sus opciones. El número de preguntas y opciones, si es el caso, se especifica en la interfaz anterior.

Figura 4.28. Nueva encuesta.

- ❖ Interfaz que indica si la encuesta fue creada satisfactoriamente.



Figura 4.29. Pantalla de confirmación.

#### 4.4. Características del servidor

La base de datos, las interfaces y todos los demás componentes, se encuentran montados en un servidor con las siguientes características:

- ❖ Velocidad del procesador: Intel® Pentium IV 2.40GHz
- ❖ Disco duro: 60 GB
- ❖ Memoria RAM: 512 Mb
- ❖ Dirección IP: 132.248.167.103
- ❖ Nombre del servidor: sua.economia.unam.mx
- ❖ Sistema operativo: Red Hat Linux 3.3.3-7
- ❖ Manejador de base de datos: MySQL versión 3.23.58
- ❖ Apache 2.0.5.1 (Fedora)
- ❖ PHP versión 4.3.10

En el servidor se encuentra además el sitio web de la división sistema universidad abierta y la plataforma de educación en línea.

#### 4.5. Plan de implementación

Durante esta fase, el sistema se emplea de manera experimental para asegurarse de que no falle; es decir, que funcione de acuerdo a las especificaciones y en la forma en que los usuarios (alumnos y administrativos) esperan que lo haga.

Se combinan por primera vez todo el equipo y los procedimientos, con el objetivo de empezar a operar el nuevo sistema con transacciones reales, a fin de efectuar los ajustes que sean necesarios. Para ello, se alimentan como entradas conjuntos de datos de prueba para su procesamiento y después se examinan los reportes, para asegurarse que es lo que realmente se requiere.

Para observar si los usuarios tratan de utilizar el sistema en alguna forma no prevista, se permite a algunos alumnos y administrativos que utilicen el sistema, con el objetivo de corregir los puntos débiles antes de implementarlo totalmente.

En las siguientes tablas, se muestran algunas pruebas realizadas:

Prueba	Actividad
<b>Ingreso al sistema</b>	<ul style="list-style-type: none"> <li>❖ Número de cuenta con guión, acceso denegado.</li> <li>❖ Generación anterior al 2000 sin cero al inicio del número de cuenta, acceso denegado.</li> <li>❖ Fecha de nacimiento menor a ocho dígitos, acceso denegado.</li> <li>❖ Formato de fecha de nacimiento diferente al especificado, acceso denegado.</li> <li>❖ Número de cuenta y fecha de nacimiento con letras, acceso denegado.</li> <li>❖ Número de cuenta de nueve dígitos contemplando las condiciones anteriores, acceso permitido.</li> <li>❖ Fecha de nacimiento de ocho dígitos y respetando el formato, acceso permitido.</li> </ul>
<b>Encuestas</b>	<ul style="list-style-type: none"> <li>❖ Omitir selección de porcentaje de asistencia, no envía respuestas.</li> <li>❖ Seleccionar en porcentaje de asistencia la opción nunca y no especificar el porqué, no envía las respuestas.</li> <li>❖ No seleccionar opción en alguna pregunta, no envía las respuestas.</li> <li>❖ Respuesta a las preguntas abiertas con número de caracteres mayor a 200, el número de caracteres debe ser menor a 200.</li> </ul>

*Tabla 4.1. Sección alumno.*

Prueba	Actividad
<b>Ingreso al sistema</b>	<ul style="list-style-type: none"> <li>❖ El semestre tiene longitud menor a seis dígitos, acceso denegado.</li> <li>❖ El nombre de usuario es menor a seis letras, acceso denegado.</li> <li>❖ La contraseña es menor a ocho caracteres alfanuméricos, acceso denegado.</li> </ul>
<b>Nueva encuesta</b>	<ul style="list-style-type: none"> <li>❖ Nombre de la encuesta de cincuenta y dos caracteres, mensaje: el número de caracteres máximo es de cincuenta.</li> <li>❖ Número de preguntas mayor a treinta, mensaje: el número de preguntas máximo es de veinte.</li> <li>❖ Ocho opciones para la pregunta, mensaje: el número de opciones como máximo son cinco por pregunta.</li> </ul>

*Tabla 4.2. Sección administrativo.*

Prueba	Actividad
<b>Ingreso al sistema</b>	<ul style="list-style-type: none"> <li>❖ El nombre de usuario es menor a seis letras, acceso denegado.</li> <li>❖ La contraseña es menor a ocho caracteres alfanuméricos, acceso denegado.</li> </ul>
<b>Alta</b>	<ul style="list-style-type: none"> <li>❖ Número de cuenta del alumno con guión y dígito verificador, no almacena el dato.</li> <li>❖ Fecha de nacimiento sin atender el formato, no almacena el dato.</li> <li>❖ Almacenar datos desde archivo con campos separados por coma, no almacena los datos.</li> <li>❖ RFC del asesor con homoclave, no almacena el dato.</li> <li>❖ Nombre completo del asesor sin respetar formato, no almacena el dato.</li> <li>❖ Inscripción sin especificar semestre, no almacena el dato.</li> </ul>

*Tabla 4.3. Sección administrador.*

Para asegurar la operación adecuada del sistema hay que evaluarlo, con el objetivo de identificar puntos débiles y fuertes; realizando una valoración de la forma en que trabaja el sistema, esto incluye su facilidad de uso, tiempos de respuesta, nivel de utilización y revisión del formato de reportes. También hay que identificar y evaluar los beneficios para la división.

Se realizan los procedimientos necesarios para la implementación total y puesta en marcha del sistema. Dichos procedimientos son los siguientes:

- ❖ Eliminar toda la información utilizada en las pruebas.
- ❖ Almacenar en el sistema los datos de los alumnos.
- ❖ Almacenar en el sistema los datos de las asignaturas.
- ❖ Almacenar en el sistema los datos de los asesores.
- ❖ Almacenar en el sistema los datos de inscripción.
- ❖ Capacitar al personal que utilizará el sistema, administrativos y encargados de inscripción.

El mantenimiento es una actividad que no se contempla en la metodología, sino en los sistemas en producción.

A continuación se presenta el cronograma de desarrollo del proyecto:

Actividades / Semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
<b>Documentación</b>	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
<b>Análisis</b>																				
* Identificación de necesidades	■																			
* Encuestas/Entrevistas	■																			
* Requerimientos del sistema		■																		
* Estudio de factibilidad		■																		
* Conclusiones		■																		
<b>Diseño y Programación</b>																				
* Diseño de la base de datos			■																	
* Diagramas			■	■																
* Especificación de módulos			■	■																
* Corrección			■	■																
<b>Implementación</b>																				
* Diseño de pantallas						■	■													
* Mapa de navegación							■													
* Mapa de archivos								■												
* Instalación del servidor (S.O.)									■	■										
* Instalación de software										■	■	■	■	■	■	■	■	■	■	■
* Programación											■	■	■	■	■	■	■	■	■	■
* Pruebas																	■	■	■	■
* Modificación																		■	■	■
* Capacitación																				■

Tabla 4.4. Cronograma.

Organizados los archivos y después de haber descrito el mapa del sistema, además de tener las interfaces y los diferentes módulos codificados; se realizan las pruebas y se hacen los cambios necesarios. Al concluir lo anterior, se entrega el sistema para ser puesto en marcha en un ambiente de trabajo real, evaluando la funcionalidad y nivel de aceptación de los usuarios.

# **CONCLUSIONES**



El principal objetivo del proyecto se cumplió de forma satisfactoria, ya que la solución planteada y llevada a cabo en base al estudio preliminar, cubrió las expectativas y requerimientos de los usuarios.

El sistema fue puesto a prueba y utilizado por algunos alumnos y administrativos, los resultados obtenidos fueron óptimos y significativos.

El sistema solucionó uno de los principales problemas que tenía la División; ahora el número de encuestas contestadas aumento de forma significativa de un 35% a un 99%, con lo que se ha generado confianza en los alumnos para poder expresar sus comentarios referentes a los asesores, el programa de estudio, funcionamiento de la División e instalaciones.

Así como también los alumnos que radican fuera del Distrito Federal han podido contestar las encuestas sin tener que desplazarse y con la seguridad de que se tomará en cuenta su opinión.

Los asesores obtienen los reportes correspondientes automáticamente después de que las encuestas han sido cerradas. Los puntajes obtenidos y comentarios plasmados en dichos reportes, han servido para que el jefe de la División del Sistema Universidad Abierta felicite por su desempeño o invite para que el asesor se prepare mejor antes de que inicie el siguiente período escolar.

El proceso de reinscripción anteriormente se realizaba un poco lento, en cambio ahora, como el sistema permite imprimir un comprobante de que el alumno contestó todas las encuestas, ya no es necesario revisar cada una de las encuestas impresas, con esto se disminuye el tiempo.

Los administrativos de forma sencilla han creado nuevas encuestas y el administrador realiza las tareas propias de manera amigable.

La metodología fue llevada a cabo de acuerdo a las especificaciones, además de que se demostró que UML no solo es aplicable al diseño de grandes aplicaciones, sino también de medianas y pequeñas. Además de que no es necesario utilizar todos los diagramas que determina UML, sino únicamente los necesarios para la aplicación en desarrollo.

El modelo de datos relacional fue una excelente elección para el diseño de la base de datos, esto en base a que no se ha tenido problema alguno con los datos almacenados y la explotación de estos, ya que las dependencias fueron identificadas correctamente.

Se comprobó el porque Apache es un servidor Web muy utilizado, ya que es estable, potente y compatible con diferentes complementos, en este caso JavaScript, que hacen amigables, interactivas y dinámicas las interfaces.

La interfaz para el manejo de la base de datos PHP, cuenta con un gran número de funciones que permiten la interacción con el manejador de base de datos; facilitando la explotación y disminuyendo las líneas de código.

Con lo mencionado anteriormente se demuestra que las herramientas, la metodología y el modelo de diseño fueron una buena elección. La base de datos ha funcionado de forma efectiva y consume muy poco tiempo en cuanto a su explotación, en parte por su rapidez del manejador y su compatibilidad con la interfaz.

Hay que mencionar que se tiene la opción para que en un futuro el sistema desarrollado, pueda interactuar directamente con el sistema de inscripción, con lo que se disminuirá el tiempo y proceso de inscripción. Además de que se puede ajustar a las necesidades de otras escuelas o facultades que realicen un proceso semejante.

Finalmente me siento satisfecho con el trabajo realizado, ya que aplique la teoría obtenida en el transcurso de la carrera así como también de los cursos extracurriculares a los que asistí. Tomando

conciencia de que el desarrollo de sistemas es un proceso que se debe de llevar minuciosamente, ya que se pueden omitir o equivocar en algún paso de la metodología y repercute al final. Además de lo importante que es la comunicación entre los actores en el desarrollo del sistema, así como con los usuarios.

# **BIBLIOGRAFÍA**

- ❖ Rumbaugh J., Jacobson I., Booch G., The Unified Modeling Language. Reference Guide, Estados Unidos, Ed. Addison Wesley, 2000.
- ❖ Kulak D., Guiney E., Use Cases, requirements in context, Estados Unidos, Ed. Addison Wesley, 1ª ed., 2000, 329pp.
- ❖ Kendall K. E., Kendall J. U., Systems analysis and design, Estados Unidos, Ed. Prentice Hall, 6ª ed., 2004, 726pp.
- ❖ Pooley R., Wilcox P., Applying UML: Advanced application, Gran Bretaña, Ed. Elsevier, 1ª ed., 2004, 202pp.
- ❖ Larman C., Applying UML and Patterns. An introduction to Object-Oriented analysis and design, Estados Unidos, Ed. Prentice Hall, 1ª ed., 1998, 507 pp.
- ❖ Rob P., Coronel C., Sistemas de bases de datos. Implementación y Administración, México, Ed. Thomson, 5ª ed., 2004, 838 pp.
- ❖ Introducción a los conceptos de bases de datos,  
[http://atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema1\\_1.htm](http://atenea.udistrital.edu.co/profesores/jdimate/basedatos1/tema1_1.htm)
- ❖ MySQL AB, Database Server Feature Comparisons,  
<http://www.mysql.com/information/crash-me.php>
- ❖ MySQL Hispano. La comunidad de usuarios de MySQL,  
<http://www.mysql-hispano.org/index.php>
- ❖ PHP: Manual de PHP,  
<http://mx.php.net/manual/es/index.php>
- ❖ Programación en castellano. Tutorial básico de MySQL,  
[http://www.programacion.com/tutorial/mysql\\_basico/](http://www.programacion.com/tutorial/mysql_basico/)
- ❖ UML,  
<http://www.creangel.com/uml/>
- ❖ Modelos de datos,  
[http://www.itlp.edu.mx/publica/tutoriales/basedat1/tema1\\_4.htm](http://www.itlp.edu.mx/publica/tutoriales/basedat1/tema1_4.htm)
- ❖ Diseño Web. Principales definiciones de los términos utilizados en Internet,  
<http://www.informaticamilenium.com.mx/paginas/espanol/sitioweb.htm>
- ❖ HTML con clase. Breve historia de la World Wide Web,  
<http://html.conclase.net/articulos/historia>
- ❖ MySQL. Reference manual,  
<http://dev.mysql.com/doc/mysql/en/index.html>
- ❖ Portada. Wikipedia, la enciclopedia libre,  
<http://es.wikipedia.org/wiki/Portada>

# **ANEXOS**

## Glosario

**API (Application Programming Interface).** Es un conjunto de especificaciones de comunicación entre componentes software. Representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software.

**Apuntador.** Un apuntador o puntero es una variable manipulable que referencia una región de memoria.

**Atributo.** Es cada una de las cualidades, propiedades o características que describen a la entidad. Los valores que toman los campos de datos pueden ser cuantitativos, cualitativos o descriptivos.

**CGI (Common Gateway Interface).** Programa usado para hacer llamadas a rutinas o controlar otros programas o bases de datos, desde una página web. También puede generar directamente HTML.

**DD (Diccionario de datos).** Es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común en todas las entradas, salidas, componentes de almacenes y cálculos intermedios.

**DDL (Lenguaje de Definición de datos).** Con el se define la estructura (tablas, índices, vistas, etc.) que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

**DML (Lenguaje de Manipulación de datos).** Es el lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, esto implica: insertar, eliminar o modificar.

**DNS (Domain Name System).** Es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres a direcciones IP y la localización de los servidores de correo electrónico.

**Distribución.** Una distribución es un conjunto de aplicaciones reunidas por un grupo, empresa o persona para permitir instalar fácilmente un sistema Linux. En general se destacan por las herramientas para configuración y sistemas de paquetes de software a instalar.

**Entidad.** Representa una colección o conjunto de objetos del mundo real cuyos miembros juegan un papel muy importante en el desarrollo del sistema, pueden ser identificados de manera única y ser descritos a través de atributos.

**Evento.** Se realiza un evento, cuando todas las actividades que llegan a un mismo nodo han sido terminadas.

**HTTP (HyperText Transfer Protocol).** El hipertexto es el contenido de las páginas web y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web y la respuesta de esa web, remitiendo la información que se verá en pantalla. También sirve el protocolo para enviar información adicional en ambos sentidos.

**IAB (Internet Advertising Bureau).** Grupo técnico de asesoramiento de la Sociedad de Internet, cuyas responsabilidades incluyen: controlar Internet, controlar los procesos de estandarización, publicar y administrar los Request for Comments.

**IDE (Integrated Development Environment).** Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un

compilador, un depurador y un constructor de interfaz gráfica. Puede dedicarse en exclusiva a un solo lenguaje de programación o a varios.

**IP (Internet Protocol).** Es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados. Proporciona seguridad. Trabaja en conjunto con TCP.

**Método.** Es una manera explícita de estructurar el pensamiento y las acciones de cada individuo. Además, el método le dice al usuario qué hacer, cómo hacerlo, cuándo hacerlo y por qué hacerlo; mientras que el lenguaje de modelado carece de estas instrucciones. Los métodos contienen modelos y esos son utilizados para describir algo y comunicar los resultados del uso del método.

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

**NCP Protocolo.** Conjunto original de protocolos de control de red de ARPANET.

**Nodo.** Es el punto de unión entre varias redes. Cada nodo de una red tiene un nombre distinto.

**OSI (Open Systems Interconnection).** Es un modelo creado para la interconexión en un contexto de sistemas abiertos. Se estructura en siete niveles: Presentación, Aplicación, Sesión, Transporte, Red, Nivel físico y Enlace de datos; que definen normas en cada uno de ellos desde las conexiones puramente físicas hasta las relaciones entre aplicaciones.

**Puerto.** Es una forma genérica de denominar a una interfaz por la cual diferentes tipos de datos pueden ser enviados y recibidos.

**RPM (RPM Package Manager).** Es una herramienta de administración de paquetes pensada básicamente para Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas.

**Red de área amplia (WAN).** Conecta equipos distantes entre sí, computadoras personales, servidores, ruteadores, puentes, etc., su alcance es de kilómetros, típicamente de alcance global.

**Red de área local (LAN).** Es la interconexión de varias computadoras y periféricos, su extensión está limitada físicamente a un edificio o a un entorno de unos pocos kilómetros.

**Registro.** Un grupo de declaración de tipos de datos, forman lo que se conoce como registro. Un registro puede ser tan pequeño como una sola variable, o tan grande como sea necesario. Uno o más registros forman la estructura de una tabla.

**Sitio espejo (Mirror).** Es una réplica exacta del sitio original y habitualmente se actualiza con frecuencia para garantizar que refleje el contenido del sitio original. Se utiliza para que la carga de páginas o la descarga de archivos sea más rápida al tener un espejo más cerca del cliente.

**TCP (Transfer Control Protocol).** Es el protocolo que se encarga de la transferencia de los paquetes (información) a través de Internet, para lo cual asegura que los paquetes lleguen al destino sin ningún error, en caso contrario solicita el reenvío. Trabaja en conjunto con IP.

**Tuberías.** Son conexiones entre procesos. La salida de un proceso se encadena con la entrada de otro, con lo que se puede procesar datos en una sola línea.

**URL (Uniform Resource Locator).** Es una secuencia de caracteres de acuerdo a un formato estándar, con lo cual se asigna una dirección única a cada uno de los recursos de información disponibles en Internet.





```

        </table>
    </td>
</tr>
<tr>
    <td colspan="3" align="center"><font face="Verdana, Arial, Helvetica, sans-serif" size="4"
color="#003399"><strong>SISTEMA DE EVALUACI&Oacute;N DOCENTE </strong></font></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<table width="750" border="0" align="center">
    <tr>
        <td align="left"><font face="Verdana, Arial, Helvetica, sans-serif" size="2"><a
href="http://www.unam.mx" style="text-decoration:none" target="_blank">UNAM. </a><a
href="http://www.economia.unam.mx" style="text-decoration:none" target="_blank">Facultad de
Econom&iacute;a. </a><a href="http://www.economia.unam.mx/sua/site/principal.php" style="text-
decoration:none" target="_blank">Divisi&oacute;n Sistema Universidad Abierta.</a></font></td>
        <td align="right"><font face="Verdana, Arial, Helvetica, sans-serif" size="1" color="#003399">Este
sitio se ve mejor a 1024 x 768
        <br>2004 D.R. &copy; UNAM</font></td>
    </tr>
</table>
</table>
</form>
</body>
</html>

```

- ❖ **Validar usuario.** El siguiente código busca en la base de datos el número de cuenta y fecha de nacimiento del alumno, para permitir acceder a las encuestas.

```

<?
if($conexion = mysql_connect("localhost","user","passwd")) {
    mysql_select_db("Encuesta",$conexion) OR die("No me puedo conectar");
    $res = mysql_query("select * from Alumno where no_cuenta='$no_cuenta' and
fecha_nac='$fecha_nac'", $conexion);
    $datos = mysql_fetch_object($res);
    $res1 = mysql_num_rows($res);
    if( $res1 ) {
        setcookie("cookie_id", $datos->no_cuenta);
        setcookie("cookie_sem", "2005-1");
        header("location:bienvenida.php");
    }
    else {
        header("location:Index.php?error=Datos incorrectos");
    }
    mysql_close($conexion); //Se cierra la conexion
}
else {
    die("La conexion no se llevo a cabo");
}

```

```
}
?>
```

- ❖ **Materias del alumno.** Fragmento de código para obtener el nombre de la asignatura y el asesor, el grupo y el estado.

```
<?
if ($cookie_id && $cookie_sem) {
    include ("configuracion.php");
    $conexion = conecta();
    $res = mysql_query("select * from AseAsig where no_cuenta='$cookie_id' and
sem='$cookie_sem'", $conexion);
    $num = mysql_num_rows($res);
    include ("Encabezado.html");
}

?>

<!-- Tabla de encabezado -->
<table width="60%" border="0" align="center">
    <tr>
        <td class="titulo">Bienvenid@ alumn@ con n&uacute;mero de cuenta <?=$cookie_id?></td>
    </tr>
</table>
<br>
<table width="690" border="0" align="center">
    <tr>
        <td height="39" class="titulo2">Asignatura</td>
        <td class="titulo2">Asesor</td>
        <td class="titulo2">Grupo</td>
        <td class="titulo2">Estado</td>
    </tr>

<?
if ($num) {
    for ( $i = 0; $i < $num; $i++) {
        $estado = "No contestada";
        $datos = mysql_fetch_object($res);
        print "<tr>";
        //Consulta para traer el nombre de la asignatura
        $nom_asig1 = mysql_query("select * from Asignatura where cve_asig='$datos-
>cve_asig'", $conexion);
        $nom_asig = mysql_fetch_object($nom_asig1);

        //Comprobar si es a distancia o presencial
        if ( (ereg("[3]{1}$", $datos->grupo) || ($datos->grupo == '9190') || (ereg("[4]{1}$", $datos-
>grupo)) ) {
            //Consulta para saber si ya contesto la encuesta a distancia
            $contestada = mysql_query("select * from AseDistancianva where cve_asig='$datos-
>cve_asig' and no_cuenta='$datos->no_cuenta' and semd='$cookie_sem'", $conexion);
            $num_cont = mysql_num_rows($contestada);
            if ( $num_cont ) {
                print '<td height="39" class="preg">'.$nom_asig->nombre_asig.';
                print "</td>";
                $estado = "Contestada";
            }
            else {
```

```

                print '<td height="39" class="ligas"><a
href="distancianva.php?cve_asig='.$datos->cve_asig.'&rfc_asesor='.$datos->rfc_asesor.'" style="text-
decoration:none">'.$nom_asig->nombre_asig.'";
                print "</a></td>";
            }
        }

//Consulta para traer el nombre del asesor
$nom_ase1 = mysql_query("select nombre_asesor from Asesor where rfc_asesor='".$datos-
>rfc_asesor'", $conexion);
$aseso = mysql_fetch_object($nom_ase1);
print '<td class="preg">'.$aseso->nombre_asesor.'";
print "</td>";
print '<td class="preg">'.$datos->grupo.'";
print "</td>";
print '<td class="estado">'.$estado.'";
print "</td>";
print "</tr>";
}
?>

```

- ❖ **Archivo de configuración.** Código utilizado por gran parte de los scripts. Se conecta al servidor y selecciona la base de datos.

```

<?
function conecta() {
    if($conexion = mysql_connect("localhost","user","passwd")) {
        mysql_select_db("Encuesta",$conexion);
        return $conexion;
    }
    else {
        die("La conexion no se llevo a cabo");
    }
}
?>

```

- ❖ **Almacenar encuesta.** El siguiente código almacena las respuestas del alumno en la base de datos, dependiendo del tipo de encuesta.

```

<?
if ($cookie_id && $cookie_sem) {
    include ("Encabezado.html");
    include ("configuracion.php");
    $conexion = conecta();
    switch( $tipo_enc ) {
        case "distancia":
            {
                //Compara si selecciono el porcentaje de asistencia, n=nunca->primera parte sino
                entra a la seg part
                if ( $porcentaje == "n" ) {
                    $res = mysql_query("insert into
                    AseDistancianva(cve_asig,rfc_asesor,no_cuenta,semd,porc_dis,porq_dis) values
                    ('$cve_asig','$rfc_asesor','$cookie_id','$cookie_sem','$porcentaje','$porque')",$conexion);
                }
                else {

```

```

        $res = mysql_query("insert into
AseDistancianva(cve_asig,rfc_asesor,no_cuenta,semd,porc_dis,porq_dis,dis01,dis02,dis03,dis04,dis0
5,dis06,dis07,dis08,dis09,dis10,dis11,dis12,dis13,dis14,dis15,dis16,dis17,dis18,dis19,dis_ab01,dis_ab
02) values
('$cve_asig','$rfc_asesor','$cookie_id','$cookie_sem','$porcentaje','$porque','$p01','$p02','$p03','$p04',
'$p05','$p06','$p07','$p08','$p09','$p10','$p11','$p12','$p13','$p14','$p15','$p16','$p17','$p18','$p19','$pa0
1','$pa02')",$conexion);
    }
    //Si se almacenaron los datos
    if ($res != 0) {
        print "<br><br><br><BR><BR>";
        print "<TABLE width=\"690\" align=\"center\" border=\"0\">";
        print "<TR><TD class=\"preg\">Se han almacenado tus respuestas,
gracias.</TD></TR>";
        print "<TR><TD>&nbsp;</TD></TR>";
        print "<TR><TD class=\"preg\"><A HREF=\"bienvenida.php\"
style=\"text-decoration:none\">&lt;&lt;Regresar</A></TD></TR>";
        print "</TABLE>";
    }
    else {
        print "<br><br><br><BR><BR>";
        print "<TABLE width=\"690\" align=\"center\" border=\"0\">";
        print "<TR><TD class=\"preg\">No se almacenaron tus
respuestas, revisa que hayas contestado todas las preguntas.</TD></TR>";
        print "<TR><TD>&nbsp;</TD></TR>";
        print "<TR><TD class=\"preg\"><A HREF=\"bienvenida.php\"
style=\"text-decoration:none\">&lt;&lt;Principal</A></TD></TR>";
        print "</TABLE>";
    }
}
break;
}
}
?>

```

- ❖ **Reporte de asesores a distancia.** Fragmentos de código de las funciones utilizadas para generar los reportes.

```

//Funcion para imprimir
<SCRIPT LANGUAGE="JavaScript">
function imprimir() {
    version = parseInt(navigator.appVersion);
    if (version >= 4)
        window.print();
}
</SCRIPT>

//Obtener los datos de la asignatura y de los alumnos que contestaron
<?
    if ($semestre) {
        include ("configuracion.php");
        include ("funciones.php");
        $conexion = conecta();
        $res_asig = mysql_query("select * from Asignatura where
cve_asig='$cve_asig'", $conexion);
        $datos_asig = mysql_fetch_object($res_asig);
    }
}
?>

```



## Formatos

### ❖ Formato de encuesta de asesores presenciales.



PROGRAMA DE EVALUACIÓN DOCENTE  
ENCUESTA SOBRE EL DESEMPEÑO DE LOS ASESORES PRESENCIALES  
DEL PERIODO LECTIVO 2002-II

Con el objeto de mejorar la calidad académica de nuestra Facultad, se realiza una evaluación al desempeño DE CADA ASESOR, por lo que se le entrega a los estudiantes un cuestionario por cada asignatura. Agradeceremos que conteste honestamente las preguntas, marcando **sólo una opción dentro del paréntesis correspondiente**.

La información que nos proporcione es ANÓNIMA y muy importante, por lo que le pedimos conteste TODO el cuestionario **sólo si tiene elementos suficientes para ello**.

¿A qué porcentaje de asesorías presenciales asistió? 100 a 76% ( ) De 75 a 50% ( ) Menos de 50% ( )

Explique por qué: \_\_\_\_\_

Para responder en la siguiente sección emplee esta tabla de respuestas:

NUNCA	OCASIONALMENTE	ALGUNAS VECES	FRECUENTEMENTE	SIEMPRE
1	2	3	4	5

1. ¿El asesor asiste a las asesorías grupales programadas?	(1)	(2)	(3)	(4)	(5)
2. ¿El asesor se presenta puntualmente en horarios establecidos para la asesoría grupal?	(1)	(2)	(3)	(4)	(5)
3. ¿El asesor muestra dominio de la asignatura?	(1)	(2)	(3)	(4)	(5)
4. ¿La asesoría se desarrolla sin perder el punto central del tema?	(1)	(2)	(3)	(4)	(5)
5. ¿Relaciona su explicación con otras asignaturas del plan de estudios?	(1)	(2)	(3)	(4)	(5)
6. ¿El asesor utiliza el material didáctico diseñado para la asignatura?	(1)	(2)	(3)	(4)	(5)
7. ¿Utiliza el asesor el material didáctico para orientar su trabajo durante el curso?	(1)	(2)	(3)	(4)	(5)
8. ¿El asesor se muestra accesible para atender dudas, a través de comunicación verbal o escrita fuera de la asesoría grupal?	(1)	(2)	(3)	(4)	(5)
9. ¿El asesor dedica el tiempo necesario para resolver las dudas en su totalidad?	(1)	(2)	(3)	(4)	(5)
10. ¿El asesor responde de manera clara y precisa a las dudas que le plantean?	(1)	(2)	(3)	(4)	(5)
11. ¿El asesor realiza actividades que estimulan la participación del grupo?	(1)	(2)	(3)	(4)	(5)
12. ¿El asesor lleva a cabo actividades de aprendizaje y ejercicios que refuercen los contenidos de la asesoría grupal?	(1)	(2)	(3)	(4)	(5)
13. ¿El asesor resume o da conclusiones al término de cada sesión de asesoría grupal?	(1)	(2)	(3)	(4)	(5)
14. ¿El asesor recomienda lecturas o actividades complementarias al material?	(1)	(2)	(3)	(4)	(5)
15. ¿El asesor explica claramente la forma de evaluar?	(1)	(2)	(3)	(4)	(5)
16. ¿El asesor se basa en el programa de la asignatura para realizar sus evaluaciones?	(1)	(2)	(3)	(4)	(5)
17. ¿Considera que las formas de evaluación son adecuadas para medir los conocimientos adquiridos?	(1)	(2)	(3)	(4)	(5)
18. ¿El asesor entrega oportunamente calificaciones?	(1)	(2)	(3)	(4)	(5)
19. ¿Se inscribiría o recomendaría al asesor en otros cursos?	(1)	(2)	(3)	(4)	(5)

En el siguiente espacio escriba de manera responsable, otros elementos sobre el desempeño del asesor que no hayan sido contemplados en las preguntas anteriores. Haga hincapié en las cosas que permitan realizar mejor nuestro trabajo. Puede emplear el reverso de la hoja para ampliar sus respuestas.

Mencione cuáles fueron los elementos valiosos que consideró del curso:

Anote las sugerencias que mejoren el desempeño del asesor y del curso:

Muchas gracias.

❖ **Formato de satisfacción del cliente.**

<b>ENCUESTA DE SERVICIOS</b>					
<b>SISTEMA DE EVALUACIÓN DOCENTE VÍA WEB DE LA FACULTAD DE ECONOMÍA - SISTEMA UNIVERSIDAD ABIERTA</b>					
Nombre: _____		Fecha: / /			
Puesto: _____					
Conteste las siguientes preguntas, considerando: E = Excelente, B = Bueno, R = Regular, M = Malo					
		M	R	B	E
1	El tiempo consumido para el desarrollo del proyecto es:				
2	La actitud de la persona que llevo a cabo el proyecto fue:				
3	Los conocimientos y disponibilidad de la persona que lo atendió fue:				
4	La comunicación durante el desarrollo del proyecto con la persona encargada fue:				
5	La claridad del manual de usuario es:				
S = Si o N = No		S		N	
6	¿La solicitud se cumplió satisfactoriamente?				
7	¿La capacitación recibida cubrió las expectativas?				
8	Durante la capacitación, ¿se resolvieron sus dudas?				
9	¿El sistema funcionó de forma satisfactoria trabajando en tiempo real?				
10	Sí hubo fallas al momento de utilizarlo, ¿se corrigieron en tiempo razonable?				
11	¿Se agrego información al manual de usuario, después de corregir la falla?				
Comentarios Adicionales					

Gracias