



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA PROTOTIPO PARA EL CONTROL
VÍA WEB DE ENCUESTAS PARA EL
PROGRAMA DE EVALUACIÓN DEL PROCESO
DE ENSEÑANZA – APRENDIZAJE DE LA
FACULTAD DE INGENIERÍA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A N:
JORGE VICENTE CLEMENT TOFFOLÓN
CHRISTIAN MAURICIO LUNA RODRÍGUEZ

Director de Tesis:
M. en I. Jorge Valeriano Assem





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

“Si he conseguido ver más lejos, es porque me he parado en hombros de gigantes.

No se lo que pareceré a los ojos del mundo, pero a los míos es como si hubiese sido un muchacho que juega en la orilla del mar y se divierte de tanto en tanto encontrando un guijarro más pulido o una concha más hermosa, mientras el inmenso océano de la verdad se extendía, inexplorado frente a mí”.

Isaac Newton

Introducción.

| | |
|-------------------------------|---|
| Proceso enseñanza aprendizaje | 4 |
|-------------------------------|---|

Antecedentes

| | |
|--|---|
| Antecedentes | 5 |
| <i>Situación actual</i> | |
| Deficiencias actuales del programa | 6 |
| Importancia de la evaluación del proceso enseñanza-aprendizaje | 7 |

1. Marco teórico

| | |
|---|----|
| 1.1 Herramienta de Desarrollo | 10 |
| <i>Lenguaje de programación PHP 4</i> | 10 |
| Características Fundamentales | 10 |
| Funcionamiento | 14 |
| Conectividad a base de datos con PHP | 15 |
| 1.2 Arquitectura Cliente Servidor | 18 |
| Modelos de la Arquitectura Cliente/Servidor | 20 |
| 1.3 Protocolo HTTP | 22 |
| 1.4 HTML | 25 |
| 1.5 Manejadores de Bases de Datos | 26 |
| 1.6 Metodología | 39 |
| 1.7 Modelado de la aplicación | 41 |

2. Definición del problema y análisis

| | |
|-----------------------------|----|
| 2.1 Definición del problema | 48 |
| 2.2 Sistema propuesto | 49 |
| Objetivos | 49 |
| 2.3 Esquemas de operación | 50 |
| Esquema actual | 50 |

| | |
|---|----|
| Esquema propuesto | 51 |
| Mejoras al actual esquema | 52 |
| 2.4 Método empleado para la solución | 53 |
| 2.5 Etapas de Desarrollo de la aplicación | 54 |
| Servidor de Base de datos y Base de datos (Back-End tier) | 54 |
| Diseño y construcción de la base de datos en SQL Server | 55 |
| Servidor de Web y de aplicaciones (Middle tier) | 57 |
| Cliente (Front - End tier) | 59 |
| 2.6 Diseño WEB | 59 |
| 2.7 Diseño y análisis de la encuesta | 62 |
| 2.8 Evaluación del sistema | 69 |

3. *Desarrollo, implantación y pruebas del sistema.*

| | |
|--|----|
| 3.1 Creación de la Base de Datos | 72 |
| 3.2 Modelo Entidad Relación | 72 |
| Base de Datos | 72 |
| 3.3 Programación de los módulos | 81 |
| 3.4 Herramientas de desarrollo y contenido | 81 |
| 3.5 Integración del sistema | 82 |
| Módulo de conexión | 82 |
| Módulo de validación de datos | 82 |
| Módulo de administración | 85 |
| Módulo de encuestas | 86 |
| Módulo de reportes | 87 |
| Módulo de procesos | 92 |
| 3.6 Pruebas finales del programa piloto | 92 |

4. *Manual de usuario.*

| | |
|-----------------------|----|
| 4.1 Manual de usuario | 99 |
|-----------------------|----|

| | |
|--|-----|
| 4.2 Manual de instalación | 99 |
| Requerimientos de Hardware y Software | 99 |
| 4.3 Funcionamiento del sistema | 100 |
| Convenciones generales del sistema | 100 |
| 4.4 Descripción del ambiente de trabajo | 100 |
| 4.5 Procesos principales del sistema | 100 |
| Opciones del administrador | 102 |
| Altas | 103 |
| 4.6 Procesos estadísticos | 105 |
| Ver Reporte tipo | 106 |
| Encuesta del alumno | 107 |
| Reportes de profesor, jefe de departamento y división. | 109 |

5. *Manual Técnico*

| | |
|-------------------------------------|-----|
| 5.1 Manual Técnico | 112 |
| Convenciones del Manual | 112 |
| Presentación | 112 |
| 5.2 Introducción | 112 |
| 5.3 Objetivos | 112 |
| 5.4 Archivos del sistema | 113 |
| Componentes elementales del sistema | 113 |
| Extensión de los archivos | 114 |
| Programas principales | 115 |
| 5.5 Base de datos del sistema | 118 |
| Definición de la base de datos | 118 |
| 5.6 Estructura del sistema | 119 |
| Listado de tablas | 119 |
| Índices de la base de datos | 134 |
| 5.7 Diagramas de bloques. | 144 |
| Módulos y opciones | 144 |
| Descripción de los módulos | 144 |

Conclusiones. 161

Glosario. 166

Bibliografía 180

1. Introducción.

Proceso enseñanza aprendizaje

Para mantener la eficacia educativa de una institución como la Universidad Nacional Autónoma de México, específicamente en el caso de la Facultad de Ingeniería resulta de gran importancia el contar con mecanismos o sistemas de autoevaluación que permitan determinar el grado de cumplimiento de sus objetivos e impulsar el mejoramiento continuo, que incluya los resultados de las evaluaciones de los alumnos, los docentes así como el plan de estudios.

Es así que atendiendo a dicha necesidad que desde hace 15 años, en la Facultad de Ingeniería se aplica, de manera general y uniforme, un instrumento para evaluar la labor docente mediante la opinión de los alumnos con el denominado programa de evaluación del proceso enseñanza – aprendizaje, el cual tiene como objetivo el de mejorar la calidad académica de nuestra Facultad mediante el ejercicio de encuestas mediante periodos semestrales en los cuales el alumno tiene la oportunidad de realizar una evaluación del proceso enseñanza – aprendizaje al expresar su punto de vista acerca del ejercicio docente y también de su propio desempeño como estudiante.

Con este ejercicio se debe lograr impulsar el mejoramiento continuo de la Facultad así como también de la estructura académica y de los planes y programas de estudio.

Dicho cuestionario se aplica tres semanas antes de la terminación del semestre y la aplicación ocupa entre 20 y 25 minutos y se realiza en el salón dentro de horario de clases de cada grupo – asignatura estando el profesor presente.

Cada alumno responde tantos cuestionarios como asignaturas tenga inscritas ese semestre. De esta manera, se responden alrededor de *45,000 encuestas.

* Información obtenida de la Coordinación de Proyectos Académicos en su documento “Estructura y Criterios de calificación del cuestionario de opinión de los alumnos acerca de la docencia en la Facultad de Ingeniería”

Antecedentes

Situación actual

Actualmente el programa de evaluación del proceso enseñanza – aprendizaje cuenta con una encuesta impresa de tipo óptica donde el estudiante rellena alvéolos para contestar un total de 29 preguntas divididas entre preguntas sobre el profesor que son 20 y 10 preguntas sobre el alumno además de contar adicionalmente con una área abierta de comentarios y sugerencias al profesor que imparte la asignatura específica.

Dicho ejercicio de evaluación se realiza durante las 3 semanas previas a la conclusión del semestre, mediante la visita de la persona que va a aplicarla; en el aula y en los horarios de clases; se distribuyen entre el alumnado las hojas impresas de tipo óptica en donde cada uno llena datos particulares como lo son el nombre del profesor, asignatura, semestre, clave de asignatura, grupo, tipo de clase y número de profesor, una vez hecho esto se procede a responde la encuesta en sus 20 preguntas que corresponden al apartado del profesor y 10 de auto evaluación sobre el alumno, además de posibles comentarios y sugerencias. Estas encuestas se guardan en un sobre cerrado para mandarlas al área de codificación, procesamiento y reporte de la información, a cargo del Centro de Cálculo de la Facultad, para distribuir los resultados en forma de reportes que se elaboran en dos clases, los que van dirigidos al profesor por cada grupo que tuvo a cargo y los que contienen información de todos los grupos de una asignatura, departamento o división, con la calificación por pregunta, la calificación por concepto y la calificación total, por cada grupo de cada profesor.

Deficiencias actuales del programa

- ☐ Pueden existir errores al momento de llenar la información
- ☐ El llenado de la encuesta resulta tedioso y monótono, lo cual le resta fiabilidad a la encuesta.
- ☐ Interrumpe el desarrollo de las clases.
- ☐ Hay que programar la coordinación del personal que aplica las encuestas con los horarios de clase de cada profesor.
- ☐ Existen ocasiones en las que los académicos no se encuentran en las aulas designadas en el momento de la realización de la encuesta, ya sea por motivos de ausencia, enfermedad, cambio de aula, etc.
- ☐ La muestra de alumnos no siempre es el la totalidad de la comunidad estudiantil.
- ☐ La información que arroja la encuesta no se encuentra disponible de manera rápida.
- ☐ La información no se captura digitalmente en una base de datos, para su posterior análisis.
- ☐ Gastos innecesarios, ya que se emplean recursos tales como papelería, recursos humanos para elaborar las encuestas, entre otros; lo cual no es congruente con el carácter de nuestra Facultad.
- ☐ Falta de flexibilidad al tratar de incluir o cambiar los contenidos de la encuesta, para adecuarlos a las necesidades actuales de la Facultad.

Importancia de la evaluación del proceso enseñanza-aprendizaje

El objetivo principal de la evaluación del proceso enseñanza-aprendizaje es la realimentación en el proceso enseñanza-aprendizaje; esto significa que los datos obtenidos en la evaluación servirán a los que intervienen en dicho proceso (docentes-alumnos) en forma directa para mejorar las deficiencias que se presenten en la realización de dicho proceso e incidir en el mejoramiento de la calidad y en consecuencia el rendimiento de la experiencia educativa que brinda nuestra institución.

Uno de los problemas que más preocupa a los educadores de nuestros días es el de lograr medios idóneos para establecer hasta qué punto los educandos alcanzan las metas educativas preestablecidas.

Si consideramos a la enseñanza como el control de las situaciones en las que ocurre la modificación de conducta o la adquisición de una habilidad en el alumno, es importante que el educador cuente con los procedimientos e instrumentos idóneos para juzgar el grado en que se dan los cambios, tanto al final del proceso como durante el mismo. Mediante la evaluación se puede conocer hasta qué punto los alumnos han modificado su conducta como un resultado, planeado y directo de la acción educativa.

El proceso de enseñanza-aprendizaje incluye una serie continua e interrelacionada de decisiones relativas a la instrucción que buscan incrementar la calidad del aprendizaje de los alumnos. Sin embargo, esta efectividad depende en gran medida de la calidad de información dada por la evaluación sobre la cual se habrán de basar las decisiones que tomen en cada etapa de dicho proceso. De esta situación se desprende la primordial importancia que adquiere la evaluación dentro del proceso de enseñanza-aprendizaje.

Dado el enfoque que tiene este trabajo, de aquí en adelante sólo nos abocaremos a la evaluación referida a criterio y así diremos que la principal función de la evaluación es la de retroalimentar el proceso de enseñanza-aprendizaje, es decir que los datos obtenidos en la evaluación servirán a los que intervienen en dicho proceso (maestros y alumnos) en forma directa para mejorar las deficiencias que se presentan en la realización del proceso.

La correcta evaluación permite así,

A los académicos

- ☐ Saber cuáles fueron los objetivos alcanzados y en qué medida se dio el logro.
- ☐ Tener la posibilidad de formular un análisis de las causas que pudieron haber ocasionado las deficiencias en las metas propuestas y tomar decisiones.
- ☐ Evitar incurrir en los mismos errores en experiencias posteriores.
- ☐ Reforzar oportunamente las áreas de estudios en que el aprendizaje haya sido insuficiente (detectable en el rendimiento grupal frente a los instrumentos de evaluación).
- ☐ Juzgar la viabilidad de los programas a la luz de las circunstancias y condiciones reales de operación.

Al alumnado:

- ☐ Tener una fuente de información para que se reafirmen los aciertos y se corrijan los errores
- ☐ Poder contar con un medio para expresar su opinión en torno a la experiencia del proceso aprendizaje enseñanza.
- ☐ Dirigir su atención hacia los aspectos centrales del material de estudio.
- ☐ Reforzar las áreas de estudio en que el aprendizaje haya sido insuficiente.

Morán*, concibe la evaluación no como "un acto mediante el cual un profesor juzga a un alumno, sino un proceso a través del cual el profesor y el alumno aprecian en qué grado logró este último los aprendizajes que ambos perseguían" y que "como actividad indispensable en el proceso educativo puede proporcionar una visión clara de los errores para corregirlos, de los obstáculos para superarlos y de los aciertos para mejorarlos" y cumplir de esta manera las funciones que tienen la evaluación educativa en el proceso de enseñanza-aprendizaje.

* MORÁN OVIEDO. La Evaluación en los aprendizajes y sus Implicaciones Sociales, Perfiles Educativos No. 3, México, 1980.

CAPÍTULO 1

Marco Teórico

1.1 Herramienta de Desarrollo

Lenguaje de programación PHP 4

Características Fundamentales

Funcionamiento

Conectividad a base de datos con PHP

1.2 Arquitectura Cliente Servidor

Modelos de la Arquitectura Cliente/Servidor

1.3 Protocolo HTTP

1.4 HTML

1.5 Manejadores de Bases de Datos

1.6 Metodología

1.7 Modelado de la Aplicación

1.1 Herramienta de Desarrollo

Lenguaje de programación PHP 4

El lenguaje de programación PHP permite la generación dinámica de contenidos en un servidor Web. También conocido con el nombre oficial de HyperText Preprocessor (teniendo sus orígenes a principios de 1995 en el conjunto de macros conocidos como Personal Home Page Tool). Originalmente creado por Rasmus Lerdorf como un conjunto de utilidades para adicionar dinamismo a las páginas Web (conocidas como PHP/FI y posteriormente PHP 2.0). Cuando este conjunto de herramientas ganó popularidad fueron retomadas y completamente rediseñadas por Zeev Suraski y Andi Gutmans naciendo el PHP 3.0.

A partir de aquí se volvió a rediseñar completamente el intérprete, añadiéndole más potencia y nuevas funciones, para dar lugar a lo que actualmente se conoce como PHP4.

PHP es un lenguaje de programación que contiene muchos conceptos de C, Perl y Java; siendo su sintaxis muy similar a la de dichos lenguajes, por lo que es sencillo de aprender.

El código de PHP está embebido en los documentos HTML (HyperText Markup Language) de manera que facilita la incorporación información actualizada e el sitio Web.

Características Fundamentales

Entre las características fundamentales con las que cuenta PHP 4 contamos con las siguientes:

- ❑ Es un potente y robusto lenguaje de programación embebido en documentos HTML.
- ❑ Dispone de librerías de conexión con un conjunto muy grande de gestores de bases de datos para el almacenamiento, actualización y modificación de información de manera permanente en el servidor.

- ☐ Proporciona soporte a múltiples protocolos de comunicación de Internet tales como:
 - HTTP HyperText Transfer Protocol
 - IMAP Protocol
 - FTP File Transfer Protocol
 - LDAP Protocol
 - SNMP Simple Network Management Protocol, etc.
- ☐ Codificación y decodificación MIME base G4
- ☐ PHP es un lenguaje de código fuente abierto (Open Source): el código del intérprete está accesible para permitir posibles mejoras o sugerencias en su desarrollo ya que ha sido creado en lenguaje C.
- ☐ Gratuito (Freeware)
- ☐ Portable y multiplataforma: Existen versiones del intérprete para múltiples plataformas (Windows 95, 98, NT, 200 Unix, Linux, etc.) Lo cual repercute en que se pueda portar las aplicaciones de una plataforma a otra sin modificar una sola línea de código.
- ☐ Eficiente ya que consume muy pocos recursos en el servidor.
- ☐ Alta velocidad de desarrollo con una gran cantidad de librerías útiles y bien documentadas.

Cosas para tener en cuenta en un lenguaje de scripts

Las cuatro grandes características: Velocidad, estabilidad, seguridad y simplicidad.

- ☐ Velocidad: No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software.

- ☐ Estabilidad: La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para los bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- ☐ Seguridad: El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo `.ini`
- ☐ Simplicidad: Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

Ventajas adicionales de PHP

- ☐ PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.
- ☐ La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C incluimos al Java y Javascript, de hecho mucha de la funcionalidad del PHP se la debe al C en funciones como `fread()` o `strlen()`, así que muchos programadores se sentirán muy identificados.
- ☐ PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend*), un conjunto de módulos y una variedad de extensiones de código.
- ☐ Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THHTTPD. Otra alternativa es configurarlo como módulo CGI.

* El nombre Zend proviene de sus autores Zeev Suraski y Andi Gutmans
<http://www.zend.com>

- ☐ Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran.
- ☐ Una gran variedad de módulos cuando un programador PHP necesite una interfase para una librería en particular, fácilmente podrá crear una API* para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.
- ☐ Rapidez. PHP generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- ☐ PHP es Open Source*, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.

Algunas desventajas

El manejo de errores no es tan sofisticado como Cold Fusion o ASP. No existe IDE* o Debugger. Una IDE puede no ser importante para la mayoría de los programadores y un debugger ha sido prometido por Zend Tech para un futuro muy cercano.

PHP en Windows

En ambientes Windows compite muy de cerca con ASP* y Cold Fusion, aquí la elección se basa en asuntos un poco más técnicos y en la política que desee utilizarse para el sitio. ASP junto a IIS es probablemente más estable que PHP con IIS. Pero en términos puramente técnicos, PHP bajo

*API Application Program Interface

*IDE Integrated Development Environment, o en español, Ambiente de Desarrollo Integrado

*Para ampliar información sobre open source puede consultar <http://www.opensource.org>

* ASP Active Server Page es una tecnología propietaria de Microsoft para desarrollo de contenidos web similar a PHP.

Windows NT es mucho más estable que los otros dos (además de ser más rápido y utilizar menos recursos). De cualquier manera ASP ofrece una mejor integración con este ambiente sobre todo si se desea utilizar COM.

Como muchas otras cosas en la caja de herramientas de un programador, PHP ha sido creado casi accidentalmente. Empezando como experimento de una persona, modificado para adaptarse a las necesidades de muchos otros programadores de un lenguaje rápido, simple, robusto que pueda trabajar en conjunto a otros componentes. En los últimos tres años ha crecido de forma desmedida, tomando así el papel principal del scripting del tipo Server Side.

Funcionamiento

Las páginas exclusivamente con código HTML no requieren de ser ejecutadas en el servidor ya que son interpretadas en el navegador del cliente (browser). Sin embargo en el caso de PHP se requiere de la intervención de un servidor.

Cuando llega una petición mediante el protocolo HTTP a un servidor Web, este localiza el documento solicitado por el cliente y, en función de una serie de parámetros de la propios de la configuración del servidor, se decide la acción a realizar con el documento como por ejemplo la ejecución de un query de SQL.

Haciendo una descripción de los pasos realizados entre el cliente y el servidor para la obtención de un documento dinámico mediante PHP como se ilustra en la figura 1.1 tenemos:

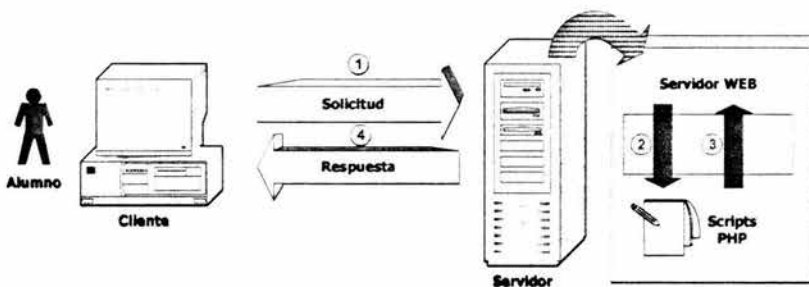


Figura 1.1 Fases de obtención de un documento dinámico.

1. El usuario pulsa sobre un enlace realizando una solicitud y el navegador envía la petición al servidor utilizando el protocolo HTTP.
2. La solicitud llega hasta el servidor Web correspondiente a través de la red. El servidor localiza el documento por la extensión del nombre del archivo, determinado que se trata de un archivo php (Los documentos que contiene código PHP usualmente tiene la extensión .php o .phtml siendo posible la configuración de extensiones válidas en el servidor)
3. El intérprete ejecuta el script solicitado y genera un resultado (habitualmente una página HTML) que se devuelve al servidor para que este a su vez lo transfiera al cliente.
4. Se visualiza el resultado en el navegador del cliente.

Conectividad a base de datos con PHP

La utilización más importante de PHP se da en la utilización del lenguaje SQL como herramienta estándar para realizar consultas a cualquier gestor de base de datos, y la forma en la que PHP se puede conectar con dichos gestores de bases de datos que estén alojados en el servidor.

PHP permite al programador relacionarse con algunas de las bases de datos más importantes como lo son:

-  Adabas
-  Dbm
-  Informix
-  MS SQL Server
-  PostgreSQL
-  MySQL
-  mSQL
-  Oracle
-  Sybase

Dentro de una petición del servidor Web hacia el motor de PHP, se puede encontrar la ejecución de sentencias de manipulación de datos almacenados por un gestor de base de datos. La forma en la que PHP realiza la conexión (tal como se muestra en la figura 1.2) con dicho gestor se da de dos formas: Por ODBC (Open Data Base Connectivity) y por funciones nativas.

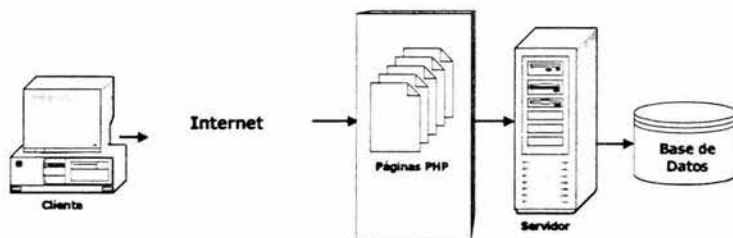


Figura 1.2. Consulta a una Base de datos mediante PHP

En particular, la conexión ODBC consiste en una API estándar, que presenta una capa de software que oculta los detalles de conexión y que para su utilización requiere de la instalación y activación de su módulo correspondiente.

En la segunda forma de conexión mediante funciones nativas propias de cada gestor, se proporciona un método de acceso a los datos mucho más eficiente, ya que no necesita la intervención de esa capa de software, incrementando la velocidad de consultas.

De modo práctico las operaciones PHP en las bases de datos podrían dividirse en tres niveles:

- abrir/cerrar conexiones
- enviar preguntas SQL
- leer los resultados de una query

Las operaciones abrir/cerrar una conexión a una base de datos sirven para abrir/cerrar una conexión con la base de datos elegida con el fin de hacer operaciones como leer/escribir y transmitir los datos en formato html.

Para abrir/cerrar una conexión en el gestor de base de datos de SQL Server contamos con las funciones de mssql:

```
mssql_connect(host, usuario, contraseña)
mssql_close()
```

Para mandar preguntas se puede utilizar la instrucción:

```
mssql_db_query(base de datos, query, ID_base de datos)
```


1.2 Arquitectura Cliente Servidor

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- ☐ El servidor presenta a todos sus clientes una interfaz única y bien definida.
- ☐ El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- ☐ El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- ☐ Los cambios en el servidor implican pocos o ningún cambio en el cliente.

El cliente es la entidad por medio de la cual un usuario solicita un servicio, realiza una petición o demanda el uso de recursos. Este elemento se encarga, básicamente, de la presentación de los datos y/o información al usuario en un ambiente gráfico.

Se comunica con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad; además, requiere el uso de los recursos de la computadora para cualquier actividad y puede interactuar con uno o varios servidores.

Los clientes se suelen situar en PC's o en estaciones de trabajo se encargan de realizar el FRONT END, que es la parte de la aplicación que interactúa con el usuario, en ellos permanecen las aplicaciones particulares de cada usuario, y realizan funciones como:

- ☐ Manejo de la interfaz del usuario.
- ☐ Captura y validación de los datos de entrada.
- ☐ Generación de consultas e informes sobre las bases de datos.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios. El servidor es la entidad física

que provee un servicio y devuelve resultados; ejecuta el procesamiento de datos, aplicaciones y manejo de la información o recursos. En el servidor se realiza el BACK END que es la parte destinada a recibir las solicitudes del cliente y dónde se ejecutan los procesos.

En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente.

Servidor de Bases de datos.

El cliente envía solicitudes de SQL en calidad de mensajes (un mensaje por instrucción); el servidor hace uso de su propia capacidad de procesamiento para encontrar los datos solicitados y devolverlos por medio de la red, sin enviar todos los registros. Este tipo de servidores permiten hacer consultas específicas y obtener reportes flexibles.

Servidor de Transacciones.

El cliente activa procedimientos remotos que residen en el servidor con un mecanismo de bases de datos de SQL. Es decir, el intercambio por la red consiste en un solo mensaje de solicitud/respuesta que ejecuta un grupo de instrucciones SQL (llamadas transacciones) en el servidor; al crear la aplicación Cliente/Servidor, se genera código tanto para el cliente como para el servidor. A estas aplicaciones se les llama procesamiento de transacciones en línea (OLTP: online transaction processing), las cuales requieren de firmes controles sobre la seguridad e integridad de la base de datos.

Servidor de Web.

Se usan como una forma inteligente para comunicación entre empresas a través de Internet. Este servidor permite transacciones con el acondicionamiento de un browser específico. Este modelo está integrado por clientes compactos y portátiles en comunicación con servidores amplios. Tal comunicación se da mediante un protocolo denominado HTTP, el cual define un conjunto simple de órdenes, los parámetros se transmiten en cadenas sin estipulaciones de datos teclados. En el proceso de ampliación de Internet, ya existen combinaciones con objetos distribuidos como en PHP para ofrecer modalidades más interactivas.

Servidor de Aplicaciones.

En este medio se almacenan y ejecutan las aplicaciones de software utilizadas por los usuarios, evitando así la duplicidad de las mismas, permiten un mejor control para la actualización de versiones y productos.

Modelos de la Arquitectura Cliente/Servidor

Dado que la percepción sobre la Arquitectura Cliente/Servidor es algo abstracto, algunos autores la definen en base a modelos de dos y tres niveles de acuerdo al número de estratos respectivos para representar a sus componentes:

Modelos de Dos Niveles (Two Thier).

Se considera como modelo Cliente/Servidor de dos niveles o capas a la estructura más simple, cuyos componentes son:

Clientes: Por medio de la interfaz con el usuario vía una petición se solicita un servicio, el uso de un recurso, o bien el acceso a un conjunto de datos.

Servidores: Satisfacen la solicitud del usuario recibiendo la petición, direccionándola y enviando la respuesta al cliente, ya sea la consulta respectiva de datos, ejecutando el proceso requerido o permitiendo el acceso y/o uso del recurso.

En este modelo se acostumbra instalar las bases de datos dentro del servidor, por las ventajas de almacenamiento y velocidad que ofrece en comparación con las del cliente.

Modelo de Tres Niveles (Three Thier).

El objetivo de este modelo que se presenta gráficamente en la figura 1.3, es dividir las funciones de una aplicación en tres componentes que son Presentación, Servidores e Información:

Presentación: También conocido como Front – End. Este componente se encarga de la interacción hombre máquina a través del monitor, teclado, ratón, o bien mediante algún otro medio como reconocedor de voz.

Servidores: Es la capa media y está compuesta por varios servidores o componentes de Software localizados en una o más plataformas que se encargará de conectar los sistemas existentes.

Información: Nivel conocido también como Back - End. En este componente se incluye la información en sí, los sistemas y aplicaciones existentes.

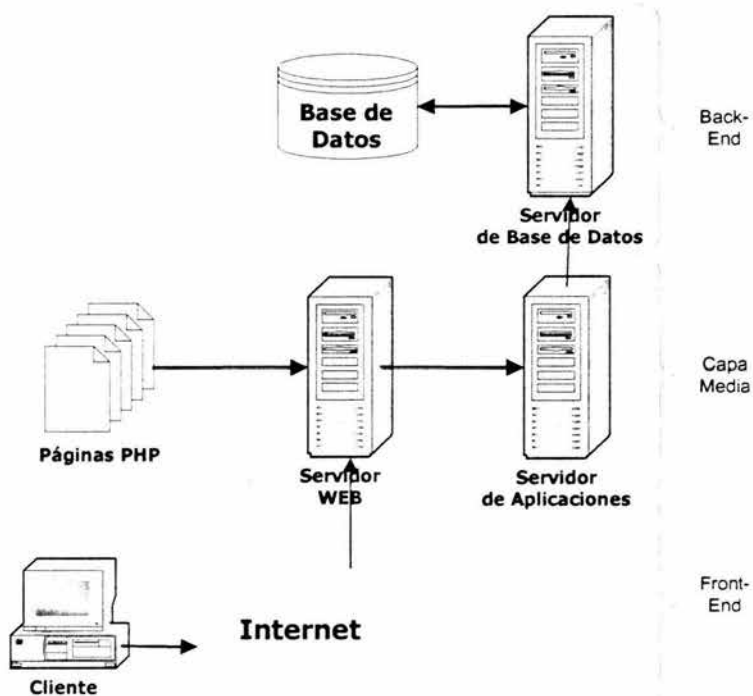


Figura 1.3 Modelo de la Arquitectura Cliente/Servidor de Tres Niveles (Three Thier).

1.3 Protocolo HTTP

HTTP (HyperText Transfer Protocol) es un protocolo de Internet que fue creado para que las computadoras se comuniquen mientras intercambian documentos, agregando conectividad e interfaces. Usando HTTP, una computadora que pida un archivo a otra sabrá, al recibirlo, si se trata de imagen, video o texto.

Con esta nueva función que agregó HTTP, Internet comenzó a demostrar que es inútil recuperar un archivo si no se sabe qué tipo de datos tiene. Con la cantidad de documentos diferentes que hay en la Web, sería imposible saber qué tipo de dato tiene uno de ellos por adelantado, ya que podría contener cualquier cosa. Pero, afortunadamente, se diseñó la Web para que pudiera entender los tipos de datos y pasa esa información.

Aunque no es necesario conocer cómo funciona HTTP para, por ejemplo, construir un programa CGI, es bueno que se tome conocimiento de todo esto para hacer la programación de tareas con mayor fluidez y confianza. Como en cualquier rubro, un vistazo a los principios de base nos permite tener una mayor comprensión general y un mejor y más completo modelo mental en el que basar nuestro trabajo.

Debajo de la interfaz de usuario, representada por los navegadores, están los protocolos. Estos viajan por los cables que conforman la red hacia los servidores o motores que procesan los pedidos de información (requests) y devuelven resultados. El protocolo de Web se conoce como HTTP. Es el mecanismo de base en el cual opera CGI, por lo que es quien determina lo que se puede y lo que no puede hacer con CGI.

Según la especificación del protocolo, "HTTP es un protocolo del *nivel de aplicación* con la agilidad y velocidad necesaria para sistemas de información distribuidos, colaborativos y de hipermedia. Es un protocolo orientado a objetos, genérico, que puede usarse para muchas tareas extendiendo sus métodos. Una característica de HTTP es que permite que los sistemas se construyan independientemente de la información que se transfiere".

Representación abierta y extensible de tipos de datos: HTTP usa Internet Media Types, comúnmente llamados tipos de contenido MIME (Multipart Internet Mail Extension), para brindar una negociación de tipos extensible y abierta. Con HTTP, donde el emisor y el receptor se pueden comunicar directamente, las aplicaciones tienen permitida una mayor libertad en el uso de tipos no registrados. Para aplicaciones de mail, por ejemplo, donde no hay una negociación de tipos entre el emisor y el receptor, es razonable poner límites estrictos en el conjunto de tipos permitidos.

Cuando el cliente envía una transacción al servidor, se agregan encabezados (headers) para conformar las especificaciones estándares de e-mail (RFC822). La mayoría de los pedidos de clientes esperan una respuesta en "sólo texto" o en HTML. Cuando el servidor HTTP transmite información de vuelta al cliente, incluye un encabezado simil-MIME para informar al cliente qué tipo de dato viene después de dicho encabezado. La traducción depende de que el cliente procese la utilidad adecuada que corresponda a ese tipo de dato. Por ejemplo, un visor de imágenes si es una imagen, de video si es un video, etc.).

Campos de encabezados: Una transacción HTTP consiste de un encabezado seguido, opcionalmente, por una línea en blanco y algún dato. El encabezado especificará cosas como la acción requerida del servidor, o el tipo de dato retornado, o el código de estado. El uso de campos de encabezados enviados en las transacciones HTTP le dan gran flexibilidad al protocolo. Estos campos permiten que se envíe información descriptiva en la transacción, permitiendo así la autenticación, encriptación e identificación de usuario. Un encabezado es un bloque de datos que precede a la información propiamente dicha, por lo que muchas veces se hace referencia a él como metadato - porque tiene datos sobre los datos-. Si se reciben líneas de encabezado del cliente, el servidor las coloca en las variables de ambiente de CGI con el prefijo HTTP_ seguido del nombre del encabezado. Cualquier carácter guión (-) del nombre del encabezado se convierte a caracteres "_". El servidor puede excluir cualquier encabezado que ya esté procesado, como Authorization, Content-type y Content-length. El servidor puede elegir excluir alguno o todos los encabezados, si incluirlos excede algún límite del ambiente de sistema. Ejemplos de esto son las variables HTTP_ACCEPT y HTTP_USER_AGENT.

HTTP_ACCEPT son los tipos MIME que el cliente aceptará, dado los encabezados HTTP. Otros protocolos quizás necesiten obtener esta información de otro lugar. Los items en esta lista debes estar separados por una coma, como lo dice la especificación HTTP: tipo, tipo
HTTP_USER_AGENT usa el navegador que utiliza el cliente para enviar el pedido. El formato general para esta variable es: software/versión librería/versión

El servidor envía al cliente:

Un código de estado que indica si el pedido fue exitoso o no. Los códigos de error típicos indican que el archivo solicitado no se encontró, que el pedido no está en forma o que se requiere autenticación para acceder al archivo.

La información propiamente dicha. Como HTTP permite enviar documentos de todo tipo y formato, es ideal para transmitir multimedia, como gráficos, audio y video. Esta libertad es una de las mayores ventajas de HTTP.

También envía información sobre el objeto que se retorna.

Métodos HTTP

HTTP ofrece un conjunto de métodos para indicar el propósito del pedido. Los tres más usados son HEAD, GET y POST.

El método GET: Se usa para pedir un documento específico. Cuando se hace click en un link, se está usando el método GET. Este método debería usarse cuando el acceso URL no cambiará el estado de una base de datos.

La semántica de GET cambia a un GET condicional si el mensaje de pedido incluye un campo de encabezado If-Modified-Since. Un GET condicional pide que la fuente identificada sea transferida sólo si ha sido modificada después de la fecha dada por el encabezado. El GET condicional intenta reducir el uso de la red permitiendo que las entidades que están en caché sean refrescadas sin requerir la transferencia innecesaria de datos.

El método HEAD: Este método sólo para preguntar información sobre un documento, no para el propio documento. HEAD es mucho más rápido que GET, ya que se transfiere mucha menos información. Es generalmente

usada por clientes que usan caching para ver si el documento ha cambiado desde la última vez que lo accedieron. Si no cambió, la copia local puede usarse nuevamente; de otra forma, la versión actualizada debe recuperarse con GET. La metainformación contenida en los encabezados HTTP en respuesta a un pedido HEAD deberían ser idénticos a la información enviada en respuesta a un pedido GET. Este método se puede usar para obtener metainformación sobre una fuente identificada por el URI sin transferir la información. Este método mucho para testear la validez, accesibilidad y reciente modificación de links.

El método POST: Es usado para transferir datos del cliente al servidor; está diseñado para permitir que un método uniforme cubra funciones como: anotación de fuentes existentes; envío de mensajes para un bulletin board, grupo de noticias o lista de mailing; proveer un bloque de datos (generalmente un formulario) a un proceso de manejo de datos; extensión de una base de datos.

1.4 HTML

El HyperText Markup Language (HTML) es generalmente aceptado y considerado como un standard de WWW (World Wide Web). El hipertexto es una forma de presentar textos que contengan enlaces a otros textos de una manera no lineal. HTML es un formato no propietario, cuyo desarrollo se ha basado en SGML (Standard Generalized Markup Language).

Básicamente son archivos de texto planos que contienen código incluido que se trata de un conjunto de etiquetas que sirven para definir la forma en la que presentar el texto y otros elementos de una página Web.

El HTML se creó en un principio con objetivos divulgativos. No se pensó que la Web llegara a ser un área de ocio con carácter multimedia, de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, se han ido incorporando modificaciones con el tiempo, denominadas como estándares del HTML.

Numerosos estándares se han presentado siendo el HTML 4.01 el último estándar dado a conocer en febrero de 2001.

El HTML es un lenguaje de programación muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida pueda enfrentarse a la tarea de crear una Web.

El lenguaje HTML consta de etiquetas que tienen esta forma `` o `<P>`. Cada etiqueta significa una cosa, por ejemplo `` significa que se escriba en negrita (bold) o `<P>` significa un párrafo, `<A>` es un enlace, etc. Casi todas las etiquetas tienen su correspondiente etiqueta de cierre, que especifica los alcances de las etiquetas o tags, es decir, indica que a partir del punto de cierre no debe tener efecto la etiqueta. Por ejemplo `` se utiliza para indicar que se deje de escribir en negrita. Así que el HTML no es más que una serie de etiquetas que se utilizan para definir la forma o estilo que queremos aplicar a nuestro documento.

1.5 Manejadores de Bases de Datos

Los sistemas manejadores de bases de datos DBMS (Database management systems) son un conjunto de aplicaciones que interactúan con la base de datos, permitiendo definición, control y manipulación de los datos contenidos en ella.

Su objetivo primordial es el de crear un ambiente propicio para almacenar y recuperar la información de manera eficiente, conveniente y confiable.

Entre las funciones principales que tiene un manejador de base de datos son:

- ☐ Interacción con un manejador de archivos que manipulan información mediante instrucciones específicas en lenguaje de bajo nivel.
- ☐ Verifica la integridad de la base de datos, mediante la creación y mantenimiento de reglas de consistencia.

- Proporciona seguridad en el acceso de los sistemas mediante passwords y otros medios de validación, identificación y restricción de permisos.
- Ofrece respaldo y recuperación en caso de contingencia, mediante medios automáticos o semiautomáticos para reestablecer el estado original de la base de datos hasta antes del momento de la falla.
- Controla la concurrencia y acceso en un ambiente multiusuario, con el objeto de mantener la integridad y consistencia de la base, determinando en que momento se puede tener acceso o no al código o a los datos por parte del usuario.
- Inspecciona en orden el esquema externo del usuario, la correspondencia externa / conceptual, el esquema conceptual, y por último la correspondencia conceptual / interna además de la definición de la estructura de mantenimiento.
- Ejecuta las operaciones necesarias sobre la base de datos.

Manejadores de Bases de Datos Relacionales.

El modelo relacional ha proporcionado un criterio mediante el cual la mayoría de las aplicaciones comerciales DBMS han sido evaluadas en su funcionamiento y confiabilidad. Virtualmente todos los sistemas DBMS Oracle, DB2, Informix, SQL server, Paradox, Approach, etc. se basan en el modelo relacional y demandan ser RDBMS.

En 1984 Codd publicó 12 reglas que un verdadero sistema relacional debería de cumplir. En la práctica algunas de ellas son difíciles de realizar.

REGLA 0

Para que un sistema se denomine sistema manejador de bases de datos relacionales, este sistema debe usar (exclusivamente) sus capacidades relacionales para administrar la base de datos.

REGLA 1: REGLA DE LA INFORMACIÓN

Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera: con valores en tablas.

REGLA 2: REGLA DEL ACCESO GARANTIZADO

Para todos y cada uno de los datos (valores atómicos) de una BDR se garantiza que son accesibles a nivel lógico utilizando una combinación de nombre de tabla, valor de clave primaria y nombre de columna.

REGLA 3: MANEJO SISTEMÁTICO DE VALORES NULOS

Los valores nulos (que son distintos de la cadena vacía, blancos, 0, etc.) son soportados por los DBMS totalmente relacionales para representar información desconocida o no aplicable de manera sistemática, independientemente del tipo de datos.

REGLA 4: CATÁLOGO DINÁMICO EN LÍNEA BASADO EN EL MODELO RELACIONAL

La descripción de la base de datos se representa a nivel lógico de la misma manera que los datos normales, de modo que los usuarios autorizados pueden aplicar el mismo lenguaje relacional a su consulta, igual que lo aplican a los datos normales.

REGLA 5: REGLA DEL SUBLenguaje DE DATOS COMPLETO

Un sistema relacional debe soportar varios lenguajes y varios modos de uso de terminal (ej: rellenar formularios, etc.). Sin embargo, debe existir al menos un lenguaje cuyas sentencias sean expresables, mediante una sintaxis bien definida, como cadenas de caracteres y que sea completo.

REGLA 6: REGLA DE ACTUALIZACIÓN DE VISTAS

Todas las vistas que son teóricamente actualizables se pueden actualizar por el sistema.

REGLA 7: INSERCIÓN, ACTUALIZACIÓN Y BORRADO DE ALTO NIVEL

La capacidad de manejar una relación base o derivada como un solo operando se aplica no sólo a la recuperación de los datos (consultas), si no también a la inserción, actualización y borrado de datos.

REGLA 8: INDEPENDENCIA FÍSICA DE DATOS

Los programas de aplicación y actividades en terminales permanecen inalterados a nivel lógico en cualquier momento que se realicen cambios en las representaciones de almacenamiento o métodos de acceso.

REGLA 9: INDEPENDENCIA LÓGICA DE DATOS

Los programas de aplicación y actividades en terminales permanecen inalterados a nivel lógico cuando quiera que se realicen cambios a las tablas base que preserven la información.

REGLA 10: INDEPENDENCIA DE INTEGRIDAD

Los limitantes de integridad específicos para una determinada base de datos relacional deben poder ser definidos en el sublenguaje de datos relacional, y almacenables en el catálogo, no en los programas de aplicación.

REGLA 11: INDEPENDENCIA DE DISTRIBUCIÓN

Una BDR tiene independencia de distribución.

REGLA 12: REGLA DE LA NO SUBVERSIÓN

Si un sistema relacional tiene un lenguaje de bajo nivel (un registro de cada vez), ese bajo nivel no puede ser usado para saltarse (subvertir) las reglas de integridad y los limitantes expresados en los lenguajes relacionales de más alto nivel (una relación o conjunto de registros de cada vez).

Sistema de procesamiento de archivos.

El sistema de procesamiento de archivos es otro método para la manipulación de bases de datos que consiste en un conjunto de programas que acceden a la misma, que no optimizan los métodos utilizados durante el acceso provocando, entre otros, los siguientes problemas:

Redundancia.

Se presenta cuando se repiten innecesariamente datos en los archivos que conforman la base de datos. Esta redundancia aumenta los costos de almacenamiento y acceso y además puede llevar a inconsistencias, es decir, datos que no concuerdan entre dos tablas con el mismo campo.

Inconsistencia.

Ocurre cuando existe información contradictoria o incongruente en la base de datos.

Dificultad en el acceso a los datos.

Debido a que los sistemas de procesamiento de archivos generalmente se conforman en distintos tiempos o épocas y ocasionalmente por distintos programadores, el formato de la información no es uniforme y se requiere de establecer métodos de enlace y conversión para combinar datos contenidos en distintos archivos.

Aislamiento de los datos.

Se refiere a la dificultad de extender o actualizar las aplicaciones que permitan controlar a la base de datos, como pueden ser, nuevos reportes, utilerías y demás debido a la diferencia de formatos en los archivos almacenados.

Anomalías en el acceso concurrente.

Ocurre cuando el sistema es multiusuario y no se establecen los controles adecuados para sincronizar los procesos que afectan a la base de datos. Comúnmente se refiere a la poca o nula efectividad de los procedimientos de bloqueo.

Problemas de seguridad.



Se presentan cuando no es posible establecer claves de acceso y seguridad en forma uniforme para todo el sistema, facilitando así el acceso a intrusos.

Problemas de integridad

El problema ocurre cuando no existe a través de todo el sistema procedimientos uniformes de validación para los datos.

Tipos de usuarios de una base de datos.

Entre los usuarios de una base e datos podemos encontrar:

-  El administrador de la base de datos (sa,dba)
-  El programador de aplicaciones, quién se encarga de escribir los programas de aplicación que utiliza la base.

- El usuario final, el cual tiene acceso a los datos de la base a través de alguna aplicación desarrollada o utilizando alguna interfaz proporcionada por el DBMS.

Niveles de abstracción en la información.

Existen tres niveles en los que los datos pueden ser abstraídos o aislados, dependiendo de las características y permisos de cada uno de los usuarios de una base de datos.

Nivel físico.

Dónde se determinan las características y medios de almacenamiento secundarios. Usualmente se deja la tarea al sistema operativo a este nivel de abstracción, debido a la complejidad y al alto nivel técnico requerido a nivel Hardware.

Nivel conceptual.

Es la parte en la que se diseña y crean las estructuras lógicas de la base de datos, esto es las tablas y las relaciones lógicas entre ellas. Los usuarios ubicados en este nivel poseen conocimientos técnicos necesarios para el análisis lógico de los datos y conocen un protocolo o lenguaje para la creación y manipulación de datos (como lo es el SQL).

Nivel de edición.

Es la capa visible hacia el usuario final, permitiéndole la edición o actualización de registros bajo los criterios previamente definidos en el nivel conceptual. Es también utilizado un lenguaje de transacciones para realizar dichas operaciones.

Funciones del administrador de la base de datos (DBA).

El DBA tiene el control centralizado de la base de datos, con la directiva de mantener a integridad y seguridad de los datos. Entre sus principales funciones tenemos:

- Definición del esquema.
- Definición de las estructuras de almacenamiento.
- Modificación de esquema y organización.
- Creación y mantenimiento de los permisos de acceso.

- ▣ Especificación de las limitantes de integridad

Modelos de datos.

Un modelo de datos está constituido por una serie de herramientas conceptuales que describen los datos, las relaciones entre ellos, la semántica y sus limitantes.

Estos se clasifican en tres grupos principalmente:

Modelos lógicos basados en objetos.

Son aquellos que nos permiten una definición clara y concisa de los esquemas conceptuales y de visión. Su característica principal es que permiten definir en forma detallada las limitantes de los datos. Dentro de este tipo de modelos tenemos los siguientes:

- ▣ Modelo entidad relación.
- ▣ Modelo binario
- ▣ Modelo semántico de los datos
- ▣ Modelo infológico

Modelos lógicos basados en registros.

Operan sobre niveles conceptuales y de visión. Sus características principales son que permiten una descripción más amplia de la implantación, pero no son capaces de especificar con claridad las limitantes de los datos. Entre ellos encontramos:

- ▣ Modelo Relacional
- ▣ Modelo de Red.
- ▣ Modelo jerárquico.

Modelos físicos de datos.

Describen los datos en el nivel más bajo y permiten identificar algunos detalles de implantación para el manejo del hardware de almacenamiento. Ejemplos de este tipo de modelos son:

- ▣ Modelo unificador.
- ▣ Modelo de memoria de cuadros.

Independencia de los datos.

La independencia de los datos es la capacidad que ofrecen las bases de datos de mantener el control sobre la información sin importar el tipo de aplicación y el lenguaje en el que esté escrita. Hay dos tipos de independencia de los datos:

- ☐ Independencia física.
- ☐ Independencia lógica.

Componentes de un RDBMS.**Lenguaje de definición de datos (DDL).**

Permite la definición de estructuras dentro de la base de datos y del servidor, como son tablas, vistas, procedimientos, triggers, reglas, defaults, etc.

Lenguaje de manipulación de datos (DML).

Permite manipular los datos almacenados en la base de datos, proporciona instrucciones tales como insert, delete, update, select, commit y rollback.

Lenguaje de control de datos (DCL).

Proporciona seguridad a los datos contenidos en las bases de datos, es decir, proporciona instrucciones tales como grant, revoke, con lo que es posible otorgar o negar permisos.

Diccionario de datos (DD).

Contiene los datos acerca de los datos almacenados en la base de datos, también conocidos como metadatos, contienen información acerca de la estructura y contenido de una base de datos. Contiene referencias a tablas de sistema, tablas de usuario, reglas, defaults, índices, constraints, etc.

En forma comercial existen diversos paquetes y/o lenguajes mediante los cuales se puede construir un modelo relacional. El lenguaje que se considera estándar para este tipo de aplicaciones es el SQL (Structured Query Language), este lenguaje de consulta estructurado proporciona formatos y sintaxis para la manipulación y definición de los datos.

Modelo entidad - relación.

El modelo de datos entidad-relación (E-R) se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos. Fue desarrollado por P. Chen en 1976 para facilitar el diseño de las bases de datos, permitiendo la especificación de un esquema integral, que representa la estructura lógica global de una base de datos. Dentro de sus principales características encontramos:

- ☐ Simplicidad; las tablas se definen en una forma familiar y explicables por sí mismas para representar los datos. No se requiere un entendimiento especial para entender y utilizar los datos.
- ☐ Las tablas diseñadas correctamente mantienen un rigor matemático, es decir, dicen lo que significan y significan lo que dicen. Pueden ser implantadas y procesadas por una variedad de configuraciones de software y hardware.
- ☐ Flexibilidad. Las tablas y relaciones no solamente muestran la estructura física del almacenamiento de los datos, sino la relación lógica de los registros respecto a toda la base de datos.

Entidad.

Puede ser una persona, lugar, evento o un objeto identificado en forma única y del cual se registra información y que además se encuentra dentro del alcance del sistema, por lo que éste debe mantener, correlacionar y desplegar información.

Existen dos tipos de entidades: tangibles e intangibles.

Tabla.

Dentro de un modelo entidad-relación, se utiliza como estructura de almacenamiento de los datos a las tablas. Una tabla presenta las siguientes características:

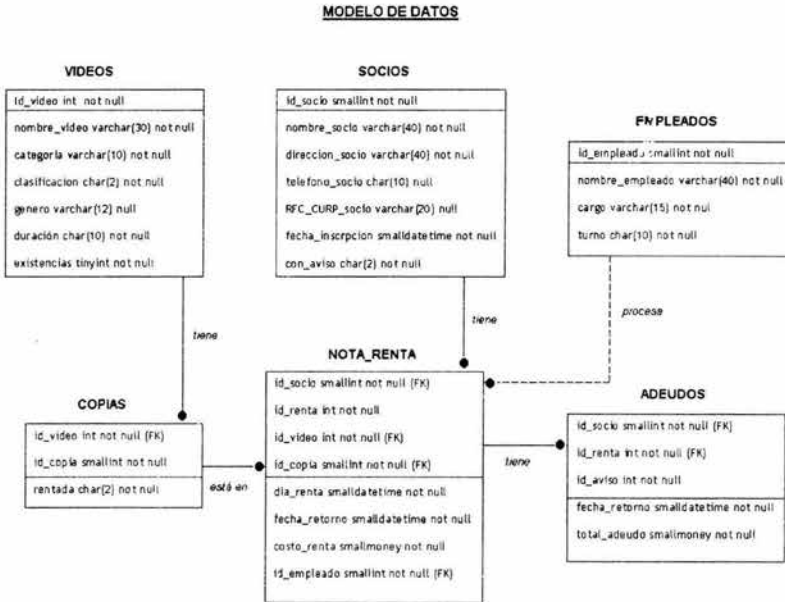
- ☐ Cada tabla debe tener un nombre lógico.
- ☐ No existen ligas o apuntadores entre tablas.
- ☐ El orden de los registros no es significativo.
- ☐ El orden de las columnas no es significativo al aplicarles un alias o nombre de atributo.

- ❑ Todos los valores son indivisibles, esto es, no se pueden descomponer sin pérdidas de información.
- ❑ No debe haber renglones repetidos
- ❑ Debe haber uno o más atributos que identifiquen de manera unívoca cada renglón.

Llaves.

Se denomina llave o clave al atributo que permite significar de manera única a una entidad, es decir, a partir de un campo llave se pueden inferir otros campos de una tabla, por lo que cada tupla o renglón debe estar asociada con una llave que permita su identificación. Existen diversos tipos de llaves que son:

- ❑ **Primaria (PK).** Es el atributo o grupo de atributos que identifican de manera única cada tupla en una tabla.
- ❑ **Candidata.** Es aquel atributo susceptible a ser elegido como llave primaria.
- ❑ **Alterna (AK).** Es aquel atributo que era candidato a ser llave primaria pero no fue seleccionado, sin embargo, puede ser considerado para un futuro.
- ❑ **Foráneas (FK).** Es aquel atributo o grupo de atributos que en la tabla padre forman la llave primaria, y en la tabla hija son un atributo más.



Ejemplo gráfico de un modelo entidad – relación típico.

Bases de datos orientadas a objetos.

El modelo de bases de datos orientado por objetos es una adaptación a los sistemas de bases de datos. Se basa en el concepto de encapsulamiento de datos y código que opera sobre estos en un objeto. Los objetos estructurados se agrupan en clases. El conjunto de clases está estructurado en sub y superclases basado en una extensión del concepto ISA del modelo Entidad - Relación. Puesto que el valor de un dato en un objeto también es un objeto, es posible representar el contenido del objeto dando como resultado un objeto compuesto. La interfaz entre un objeto y el resto del sistema se define mediante un conjunto de mensajes.

Un objeto tiene asociado:

- ☐ un conjunto de variables que contienen los datos del objeto. El valor de cada variable es un objeto.
- ☐ Un conjunto de mensajes a los que el objeto responde.
- ☐ Un método, que es un trozo de código para implementar cada mensaje. Un método devuelve un valor como respuesta al mensaje.

Manejador de base de datos MS SQL server.

El RDBMS MS SQL server es un producto desarrollado por Microsoft, que cuenta con las características necesarias para considerarlos como un manejador de bases de datos relacionales que cumple con las características expuestas hasta el momento en cuanto a la integridad y control de los datos.

El MS SQL Server 2000 es un manejador de bases de datos y análisis que proporciona el rendimiento, la escalabilidad y la confiabilidad que requieren los exigentes entornos Web, desde aplicaciones de intranets hasta grandes aplicaciones Internet con múltiples transacciones electrónicas. Posee compatibilidad con el Lenguaje de marcado extensible (XML, Extensible Markup Language) y el Protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol) simplifican el acceso a los datos y su intercambio, al tiempo que las eficaces capacidades de análisis mejoran el valor de los datos. Las características de disponibilidad mejoradas maximizan el tiempo de actividad, las funciones de administración avanzadas automatizan las tareas rutinarias y las herramientas de programación y los servicios mejorados aceleran el desarrollo.

Otras características destacables del manejador son:

- ☐ Escalado de ampliación. SQL Server 2000 aprovecha los sistemas de multiproceso simétrico (SMP, Symmetrical Multiprocessor), que permite utilizar hasta 32 procesadores y 64 GB de RAM.

- ❑ Escalado de distribución. El escalado distribuye la base de datos y la carga de datos entre servidores.
- ❑ Disponibilidad. SQL Server 2000 consigue la máxima disponibilidad gracias a los clústeres de conmutación por error mejorados, el trasvase de registros y mejoradas estrategias de copia de seguridad.

1.6 Metodología

Modelo de Cascada

Cuando un sistema es diseñado con la intención de que éste sea manejable por un periodo largo de tiempo, se trata de obtener la entropía* más baja posible desde el principio. Este es uno de los objetivos de los métodos para el desarrollo de sistemas. Por este medio se espera incrementar el tiempo de vida del sistema. Se sabe que tarde o temprano un límite será alcanzado más allá del cual será muy caro mantener el sistema.

Para diseñar un buen sistema de software, diferentes métodos han sido propuestos para describir ya sea un proyecto para desarrollar una primera versión de un producto o para dar una vista global del ciclo de vida de un sistema completo. La definición de un "buen" sistema de software varía dependiendo de las aplicaciones. Por ejemplo, en algunos casos es el rendimiento lo que importa, mientras que en otros es más importante la interfaz gráfica con el usuario. También depende de la estructura del sistema, por ejemplo, si esta será distribuida o centralizada. Lo que es común es que todos los sistemas necesitan ser modificados en algún momento.

Tradicionalmente, el trabajo de análisis y diseño es estructurado y descrito usando diferentes tipos de modelos de cascada, ver figura 1.4. Estos modelos describen el proceso de desarrollo del sistema donde la idea inicial es que cada fase deberá completarse antes de iniciar la siguiente. Sin embargo, esto último fue rápidamente abolido y en la actualidad es permitido iniciar una fase antes de completar totalmente la anterior. El modelo de cascada ha tenido un tremendo impacto sobre los métodos de ingeniería de software, aunque esté no se sigue de manera rígida cuando es aplicado.

* La segunda ley de la termodinámica, en principio, dice que el "desorden" de un sistema cerrado no puede reducirse, éste sólo puede incrementarse o posiblemente permanecer sin cambio. Una medida de este desorden se conoce como entropía. Esta ley puede, en cierto sentido, aplicarse a los sistemas de cómputo. Un programa que es usado será modificado. Cuando un programa se modifica, su complejidad se incrementa.

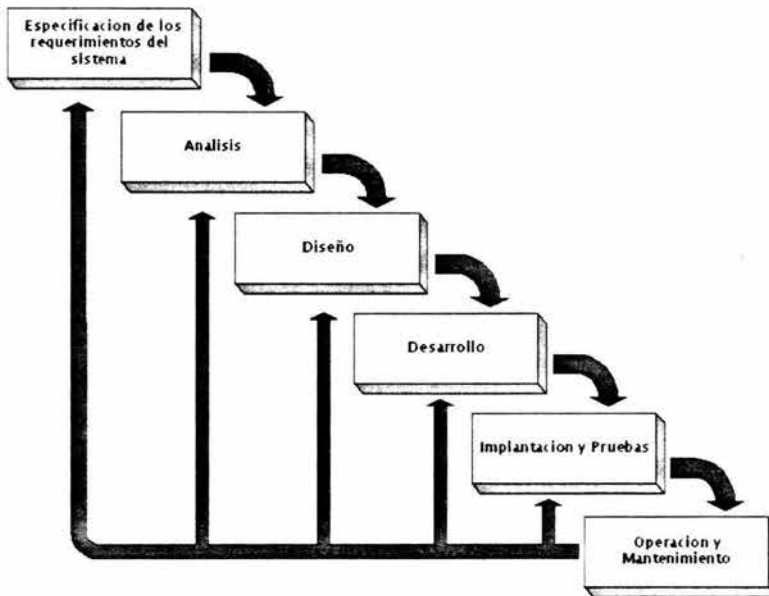


Figura 1.4. Modelo de cascada

En este trabajo de tesis empleamos la metodología de cascada debido a que la herramienta de programación usada en este caso, el lenguaje PHP, es de tipo estructurada y este modelo es ampliamente usado para estos casos así también por que los requerimientos del sistema fueron bien comprendidos y delimitados.

Este método tiene como etapas la de:

- Especificación de los requerimientos del sistema
- Análisis
- Diseño
- Desarrollo
- Implantación y pruebas
- Operación y mantenimiento

En la primera etapa que es la de Especificación de los requerimientos del sistema se llevó a cabo el estudio de que es lo que se pretendía que el sistema hiciera, así como obtener un entendimiento conceptual del problema a resolver y que tipo de herramientas, modelos y estructuras se iban a emplear.

En la etapa de análisis se hizo la definición del problema para así poder plantear los objetivos y posibles mejoras al sistema actual.

En la etapa de diseño una vez conocido el problema, se identifican los distintos procesos computacionales y se define una arquitectura (basada en los requisitos y arquitectura tecnológica) y una implantación. La especificación, está muy relacionada con el software básico específico que se utiliza en la construcción que en este caso corresponde a las herramientas de desarrollo.

En esta etapa de desarrollo se llevo a cabo la creación de los módulos, su programación así como la creación de la base de datos.

En la etapa de implantación y pruebas se hizo la integración del sistema para así poder hacer pruebas pilotos.

En la etapa de operación y mantenimiento se da cuando el sistema está concluido, aprobado por el usuario final y listo para su operación.

1.7 Modelado de la aplicación

UML

La situación actual del desarrollo de software requiere de grandes esfuerzos en la etapa diseño y análisis debido a la complejidad de las aplicaciones que diversos tipos de usuarios requieren hoy en día. Estas aplicaciones deben basarse en modelos del mundo real; y un método que aproxima de una forma evidente a dicho mundo es la metodología orientada a objetos.

El modelado de objetos permite abstraer los sistemas en clases y métodos que son manipulados mediante aplicaciones y lenguajes orientados a objetos, con el objeto de facilitar el control en el diseño e implantación de sistemas de gran magnitud, los cuales resulten imposibles, en muchos casos, de ser comprensibles para una persona.

Otro aspecto importante en la metodología orientada a objetos es la relación existente entre el análisis de requisitos, que es la relación entre la

asignación de software al nivel del sistema y el diseño del software, para lo cual han surgido diversas metodologías como son la OOSE (Object Oriented Software Engineering), OMT (Object Modeling Technique) y el método Booch. Para evitar conflictos entre estos métodos y manejar convenientemente la metodología, fue creado el Lenguaje Unificado de modelado (UML).

La versión 1.0 de UML fue lanzada en Enero de 1997, ha sido utilizado con éxito en todo tipo de industria a todo nivel. El modelado tiene como objetivo principal el de describir la conducta de una aplicación basada en los requerimientos de funcionalidad, desempeño y confiabilidad.

Principales beneficios de UML.

- ☐ Mejores tiempos totales de desarrollo (reducidos en un 50 %).
- ☐ Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- ☐ Establecer conceptos y artefactos ejecutables.
- ☐ Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- ☐ Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- ☐ Mejor soporte a la planeación y al control de proyectos.
- ☐ Alta reutilización y minimización de costos.
- ☐ Básicamente UML describe al modelo mediante una colección de vistas, diagramas, símbolos y reglas que constituyen un lenguaje de modelado. Cada uno de estos elementos describen un aspecto específico del producto o sistema en construcción.

Etapas de desarrollo

UML utiliza diversos elementos del lenguaje de modelado para diversas etapas de desarrollo de sistemas, estas etapas son:

- ☐ Análisis de Requerimientos
- ☐ Análisis
- ☐ Diseño
- ☐ Programación
- ☐ Pruebas

Casos de uso

Un caso de uso es un escenario típico de interacción entre el usuario y la aplicación. Esta interacción puede parecer de repente demasiado sencilla u obvia, por lo cual es usualmente omitida en un esquema de documentación, y por consiguiente, no es considerada como parte de una planeación plenamente constituida.

Estas interacciones o casos de uso tienen como características las siguientes:

- ▣ El caso de uso capta alguna función visible para el usuario.
- ▣ El caso de uso puede ser pequeño o grande.
- ▣ El caso de uso logra un objetivo discreto para el usuario.

Para lograr esto es necesario analizar la funcionalidad deseada en el sistema directamente con los usuarios, para entonces definir las interacciones requeridas.

Diagramas de casos de usos

Las interacciones están representadas en un diagrama de casos de usos mediante actores que son la representación de los usuarios, y los casos de uso que son los elementos a los que tiene acceso el actor en el sistema.

Es posible obtener a los actores de un diagrama de casos de uso a través de las siguientes preguntas:

- ▣ ¿Quién utilizará la funcionalidad principal del sistema (actores primarios)?
- ▣ ¿Quién necesitará soporte del sistema para realizar sus actividades diarias?
- ▣ ¿Quién necesitará mantener, administrar y trabajar el sistema (actores secundarios)?
- ▣ ¿Qué dispositivos de hardware necesitará manejar el sistema?
- ▣ ¿Con qué otros sistemas necesitará interactuar el sistema a desarrollar?
- ▣ ¿Quién o qué tiene interés en los resultados (los valores) que el sistema producirá?

Se recomienda que solo se deben mostrar aquellos actores del sistema que necesiten de un caso de uso, para así facilitar la comprensión y el diseño de los requerimientos.

Un actor además de ser un ser humano, puede ser un sistema externo que necesite de cierta información, o requiera de ciertas acciones del sistema actual.

Interacción de los casos de uso

Un caso de uso representa la funcionalidad completa tal y como la percibe un actor. Un caso de uso en UML es definido como un conjunto de secuencias de acciones que un sistema ejecuta y que permite un resultado observable de valores para un actor en particular. Gráficamente se representan con una elipse y tiene las siguientes características:

- ☐ Un caso de uso siempre es iniciado por un actor.
- ☐ Un caso de uso provee valores a un actor.
- ☐ Un caso de uso es completo.

Encontrando casos de uso

El proceso para encontrar casos de uso inicia encontrando al actor o actores previamente definidos. Por cada actor identificado, hay que realizar las siguientes preguntas:

- ☐ ¿Qué funciones del sistema requiere el actor? ¿Qué necesita hacer el actor?
- ☐ ¿El actor necesita leer, crear, destruir, modificar o almacenar algún tipo de información en el sistema?
- ☐ ¿El actor debe ser notificado de eventos en el sistema o viceversa? ¿Qué representan esos eventos en términos de funcionalidad?
- ☐ ¿El trabajo diario del actor podría ser simplificado o hecho más eficientemente a través de nuevas funciones en el sistema? (Comúnmente, acciones actuales del actor que no estén automatizadas)
- ☐ Otras preguntas que nos ayudan a encontrar casos de uso pero que no involucran actores son:
 - ☐ ¿Qué entradas/salidas necesita el sistema? ¿De dónde vienen esas entradas o hacia dónde van las salidas?
 - ☐ ¿Cuáles son los mayores problemas de la implementación actual del sistema?

Ejemplo práctico del diagrama de casos de usos

A continuación se presenta el diagrama 1.5 de casos de uso propuesto para el sistema de encuestas.

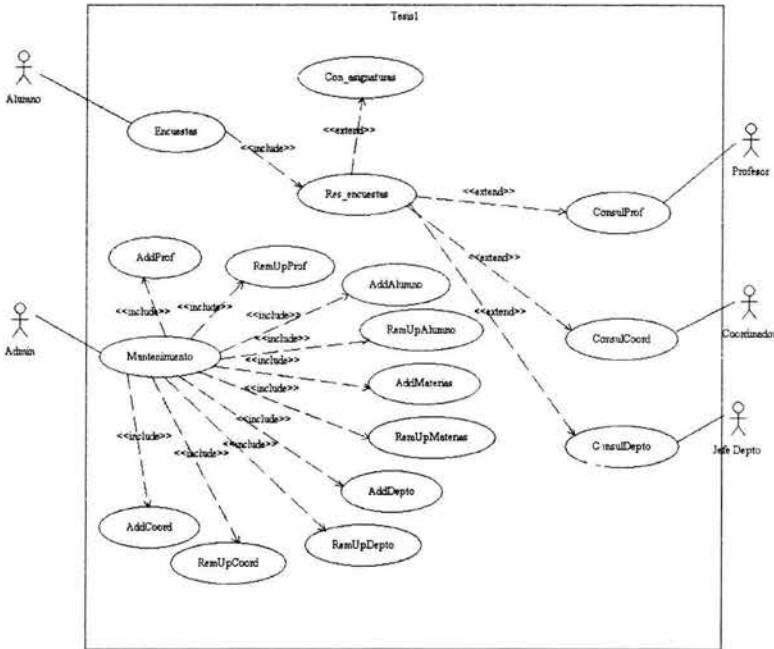


Figura 1.5. Casos de uso.

Este diagrama propone la existencia de 5 actores, que son el alumno, el administrador, el profesor, el coordinador de carrera y el jefe de departamento. Estos interactúan con diversas partes o módulos del sistema, los cuales conforman los casos de uso.

Se incluyen casos de uso principales, o de acceso directo al actor, y casos de uso intermedios o adicionales, que utilizan algunas de las características de las anteriores, o bien pueden tener alguna funcionalidad adicional.

Para relacionar las interacciones entre casos de uso, se utilizan los vínculos uses o include y extends.

El vínculo include, se refiere a la posibilidad de utilizar una porción de comportamiento de un caso de uso a otro, por ejemplo, la actualización de alumnos y profesores son comportamientos similares en el caso de uso mantenimiento, al cual tiene acceso el actor administrador.

El vínculo extends, se refiere a un caso de uso similar a otro, pero que puede tener funcionalidad adicional. Para el caso de nuestro diagrama, en el caso de uso de res_encuestas, que tiene acceso a la base de datos con los resultados, la consulta del profesor podría ser presentada también al jefe del departamento, pero con la posibilidad para éste, de observar el resultado de varios profesores y confrontarlos.

En general se utilizan extends cuando se describan variaciones de una conducta o requerimiento normal, y se utiliza include cuando se trate de uno o varios casos de uso y se desee evitar repeticiones.

Escenarios

Un escenario en UML consiste en una ruta específica dentro de un diagrama de casos de uso en UML, mostrando una particular combinación de circunstancias o condiciones que requieran ser indicadas como importantes y críticas para posteriores etapas en el desarrollo del modelo.

Realizaciones

Tanto los diagrama de casos, como la concepción del modelo en general tiene un una diversidad de concepciones y abstracciones. Esto varía dependiendo del estilo y experiencia del analista. A cada una de las propuestas de modelado de UML se le conoce comúnmente como realización, y usualmente son creadas varias realizaciones desde diversos puntos de vista, para así confrontarlos y llegar a un modelo mucho más representativo.

CAPÍTULO 2

Definición del problema y análisis

- 2.1 Definición del problema
- 2.2 Sistema propuesto
 - Objetivos
- 2.3 Esquemas de operación
 - Esquema actual
 - Esquema propuesto
 - Mejoras al actual esquema
- 2.4 Método empleado para la solución
- 2.5 Etapas de Desarrollo de la aplicación
 - Servidor de Base de datos y Base de datos (Back-End tier)
 - Diseño y construcción de la base de datos en SQL Server
 - Servidor de Web y de aplicaciones (Middle tier)
 - Cliente (Front - End tier)
- 2.6 Diseño WEB
- 2.7 Diseño y análisis de la encuesta
- 2.8 Evaluación del sistema

2.1 Definición del problema

El sistema de encuestas de profesores que se aplica al final de cada semestre, denominado “Programa de evaluación del proceso enseñanza – aprendizaje”, tiene por objeto que los alumnos de la facultad evalúen bajo su criterio y de manera anónima, el desempeño tanto del profesor del grupo como del suyo propio; en cada una de las asignaturas en las que se encuentran inscritos.

Actualmente este proceso se lleva a cabo mediante el llenado de hojas ópticas por parte de los alumnos, que son procesadas para la posterior generación de reportes, cuyos resultados son entregados a los profesores en el siguiente semestre.

Este proceso genera gastos para la facultad tanto en el empleo de recursos humanos como de recursos materiales; dichos gastos pueden ser disminuidos con la implantación del sistema informático propuesto en este proyecto. Esto aunado a la posibilidad de obtener resultados de forma inmediata.

El sistema actual ofrece prestaciones limitadas en cuanto a los resultados recabados, siendo necesario ampliar las posibilidades de manejo de información del sistema; para que profesores y responsables académicos de cada una de las divisiones de la facultad, cuenten con los datos estadísticos adecuados que permitan hacer más acertada y pertinente la toma de decisiones, en busca del mejoramiento en las condiciones del proceso educativo.

2.2 Sistema propuesto

Con el presente trabajo de tesis planteamos la construcción de un sistema prototipo que eventualmente sustituya la manera en que actualmente se realiza la evaluación del proceso enseñanza – aprendizaje de la Facultad de Ingeniería, mediante su control utilizando tecnologías Web.

Objetivos:

Diseñar e implantar un sistema de información con carácter de programa piloto que facilite y optimice el actual proceso de evaluación de enseñanza–aprendizaje en el departamento de Ingeniería en Computación de la Facultad de Ingeniería.

- Diseñar y manejar de una base de datos con la información recabada de los alumnos de la Facultad mediante las encuestas.
- Recabar, analizar y presentar los resultados de las encuestas vía Internet mediante una aplicación Web.
- Ofrecer los reportes requeridos por parte de profesores, jefes de departamento y coordinadores de carrera, respecto a la información recabada en las encuestas, empleando dicha aplicación.
- Generar estadísticas tanto del semestre en curso como del histórico de los semestres anteriores.

2.3 Esquemas de operación

Esquema actual

La manera en la que se realiza la encuesta actualmente está gráficamente ejemplificada en la figura 2.1. En la cual se llevan a cabo los siguientes pasos en este proceso:

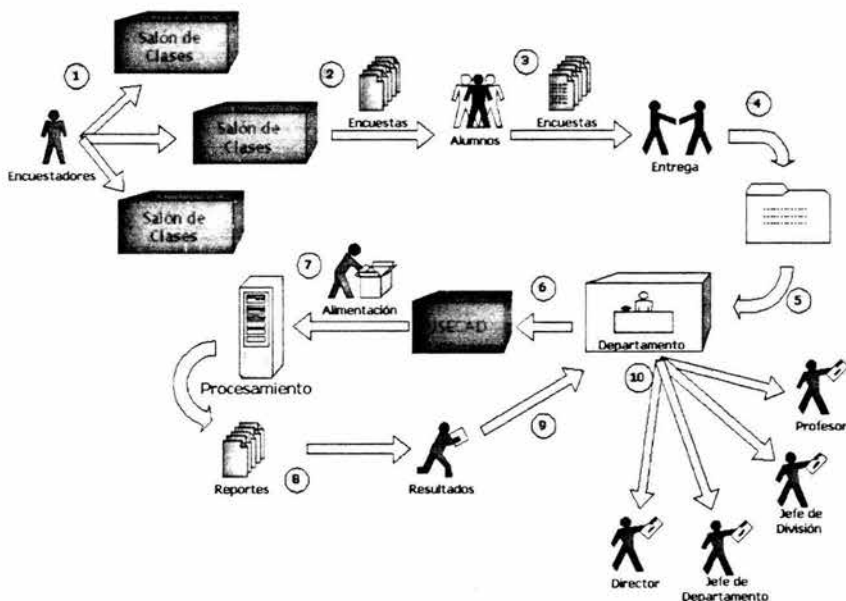


Figura 2.1. Esquema actual

1. Se designan un grupo de personas que visitarán los salones en los horarios de clase, para cubrir todas las asignaturas de la Facultad.
2. Hace entrega de las encuestas impresas tipo ópticas a los alumnos presentes.
3. Los alumnos llenan los datos correspondientes y la encuesta.
4. Se guardan las encuestas en un sobre cerrado y firmado por el profesor.
5. Se llevan las encuestas a los departamentos correspondientes.
6. Los departamentos hacen entrega de los sobres que contienen las encuestas a USECAD.

7. En USECAD se alimenta la unidad lectora óptica para que procese los resultados de la encuesta
8. Se generan los respectivos reportes.
9. Se entregan los reportes de resultados a los departamentos.
10. Y estos a su vez los distribuyen a las personas interesadas.

Esquema propuesto

El sistema Web propuesto para realizar la evaluación del proceso enseñanza – aprendizaje de la Facultad de Ingeniería, que se ejemplifica en la figura 2.2, reduce los procesos a los siguientes pasos:

1. Dentro de un período de tiempo establecido, los alumnos acceden desde cualquier locación remota a la página Web de la encuesta.
2. Acceden mediante una clave y contraseña para llenan las encuestas que les correspondan.
3. El servidor guarda la información en una base de datos y genera los reportes en línea.
4. Una vez realizadas las encuestas se puede acceder a los reportes vía remota mediante el ingreso al sistema Web por Internet.

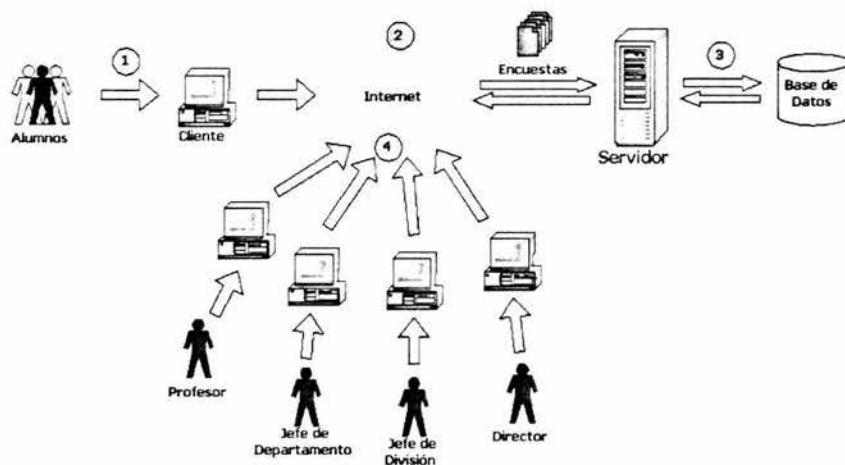


Figura 2.2. Esquema propuesto

Mejoras al actual esquema

Entre las mejoras que se efectuaran al actual esquema de trabajo para obtener la información de las encuestas del programa de evaluación del proceso enseñanza aprendizaje; se observan las siguientes:

- ☐ La captura de información será más confiable.
- ☐ No interrumpe el libre desarrollo de la clase ya que el alumno realizará la encuesta de manera externa al tiempo de desarrollo de la clase, y puede ser en casa o en cualquier otro lugar donde tenga acceso a Internet.
- ☐ Se puede contar de manera rápida con la información que arroja la encuesta.
- ☐ Se pueden tener resultados en línea.
- ☐ La información procesada se puede distribuir vía electrónica.
- ☐ La muestra de alumnos podrá ser el 100% de la comunidad estudiantil.
- ☐ La información se encontrará alojada en una base de datos, para su posterior análisis.
- ☐ Se llevará registros históricos de los resultados.
- ☐ Ahorro de recursos.
- ☐ Flexibilidad al tratar de incluir o cambiar los contenidos de la encuesta, para adecuarlos a las necesidades actuales de la Facultad.

2.4 Metodología empleada para la solución

El sistema será diseñado bajo una arquitectura cliente servidor, en un ambiente de red, y con prestaciones Web, teniendo una base de datos relacional que almacene todos los recursos informáticos recabados en las encuestas.

La base de datos será diseñada tomando en consideración todas las reglas de las bases de datos relacionales en cuanto al acceso, seguridad e integridad de los datos (reglas de Codd), bajo un modelo de entidad - relación.

El sistema deberá correr en plataforma Windows, bajo el sistema operativo Windows 2000. El manejador de base de datos a utilizar será el SQL Server de Microsoft.

La aplicación será diseñada bajo el concepto del lenguaje unificado de modelado (UML, Unified Modeling Language), y será programada utilizando el lenguaje PHP (Hypertext Preprocessor) versión 4, el cual ofrece la generación dinámica de contenidos sobre una arquitectura cliente servidor en un ambiente Web requeridos para el proyecto, además de utilizar funciones primitivas para la conexión a la base de datos.

Adicionalmente de PHP que es un lenguaje embebido en documentos HTML, se requerirá utilizar otros recursos como SQL (Structured Query Language), JavaScript y HTTP (HyperText Transfer Protocol).

2.5 Etapas de Desarrollo de la aplicación

La solución propuesta para el Sistema prototipo para el control vía Web de encuestas para el programa de evaluación del proceso de enseñanza – aprendizaje de la Facultad de Ingeniería, se compone de las diferentes etapas que se muestran en la figura 2.3.

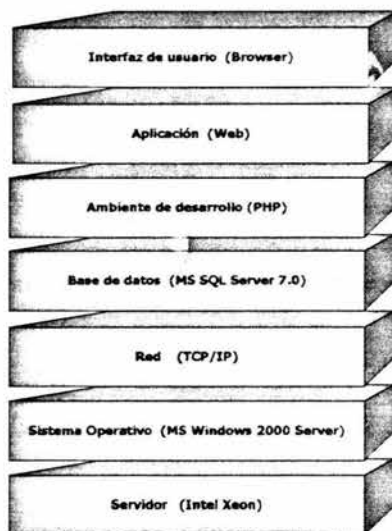


Figura 2.3. Etapas de la aplicación

Comenzaremos describiendo las etapas de desarrollo de la aplicación basándonos en la lógica seguida en el modelo de arquitectura cliente servidor de tres capas, anteriormente descrito en el capítulo 2. Las cuales se describen a continuación, mencionando la solución propuesta:

Servidor de Base de datos y Base de datos (Back - End tier)

Primeramente describiremos la tercer capa conocida como back end, en la cual se encuentra el servidor de base de datos y la base de datos.

Se empleo como servidor de BD a Microsoft SQL Server 2000 que es el eje principal de la administración y análisis de datos siguiente generación de productos y servicios Web en el S.O. Windows 2000.

Diseño y construcción de la base de datos en SQL Server

Una vez definido y establecido el diseño final del sistema; la construcción de la Base de datos (BD) se realizó a través del manejador relacional de base de datos, empleando recursos como: SQL Server Enterprise Manager y el SQL Query Analyzer. La primera de ellas es una herramienta gráfica (como se muestra en la figura 2.4) que facilita la configuración y administración del Microsoft SQL Server y sus objetos, proporcionando:

- Un engine o motor de planificación.
- Capacidad de alerta a nivel administrador.
- Operaciones de control drag-and-drop a través de múltiples servidores.
- Administración de usuarios y permisos.
- Administración de BD, tablas, vistas, procedimientos almacenados, triggers, índices, reglas, tipos de datos, etc.

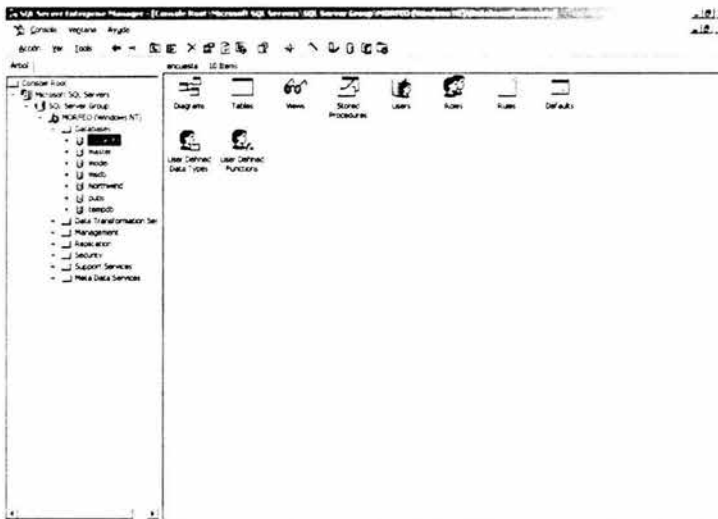


Figura 2.4. SQL Server Enterprise Manager.

Mientras que el Microsoft® SQL Query Analyzer es también una herramienta gráfica que posee una línea de comandos (como se puede ver en la figura 2.5) la cual permite entre otras cosas:

- Creación de queries y de otros scripts SQL que se ejecutan en la BD.

- Rápida creación de objetos predefinidos comúnmente usados en la BD (Templates).
- Depurador de procedimientos almacenados (T-SQL Debugger)
- Rápida inserción, actualización o borrado de columnas en las tablas.



Figura 2.5. SQL Query Analyzer

Se empleó el DBMS de Microsoft® SQL Server 7.0, principalmente por ser uno de los principales requerimientos que se presentaron por parte del Laboratorio Microsoft ya que el servidor en el que se montó físicamente la aplicación se encuentra en dicho laboratorio. Además de que éste DBMS puede iniciar automáticamente como un servicio, cada vez que arranca el sistema operativo, mediante el SQL Server Service Manager, el cual tiene una ventana de control que se activa al dar un doble clic en el icono correspondiente al SQL Server el cual se localiza en el administrador de tareas de la barra de herramientas del sistema operativo Windows 2000 Server. Una vez abierta la ventana que se muestra en la figura 2.6. Se pueden configurar los siguientes servicios del servidor de Base de Datos:

- SQL Service
- SQL Service Agent

- Coordinador de transacciones distribuidas de Microsoft.

Aquí se pueden iniciar, pausar, para o refrescar los servicios; así como elegir la opción de auto arranque cuando el sistema operativo inicie.

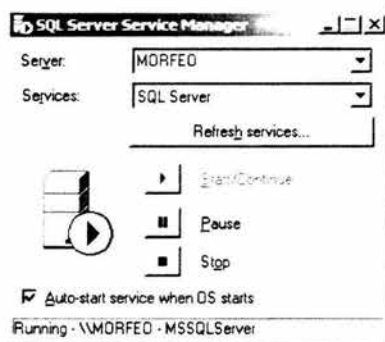


Figura 2.6. SQL Server Service Manager

Servidor de Web y de aplicaciones (Middle tier)

En el modelo de arquitectura cliente servidor de tres capas tenemos como capa media correspondiente al servidor Web y de aplicaciones. El primero de ellos es el servidor incorporado en el S.O. empleado y que es Microsoft Internet Information Server también conocido como (IIS) por sus siglas. El cual se instala de manera predeterminada en Windows 2000 Server y que tiene como características el ser una herramienta personalizable de administración con opciones flexibles de administración y herramientas de análisis capaz de alojar sitios Web que reciben millones de visitas diarias.

El IIS originalmente está configurado para la tecnología propietaria de Microsoft ASP (Active Server Page) por lo cual hay que configurar el IIS para que pueda soportar documentos PHP.

Para esto si instaló la última versión de PHP, disponible en www.php.net que son el archivo binario PHP 4.2.3. una vez hecho esto se siguieron los siguientes pasos de configuración:

1. Copiar las .dlls necesarias (básicamente msvcrt.dll, php4ts.dll y mssql.dll) en la carpeta c:\WINNT\System (manteniendo php4isapi.dll en la carpeta c:\php)
2. Ir al Administrador de Servicios de Internet.

Inicio -> Configuración -> Paneles de control -> Herramientas Administrativas

3. Buscamos el ítem de nuestro servidor Web.
4. Con el botón derecho, seleccionar en el menú emergente Propiedades
5. Ir a la pestaña "ISAPI filters", y Añadir un nuevo filtro. Como nombre le pondremos "PHP" y como ejecutable, "C:\php\php4isapi.dll" (o dondequiera que hayamos instalado dicha dll).
6. Ir a la pestaña "Home Directory". En la sección de "Application Settings", pulsar el botón "Configuration...". Añadir un nuevo "Application Mapping": el ejecutable es el mismo de antes: "C:\php\php4isapi.dll". La extensión es ".php". Marcar "All verbs". Verificar que esté marcado "Script Engine".
7. Modificar las siguientes líneas en el archivo php.ini que se encuentra en C:\WINNT:

```
; cgi.force_redirect configuration below
doc_root = C:\Inetpub\wwwroot
; Directory in which the loadable extensions (modules) reside.
extension_dir = c:/PHP
enable_dl = On
extension=php_mssql.dll
```

8. Aceptar los cambios y reiniciar el servidor Web.

Una vez hecho lo anterior ya podemos alojar nuestras páginas en el la ruta del servidor c:\Interpub\wwwroot y así el motor de PHP podrá interpretarlas.

Cliente (Front - End tier)

En el modelo de arquitectura cliente servidor de tres capas se define como primer capa o (front – end tier) la del cliente en el cual se encuentra la conexión a las otras capas mediante Internet y algún navegador.

2.6 Diseño WEB

En los inicios de Internet, la construcción de un sitio Web era una tarea más o menos simple. Constaba de unas cuantas páginas enlazadas entre sí de forma sencilla, y cada una de ellas estaba formada por un código HTML básico, unas pocas imágenes y poco más.

Pero con el paso del tiempo las exigencias del mercado hicieron aparecer más y más lenguajes de programación basados en los protocolos TCP/IP, y especialmente en el HTTP, a la par que se introdujeron en el mundo Web las tiendas virtuales y la banca electrónica, demandando sitios Web capaces de poder operar con bases de datos, con transacciones y con una interminable serie de nuevas aplicaciones concebidas para estos propósitos.

Esto ha hecho que aquellos inicialmente simples sitios Web se conviertan en verdaderos monstruos de código, formados por multitud de páginas repletas de tablas, formularios y códigos de lenguajes de programación de todo tipo. Se ha pasado de lo simple a lo complejo, proceso que continúa imparablemente.

Y al cambiar el concepto de los sitios Web también ha cambiado paralelamente la forma de planificarlos y de estructurarlos, siendo necesario ahora un complejo estudio de los mismos, con objeto de crear un ente complejo y funcional, bien estructurado, capaz de facilitar a los usuarios finales una navegación clara y una operatividad perfectamente definida.

Uno de los temas sobre los que más se habla actualmente en la Web y en las revistas especializadas es el de la “usabilidad”, debido al ruido que está produciendo. Resulta que ahora, después de tanto tiempo creando páginas Web, se le dice a los diseñadores y desarrolladores que su trabajo no es

todo lo correcto que se debiera, que no están al día de las fabulosas teorías sobre navegabilidad, accesibilidad.

A la hora de crear un sitio Web (conjunto de páginas unidas lógicamente que persiguen un fin común) nos dicen que debemos diseñar de tal forma que el resultado sea perfectamente presentable para todo tipo de público (adaptando las páginas a las condiciones físicas de cada uno), todo tipo de resoluciones, todo tipo de navegadores (aunque sean de sólo texto), todo tipo de conexiones a Internet, etc. estado actual de los equipos informáticos es estupendo: procesadores de altas prestaciones, tarjetas con aceleradores de gráficas, equipos de sonido de multitud de voces, memorias de muchos Megas, monitores de alta resolución, servidores Web capaces de soportar gran número de peticiones simultáneas. Y lo mismo ocurre con el software: sistemas operativos de elevadas prestaciones multimedia, navegadores de 4-5 generación capaces de renderizar casi todo, plugins variados que permiten visualizar y oír casi cualquier cosa, etc.

Pero nos dicen que la mayoría de los navegantes no poseen estas tecnologías avanzadas, que no se pueden crear páginas basadas en conceptos demasiado modernos.

Según ellos debemos diseñar una página Web con los 256 colores seguros o estándares con poco contenido gráfico y muy simple, sin usar lenguajes de script muy avanzados y sin objetos multimedia que precisen plugin alguno. También deberemos usar fuentes multiplataforma, como Times New, de gran tamaño y si es posible, de color negro sobre fondo blanco.

Ahora bien, si acudimos a cualquiera de los servidores Web que ofrecen estadísticas sobre las visitas recibidas podremos sacar las siguientes conclusiones:

- Un 80-90% de los usuarios de Internet poseen como sistema operativo Windows NT, 2000, Millenium o 98 (por mucho que nos pese sistemas Unix-Linux tienen una representación mínima).
- El navegador más usado en Internet Explorer 5-5.5 (aproximadamente un 70 % de los usuarios), seguido por Netscape Navigator 4x (sobre un 25 %).

- La resolución estándar en la actualidad es 800 x 600 píxel (sobre un 60%), y después 1024 x 768 (sobre un 25%), siendo casi nulo el número de usuarios con 640x400.
- En cuanto a la profundidad de color, aproximadamente un 90% de los sistemas están configurados a 24 bits (color real) e incluso a 32 bits.
- Y en lo que toca a las conexiones a Internet, aunque sigue primando el modem de 56 Kbs en el sector de usuarios domésticos (sobre el 50%), es muy cierto que se está produciendo un aumento considerable de las conexiones ADSL (sobre el 30%), RDSI (sobre el 7%) y cable y otras (sobre el 12%), tanto en las empresas como en los hogares, debido sobre todo a la disminución de precios de instalación y a las tarifas planas. Y este aumento seguirá imparabable (Datos extraídos de PC Wordl on-line).

2.7 Diseño y análisis de la encuesta

La encuesta del programa de evaluación del proceso enseñanza – aprendizaje de la Facultad de Ingeniería, como ya se mencionó está dividida en tres principales secciones, una sobre el profesor, otra sobre el alumno y la tercera comentarios y sugerencias.

A continuación se presentan los reactivos de la encuesta:

a) *SOBRE EL PROFESOR*

1.- El interés del profeso para que los alumnos aprendan es:

- muy grande
- grande
- escaso
- nulo

2.- La confianza que el profesor inspira en los alumnos para intervenir en clase es:

- muy grande
- grande
- escaso
- nulo

3.- Las clases se desarrollan de manera:

- muy interesante
- interesante
- aburrida
- muy aburrida

4.- El profesor expone la clase en forma clara:

- siempre
- frecuentemente
- algunas veces
- nunca

5.- Los conocimientos del profesor sobre la asignatura son:

- excelente
- buenos
- regulares
- deficientes

6.- El profesor propicia que los alumnos amplíen por su cuenta sus conocimientos sobre la asignatura:

- siempre
- frecuentemente
- algunas veces
- nunca

7.- El profesor propicia que los alumnos razonen sobre los temas de la asignatura:

- siempre
- frecuentemente
- algunas veces
- nunca

8.- El profesor promueve un ambiente de respeto en clase:

- siempre
- frecuentemente
- algunas veces
- nunca

9.- Las tareas, trabajos y/o ejercicios que deja el profesor, contribuyen al aprendizaje:

- siempre
- frecuentemente
- algunas veces
- nunca

10.- Al exponer los temas el profesor fomenta la habilidad de plantear y resolver problemas:

- siempre
- frecuentemente
- algunas veces

- nunca

11.- El profesor llega generalmente al salón de clase:

- a la hora señalada
- con un retraso de 10 a 20 min.
- con un retraso de 21 a 30 min.
- con un retraso de más de 30 min.

12.- El profesor regresa los exámenes, tareas, trabajos y proyectos corregidos:

- siempre
- frecuentemente
- algunas veces
- nunca

13.- Los exámenes y/o evaluaciones tiene un grado de dificultad adecuado:

- siempre
- frecuentemente
- algunas veces
- nunca

14.- El profesor termina su clase:

- a la hora señalada
- después de la hora señalada
- entre 5 y 10 min.
- Más de 10 min. antes

15.- El profesor entrega oportunamente las calificaciones:

- siempre
- frecuentemente
- algunas veces
- nunca

16.- El profesor faltó a clase durante el semestre:

- nunca
- de 1 a 3 veces
- de 4 a 6 veces
- más de 6 veces

17.- La clase se da sin perder de vista los puntos adonde se quiere llegar:

- siempre
- frecuentemente
- algunas veces
- nunca

18.- Al definir principios o conceptos se presentan ejemplos que facilitan la comprensión:

- siempre
- frecuentemente
- algunas veces
- nunca

19.- En caso de tener la oportunidad ¿volverías a tomar clase con el profesor?

- definitivamente si
- probablemente si
- probablemente no
- definitivamente no

20.- Al inicio del curso de dieron a conocer los objetivos:

- si
- no

b) SOBRE EL ALUMNO (autoevaluación)

21.- Falté a clases durante el semestre:

- nunca
- de 1 a 3 veces
- de 4 a 6 veces
- más de 6 veces

22.- Por lo general al asistir a clase llego:

- a la hora señalada
- con un retraso de 10 a 20 min.
- con un retraso de 21 a 30 min.
- con un retraso de más de 30 min.

23.- Para esta asignatura, estudio por mi cuenta a la semana:

- menos de 1 hora
- entre 1 y 2 horas
- entre 2 y 3 horas
- más de 3 horas

24.- Participo activamente en clase (preguntado, poniendo atención, aportando ejemplos, etc.)

- siempre
- frecuentemente
- algunas veces
- nunca

25.-Me siento motivado para estudiar la asignatura:

- siempre
- frecuentemente
- algunas veces
- nunca

26.- Mi aprendizaje de los temas de la asignatura ha sido:

- excelente
- bueno
- regular
- deficiente

27.- Realizo las tareas y trabajos que me asigna el profesor:

- siempre
- frecuentemente
- algunas veces
- nunca








28.- La actitud del grupo ayudó a mejorar mi desempeño:

- siempre
- frecuentemente
- algunas veces
- nunca

29.- Mi condición académica en esta asignatura es:

- primera inscripción
- segunda inscripción
- sin derecho a inscripción ARTICULO 27
- sin derecho a inscripción ARTICULO 19

Para su análisis los reactivos de la encuesta fueron divididos en rubros que engloban ciertos conceptos de interés como lo son:

-  Aprendizaje
-  Motivación
-  Evaluación
-  Responsabilidad
-  Desempeño del alumno
-  Responsabilidad del alumno
-  Tiempo de estudio del alumno

Aprendizaje se refiere a como a los conocimientos del profesor sobre la asignatura y como éste propicia un ambiente de respeto y el razonamiento de los alumnos sobre los temas expuestos además en la evaluación de éste rubro se estableció la manera en la que los alumnos perciben los criterios de evaluación acordados para el curso, tales como tareas, trabajos, ejercicios, proyectos, etc. Así como el nivel y análisis de los mismos.

Es decir, una respuesta afirmativa en estos reactivos, sugiere un alumno que considera que con ese profesor se aprende, que sabe, que respeta y es respetable. En conclusión, *es un profesor con el que se va a aprender.*

Los reactivos que cuantifican este rubro son: 5, 7, 8, 9 y 10.

La motivación en el curso se cuantificó, con los reactivos mediante la manera en que se desenvuelve la clase, la exposición y conocimiento de los temas, es decir, en este rubro se refleja el fondo y forma de cómo el profesor lleva a cabo su cátedra. Y si el *profesor es una persona que impulsa a estudiar y da su lugar a los alumnos.* Las preguntas correspondientes a esta sección son: 1, 2, 3, 4 y 6.

El concepto de evaluación trata de cómo el profesor distribuye su tiempo para el desarrollo de la clase, la manera en la que se estructura la clase siguiendo los objetivos de la misma. Es decir, estos reactivos representa la

dimensión de cómo el *profesor cuida los asuntos relativos al trabajo y rendimiento escolar*. Las preguntas correspondientes son: 12, 13 y 15.

Finalmente en lo que corresponde a la responsabilidad del profesor se midió lo referente a la asistencia y puntualidad del profesor y los reactivos que reflejan esto son: 11, 14 y 16.

Existen preguntas adicionales que están sin que se haya determinado a que rubro o concepto se van a agrupar y estas son: 17, 18, 19 y 20

Con lo que respecta al inciso de autoevaluación por parte de los alumnos, el análisis de los reactivos se llevó a cabo de la siguiente manera.

El desempeño del alumno se enfoca a la interacción del alumno con el grupo, al aprendizaje de los temas de la asignatura y a la participación del alumno. Los reactivos encargados de medir este factor son: 24, 25, 26, 27, y 28.

La responsabilidad del alumno se refiere a la asistencia y puntualidad de este a clases y las preguntas correspondientes son 21 y 22.

Finalmente tenemos el tiempo de estudio que el alumno dedica a los temas de la asignatura y tiene como único reactivo al 23 y que por sus características no representa un peso para la calificación global.

Basados en lo anterior comenzaremos la construcción del prototipo del sistema.

2.8 Evaluación del sistema

Modelo COCOMO

Creado por Barry Boehm en 1981. Su nombre significa CONstructive COst MOdel (Modelo constructivo de costo) y se puede dividir en tres modelos. COCOMO básico. Calcula el esfuerzo y el costo del desarrollo en función del tamaño del programa estimado en LOC.

COCOMO intermedio. Calcula el esfuerzo del desarrollo en función del tamaño del programa y un conjunto de conductores de costo que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

COCOMO detallado. Incorpora las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, desarrollo, etc.) del proceso.

Los modelos COCOMO están definidos para tres tipos de proyectos de software:

Orgánicos.

Proyectos pequeños y sencillos.

Equipos pequeños con experiencia en la aplicación.

Requisitos poco rígidos.

Semiacoplados.

Proyectos de tamaño y complejidad intermedia.

Equipos con variados niveles de experiencia.

Requisitos poco o medio rígidos.

Empotrados.

Proyectos que deben ser desarrollados con un conjunto de requisitos (hardware y software) muy restringidos.

COCOMO básico

Las ecuaciones del modelo COCOMO básico son de la forma:

$$E = a * KLOC^b$$

$$D = c * E^d$$

Donde: E es el esfuerzo aplicado en hombre-mes, D es el tiempo de desarrollo en meses y KLOC es el número de miles de líneas de código

estimado para el proyecto. Los coeficientes a y c y los exponentes b y d se obtienen de la siguiente tabla:

| Tipo de proyecto | a | b | c | d |
|------------------|-----|------|-----|------|
| Orgánico | 2.4 | 1.05 | 2.5 | 0.38 |
| Semiacoplado | 3.0 | 1.12 | 2.5 | 0.35 |
| Empotrado | 3.6 | 1.20 | 2.5 | 0.32 |

Aplicando el modelo COCOMO básico al ejemplo anterior y usando un tipo de proyecto orgánico obtenemos una estimación para el esfuerzo:

$$\begin{aligned}
 E &= a * KLOC^b \\
 E &= 2.4 * KLOC^{1.05} \\
 &= 2.4 * 12.6 * 1.05 \\
 &= 31.752 \\
 &= 31 \text{ hombre-mes}
 \end{aligned}$$

Para calcular la duración del proyecto usamos la estimación de esfuerzo:

$$\begin{aligned}
 D &= c * E^d \\
 D &= 2.5 * E^{0.38} \\
 &= 2.5 * 31^{0.38} \\
 &= 29.45 \\
 &= 29 \text{ meses}
 \end{aligned}$$

El valor de la duración del proyecto permite al planificador recomendar un número de personas N para el proyecto.

$$\begin{aligned}
 N &= E / D \\
 &= 31 / 29 \\
 &= 1 \text{ personas}
 \end{aligned}$$

Y si cada hombre-mes cuesta \$10,000 (entre sueldos y gastos extras), entonces el costo del sistema será de \$ 290,000

CAPITULO

3

Desarrollo, implantación y pruebas del sistema

3.1 Creación de la Base de Datos

3.2 Modelo Entidad Relación

Base de Datos

3.3 Programación de los módulos

3.4 Herramientas de desarrollo y contenido

3.5 Integración del sistema

Módulo de conexión

Módulo de validación de datos

Módulo de administración

Módulo de encuestas

Módulo de reportes

Módulo de procesos

3.6 Pruebas finales del programa piloto

Creación de la Base de Datos

La base de datos fue hecha basada en el modelo entidad relación del cual ya se habló en el capítulo 2

Modelo Entidad Relación

Modelos de datos relacionales

Los principios del modelo relacional fueron planteados por primera vez por el Dr. E. F. Codd en Junio de 1970. Una base de datos relacional usa relaciones o tablas de dos dimensiones para almacenar información.

Tabla: Dentro del enfoque relacional una tabla es conocida como una Relación: Una relación es una tabla de dos dimensiones con las siguientes propiedades:

- ☐ Cada columna contiene valores relativos al mismo atributo, y cada valor de una columna de la tabla debe ser simple (un solo valor).
- ☐ Cada columna tiene un nombre distinto (nombre del atributo), y el orden de las columnas no es importante
- ☐ Cada renglón es distinto, esto es, un renglón no puede duplicarse en otro para un grupo de columnas seleccionadas como llave
- ☐ Cada atributo no llave debe depender sólo de la llave de la relación no de ningún otro no llave

Tupla: Conjunto de valores que componen un renglón de la relación. Es equivalente a una instancia de un registro.

Grado de una tupla: Número de atributos que tiene una tupla (n de una n-tupla)

Cardinalidad: Número de tuplas de una relación.

Dominio: Conjunto de todos los valores posibles para un atributo.

Base de Datos

Para la creación de las tablas se echó mano de sentencias de SQL, dichas sentencias se clasifican según su finalidad y uso, en tres sublenguajes que son:

- ☐ El DDL (Data Description Language) o Lenguaje de Definición de Datos, el cual contiene órdenes para crear, modificar o eliminar estructuras de datos como tablas, bases de datos, índices; en las que se almacenan los datos.

Las instrucciones correspondientes son:

CREATE, ALTER, DROP, RENAME

- ☞ Por otra parte se encuentra en DCL (Data Control Language) o Lenguaje de Control de Datos, que incluye elementos para trabajar con el entorno multiusuario en el que cobra importancia la protección de los datos así como la seguridad de las tablas y el establecimiento de privilegios en el acceso. Siendo sus comandos:

GRANT, REVOKE

- ☞ El Lenguaje de Manipulación de Datos o DML (Data Manipulation Language), el cual permite insertar, modificar y eliminar datos de las tablas de la base de datos.

SELECT, INSERT, UPDATE, DELETE

Tenemos entonces que algunos de los queries usados para la creación de las tablas de la base de datos usada en el sistema y que aparecen en la figura, los cuales se capturaron y ejecutaron en el *SQL Query Analyze*; son los siguientes:

```
create table profesor
(
  id_prof varchar(15) not null primary key,
  nombre varchar(50) not null,
  ap_pat varchar(50),
  ap_mat varchar(50),
  id_division varchar(6) not null,
  grado varchar(10),
)

create table asignaturas
(
  clave char(4) not null primary key,
  nombre varchar(80) not null,
  id_division varchar(6) not null,
)

create table division
(
  id_division varchar(6) not null primary key,
  nom_division varchar(80) not null,
)
```



```

create table asigna_prof
(
  id_prof varchar(15) not null,
  clave char(4) not null,
  grupo char(2) not null,
  semestre char(6) not null,
  tipo varchar(15) not null,
  num_prof char(1) not null,
  constraint pk_AsProf primary key(clave,grupo,semestre),
  foreign key(id_prof) references profesor(id_prof),
  foreign key(clave) references asignaturas(clave),
)

```

La lista de tablas que se crearon es:

| <i>Nombre de la tabla</i> | <i>Contenido</i> |
|---------------------------|---|
| profesor | Datos del profesor |
| alumno | Datos del alumno |
| asignaturas | Datos de la asignatura |
| asigna_alumno | Tabla transitiva entre alumnos y asignaturas |
| asigna_prof | Tabla transitiva entre profesores y asignaturas |
| division | Datos de la división |
| deptos | Datos de los departamentos |
| permisos | Permisos de acceso al sistema |
| semestre | Semestre actual |
| actualiza | Resultados de los reactivos de la encuesta |
| encuesta_alumno | Datos de los alumnos que realizaron la encuesta |
| pxp | Reporte de asignatura - profesor por pregunta |
| actualiza_pxp | Actualización del reporte pxp |
| pxc | Reporte de asignatura - profesor por concepto |
| actualiza_pxc | Actualización del reporte pxc |
| ddc | Reporte de división - departamento por pregunta |
| actualiza_ddc | Actualización del reporte ddc |
| ddp | Reporte de división - departamento por concepto |
| actualiza_ddp | Actualización del reporte ddp |

| profesor | | | |
|-----------------|-------------|----------------|---------|
| id_prof | Varchar(15) | Llave primaria | No nulo |
| nombre | Varchar(50) | | No nulo |
| ap_pat | Varchar(50) | | Nulo |
| ap_mat | Varchar(50) | | Nulo |
| id_divison | Varchar(6) | | No nulo |
| grado | Varchar(15) | | Nulo |
| correo | Varchar(50) | | Nulo |

| alumno | | | |
|---------------|-------------|----------------|---------|
| num_cta | Varchar(9) | Llave primaria | No nulo |
| fecha_nac | Varchar(8) | | No nulo |
| pat_alu | Varchar(50) | | Nulo |
| mat_alu | Varchar(50) | | Nulo |
| nom_alu | Varchar(50) | | No nulo |

| asignatura | | | |
|-------------------|-------------|----------------|---------|
| clave | Char(4) | Llave primaria | No nulo |
| nom_asig | Varchar(80) | | No nulo |
| id_division | Varchar(6) | Llave foránea | No nulo |
| id_depto | Char(2) | Llave foránea | No nulo |

| asigna_alumno | | | |
|----------------------|------------|----------------|---------|
| num_cta | Varchar(9) | Llave foránea | No nulo |
| asig_alu | Char(4) | Llave foránea | No nulo |
| grupo_alu | Char(4) | Llave foránea | No nulo |
| sem_alu | Char(6) | Llave primaria | No nulo |

| asigna_prof | | | |
|--------------------|-------------|----------------|---------|
| id_prof | Varchar(15) | Llave foránea | No nulo |
| clave | Char(4) | Llave foránea | No nulo |
| grupo | Char(4) | Llave primaria | No nulo |
| semestre | Varchar(50) | Llave primaria | No nulo |
| tipo | Varchar(6) | | No nulo |
| num_prof | Char(1) | | No nulo |

| division | | | |
|--------------|-------------|----------------|---------|
| id_division | Varchar(6) | Llave primaria | No nulo |
| nom_division | Varchar(80) | | No nulo |

| deptos | | | |
|--------------|-------------|----------------|---------|
| id_division | Varchar(6) | Llave primaria | No nulo |
| nom_division | Varchar(80) | | No nulo |

| permisos | | | |
|------------|-------------|----------------|---------|
| usuario | Varchar(20) | Llave primaria | No nulo |
| pass | Varchar(20) | | No nulo |
| privilegio | Varchar(20) | | No nulo |

| semestre | | | |
|-----------------|---------|--|---------|
| Semestre_actual | char(6) | | No nulo |

| actualiza | | | |
|-----------|---------|---------------|---------|
| clave | Char(4) | Llave foránea | No nulo |
| grupo | Char(4) | Llave foránea | No nulo |
| semestre | Char(6) | Llave foránea | No nulo |
| q1#1 | Int | | Nulo |
| q1#2 | Int | | Nulo |
| q1#3 | Int | | Nulo |
| q1#4 | Int | | Nulo |
| q2#1 | Int | | Nulo |
| ... | | | |
| q29#4 | Int | | Nulo |

| encuesta_alumno | | | |
|-----------------|------------|---------------|---------|
| enc_num_cta | Varchar(9) | Llave foránea | No nulo |
| enc_asig_alu | Char(4) | Llave foránea | No nulo |
| enc_sem_alu | Char(6) | Llave foránea | No nulo |
| enc_grupo_alu | Char(4) | | No nulo |
| enc_realizo | Char(1) | | No nulo |
| enc_comen | text | | Nulo |

| pxp | | | |
|------------|---------|---------------|---------|
| clave | Char(4) | Llave foránea | No nulo |
| grupo | Char(4) | Llave foránea | No nulo |
| semestre | Char(6) | Llave foránea | No nulo |
| ni_asig | Int | | Nulo |
| ne_asig | Int | | Nulo |
| ni_grupo | Int | | Nulo |
| ni_grupo | Int | | Nulo |
| q1_pxp | Int | | Nulo |
| q2_pxp | Int | | Nulo |
| q3_pxp | Int | | Nulo |
| q4_pxp | Int | | Nulo |
| q5_pxp | Int | | Nulo |
| ... | | | |
| q29_pxp | Int | | Nulo |

| actualiza_pxp | | | |
|----------------------|---------|---------------|---------|
| clave | Char(4) | Llave foránea | No nulo |
| grupo | Char(4) | Llave foránea | No nulo |
| semestre | Char(6) | Llave foránea | No nulo |
| q1_md | Int | | Nulo |
| q2_md | Int | | Nulo |
| q3_md | Int | | Nulo |
| q4_md | Int | | Nulo |
| q5_md | Int | | Nulo |
| ... | | | |
| q29_md | Int | | Nulo |

| pxc | | | |
|------------|---------|---------------|---------|
| clave | Char(4) | Llave foránea | No nulo |
| grupo | Char(4) | Llave foránea | No nulo |
| semestre | Char(6) | Llave foránea | No nulo |
| ni_asig | Int | | Nulo |
| ne_asig | Int | | Nulo |
| ni_grupo | Int | | Nulo |
| ni_grupo | Int | | Nulo |

| | | | |
|-----------|-----|--|------|
| con1_pxc | Int | | Nulo |
| con2_pxc | Int | | Nulo |
| con 3_pxc | Int | | Nulo |
| con 4_pxc | Int | | Nulo |
| con 5_pxc | Int | | Nulo |
| con 6_pxc | Int | | Nulo |

actualiza_pxc

| | | | |
|----------|---------|---------------|---------|
| clave | Char(4) | Llave foránea | No nulo |
| grupo | Char(4) | Llave foránea | No nulo |
| semestre | Char(6) | Llave foránea | No nulo |
| con1_md | Int | | Nulo |
| con2_md | Int | | Nulo |
| con3_md | Int | | Nulo |
| con4_md | Int | | Nulo |
| con5_md | Int | | Nulo |
| con6_md | Int | | Nulo |

ddp

| | | | |
|-----------------|------------|---------------|---------|
| id_division | Varchar(6) | Llave foránea | No nulo |
| id_departamento | Char(2) | Llave foránea | No nulo |
| semestre | Varchar(6) | | No nulo |
| ni_div | Int | | Nulo |
| ne_div | Int | | Nulo |
| ni_dep | Int | | Nulo |
| ni_dep | Int | | Nulo |
| q1_ddp | Int | | Nulo |
| q2_ddp | Int | | Nulo |
| q3_ddp | Int | | Nulo |
| q4_ddp | Int | | Nulo |
| q5_ddp | Int | | Nulo |
| q6_ddp | Int | | Nulo |
| ... | | | |
| q29_ddp | Int | | Nulo |

actualiza_ddp

| | | | |
|-------------|------------|---------------|---------|
| id_division | Varchar(6) | Llave foránea | No nulo |
|-------------|------------|---------------|---------|

| | | | |
|---------|-----|--|------|
| q1_ddp | Int | | Nulo |
| q2_ddp | Int | | Nulo |
| q3_ddp | Int | | Nulo |
| q4_ddp | Int | | Nulo |
| q5_ddp | Int | | Nulo |
| q6_ddp | Int | | Nulo |
| ... | | | |
| q29_ddp | Int | | Nulo |

| ddc | | | |
|-----------------|------------|---------------|---------|
| id_division | Varchar(6) | Llave foránea | No nulo |
| id_departamento | Char(2) | Llave foránea | No nulo |
| semestre | Varchar(6) | | No nulo |
| ni_div | Int | | Nulo |
| ne_div | Int | | Nulo |
| ni_dep | Int | | Nulo |
| ni_dep | Int | | Nulo |
| con1_ddc | Int | | Nulo |
| con2_ddc | Int | | Nulo |
| con 3_ddc | Int | | Nulo |
| con 4_ddc | Int | | Nulo |
| con 5_ddc | Int | | Nulo |
| con 6_ddc | Int | | Nulo |

| actualiza_ddc | | | |
|----------------------|------------|---------------|---------|
| id_division | Varchar(6) | Llave foránea | No nulo |
| con1_ddc | Int | | Nulo |
| con2_ddc | Int | | Nulo |
| con 3_ddc | Int | | Nulo |
| con 4_ddc | Int | | Nulo |
| con 5_ddc | Int | | Nulo |
| con 6_ddc | Int | | Nulo |

Programación de los módulos

Siempre que se quiera iniciar la programación de los módulos del sistema que se desea construir, se debe definir los recursos tanto en hardware como en software así como las herramientas con las que se dispone, y que en nuestro caso son las siguientes:

Recursos del Cliente

Navegador (Browser)

Recursos del Servidor

Hardware

Servidor Compaq Proliant

Procesador Intel Xeon

Software

Servidor Web: Internet Information Server (IIS) 4.

Integrado en el sistema operativo Windows 2000 Server

Microsoft BackOffice:

Windows 2000 Server

Microsoft SQL Server (Base de Datos)

Interprete de PHP versión 4

Herramientas de desarrollo y contenido

PHP 4

HTML

JavaScript

Macromedia Fireworks MX

Macromedia Dreamweaver MX

Integración del sistema

Para el desarrollo del sistema se definió la separación de los módulos en los siguientes seis conjuntos:

- 📄 Módulo de conexión
- 📄 Módulo de validación de datos
- 📄 Módulo de administración
- 📄 Módulo de encuestas
- 📄 Módulo de reportes
- 📄 Módulo de procesos

Módulo de conexión

Este módulo tiene como principal finalidad la conexión con la base de datos. Se inició con pruebas de conectividad a la base de datos mediante el lenguaje de PHP y su conexión a SQL Server logrando establecerla mediante el siguiente código:

```
<?php
function Conectarse()
{
    if (!($link=mssql_connect("morfeo","sa","underworld")))
    {
        echo "Error conectando a la base de datos.";
        exit();
    }
    if (!mssql_select_db("encuesta",$link))
    {
        echo "Error seleccionando la base de datos.";
        exit();
    }
    return $link;
}
$link=Conectarse();
echo " Se logró la Conexión con la base de datos .<br>";
mssql_close($link);
?>
```

Módulo de validación de datos

Para que los datos ingresados mantengan una uniformidad en cuanto al formato y los tipos, antes de guardarlos en la base de datos, se requiere que haya un mecanismo que se encuentre entre la etapa de captura de

datos y la de guardar la información. Para cumplir con esto se crearon unas funciones en PHP que manejan expresiones regulares para realizar dichas validaciones de datos que fueron las siguientes:

- * *Validación de fechas:* Esta función usa la una expresión regular, la cual verifica que la fecha sea del formato dd/mm/aaaa y que solo contenga dígitos, además de que verifica que los días correspondan al mes, es decir, si el mes de junio solo cuenta con 30 días, no se pueda ingresar la fecha 31/06/2002, como una fecha válida. También cuenta con la llamada a otra función que checa los días de febrero para el caso de años bisiestos.

```

<?
function bisiestro($anio)
{
    return (($anio % 4 == 0 && $anio % 100 != 0) ||
        $anio % 400 == 0);
}

function es_fecha($cadena)
{
    echo " $cadena<br>";
    $expresion_regular= "^([[:digit:]]{2})/([[:digit:]]{2})/([[:digit:]]{4})$";

    if (ereg($expresion_regular, $cadena, $coincidencias))
    {

        if($coincidencias[2]<1 || $coincidencias[2]>12)
            return "No es un mes válido";

        if(($coincidencias[1]<1 || $coincidencias[1]>31) ||
            ($coincidencias[2] = 4 && $coincidencias[1]>30) ||
            ($coincidencias[2] = 6 && $coincidencias[1]>30) ||
            ($coincidencias[2] = 9 && $coincidencias[1]>30) ||
            ($coincidencias[2] = 11 && $coincidencias[1]>30) ||
            ($coincidencias[2] = 2 && bisiestro ($coincidencias[3])
            && $coincidencias[1] >29) ||
            ($coincidencias[2] = 2 && !bisiestro ($coincidencias[3])
            && $coincidencias[1] >28))
            return "Fecha NO válido";
        return "Fecha válido";
    }
    else
    return "no es fecha";
}

```

?>

- Validación de nombres: La cual permite que solo se ingresen nombres, sin números, ni caracteres no permitidos.

```

$arreglo=split(" ", $cadena);
for($i =0; $i<count($arreglo); $i++)
{
    $cadena2=$arreglo[$i];
    echo "$arreglo[$i]<br>";
    $tam=count($arreglo);
    echo "$tam<br>";
    $cadenax= $arreglo[$i];

    $res=es_nombre($cadenax);
    echo "<font color=#CC0000><b>$res</b> </font><br>";
}

function es_nombre($cadena2)
{

    $expresion_regular= "^[a-zñáéíóúüA-ZÑÁÉÍÓÚ]+$";

    if (ereg($expresion_regular, $cadena2, $coincidencias))
    {
        return "Nombre válido";
    }
    else
        return "Nombre con caracteres no válidos";

}

```

- Validación de semestre: Esta función verifica que se ingrese un semestre válido de caracteres numéricos.

```

function es_semestre($cadena)
{
    echo " $cadena<br>";
    $expresion_regular= "^[[:digit:]]{4}/([[:digit:]]{1})$";

    if (ereg($expresion_regular, $cadena, $coincidencias))
    {

        if($coincidencias[2]>2)
            return "no es valido el subindice $coincidencias[2]";

        if($coincidencias[1]>2000 && $coincidencias[1]<2032)
            return "Semestre valido";
        else

```

```

        return "Semestre NO valido $coincidencias[1]";
    }
    else
    return "no es semestre";
}

```

- Formateo de datos:** Esta función auxiliar permite que se la cadena ingresada no contenga caracteres no permitidos; borra los espacios vacíos que existan al inicio y al final y convierte todo a una cadena de palabras minúsculas con iniciales mayúsculas







```

function formateo($cadena)
{
    $cadena=" Jorge vICENTE CLeMe&%nt tOffolÓN";
    $cadena2=chop($cadena);
    $cadena2=strtolower($cadena);
    $cadena2=ucwords($cadena2);
    return $cadena2;
}
?>

```

Módulo de administración

En cuanto al módulo de administración el cual solo se puede ingresar mediante la contraseña de administrador que en este caso se piensa que será la secretaria del departamento o división; se determinó que debería llevar secciones que permitan hacer las siguientes tareas administrativas:

-  Menú de ingreso al módulo
-  Alta de profesor.
-  Alta de asignaturas que imparte el profesor.
-  Alta de alumnos.
-  Alta de asignaturas para alumnos.
-  Pantalla de ingreso para ver reportes.

Inicialmente tenemos en el módulo de administración una sección en la cual se despliega el menú de ingreso a dicho módulo en el cual se encuentran las opciones de Alta de profesor, Alta de asignaturas que imparte el profesor, Alta de alumnos, Alta de asignaturas para alumnos, Pantalla de ingreso para ver reportes.

En *alta de profesor* tenemos una pantalla que nos permite capturar los datos del profesor como son: clave del profesor, nombre, apellido paterno,

apellido materno, grado (Ing., Dr., M. en C. M. en I. y Lic.) y división (División Educación Continua, División de Ciencias Sociales y Humanidades, División de Estudios de Posgrado, División de Ingeniería Mecánica e Industrial, División de Ciencias Básicas, División de Ingeniería Civil Topográfica y Geodésica, División de Ingeniería Eléctrica, División de Ingeniería en Ciencias de la Tierra)

En la sección de *Alta de asignaturas que imparte el profesor* se ligan las asignaturas que imparte cada profesor en el semestre en curso y se capturan la clave del profesor, clave de la asignatura, grupo, semestre, tipo de clase (teoría, laboratorio y práctica) y número de profesor (1, 2, 3, 4 y 5).

En *alta de alumnos* se capturan el nombre, apellido materno y apellido paterno del alumno así como su número de cuenta y fecha de nacimiento, estos dos últimos datos son importantes ya que serán usados como claves de ingreso y así se permitirá ligar todos los datos del alumno al momento de contestar su encuesta.

Una vez dados de alta los alumnos con sus correspondientes datos y las asignaturas con sus claves. En la sección correspondiente al *alta de asignaturas para alumnos* se capturan el número de cuenta, grupo, clave de la asignatura y semestre.

Finalmente en la sección de *pantalla de ingreso para ver reportes* que como su nombre lo indica permite el ingreso al módulo de generación de reportes y para esto se requieren la clave de la asignatura, el grupo y semestre. Ya con estos datos se pueden hacer las ligas automáticamente a todos los demás datos como profesor, asignatura, división, departamento, etc.

Módulo de encuestas

El módulo de encuestas es el que permite desplegar los 29 reactivos y sus opciones con los que cuenta la encuesta del programa de evaluación del proceso enseñanza- aprendizaje y a su vez capturar las respuestas de los alumnos además de los datos generales de éste y sus sugerencias y comentarios que tengan respecto a la impartición de la asignatura por parte del profesor. Por ello se determinó tener dos secciones:

Pantalla de selección de materia a realizar la encuesta.

Pantalla de captura de encuesta.

La pantalla de captura de encuesta cuenta con elementos conocidos en HTML como `<input type="radio" name="radiobutton">` que permiten sustituir a los alvéolos de la encuesta tradicional de medio de lectura óptica.

Además de otro elemento llamado `<textarea>` que permite guardar los comentarios y sugerencias.

Una de las mejoras que tiene el sistema es que al momento de contestar la encuesta, como el alumno ya fue dado de alta por el sistema administrativo, ya no es necesario que llene datos como clave de la Nombre del profesor, clave de la asignatura, semestre, tipo de clase, y número del profesor. Ya que solo tiene que elegir en un listado dinámico las asignaturas en las que está actualmente inscrito y que no ha contestado aún.

Módulo de reportes

Una vez capturada la información mediante el ejercicio de las encuestas, se crearon la parte más significativa de este sistema que es el procesamiento de la información mediante un tratamiento estadístico y su clasificación en reportes para agrupar los resultados mostrándolos de manera óptima, legible y sintetizada con la finalidad de presentar las situaciones de interés de la población estudiada a diversos niveles que pueden ser académico, por departamento, por división o bien incluso a nivel facultad. Lo cual ayudará directamente en la toma de decisiones de manera más certera y expedita.

Los reportes que se enumeran a continuación fueron clasificados principalmente en cinco categorías con base en su análisis, más un reporte general que es el entregado a los académicos:

- Reporte de profesor por pregunta – respuesta
- Reporte de asignatura - profesor por pregunta
- Reporte de asignatura - profesor por concepto
- Reporte de división - departamento por pregunta
- Reporte de división - departamento por concepto
- Reporte general del profesor

En todos los reportes se muestra información general como lo es el semestre, la fecha, la división, el departamento, la asignatura, la clave y el grupo así como el nombre del profesor al cual pertenece el reporte generado a partir de las encuestas elaboradas. Adicionalmente se incluye información como el número de alumnos inscritos en la asignatura, la división o en el departamento según sea el caso y se le denominó como (NI). También el número de encuestas realizadas en la asignatura, la división o en el departamento según sea el caso y que se le dio el pnmónico de (NE) y finalmente el porcentaje de encuestados que en este caso sería el redondeo de $(NE)/(NI) \times 100$.

Para realizar lo anterior se empleo el código siguiente:

```
//profesor
$query="select profesor.nombre nombre,profesor.ap_pat pat,profesor.ap_mat mat
from profesor,asigna_prof
where asigna_prof.id_prof=profesor.id_prof
and asigna_prof.clave = '$clave' and asigna_prof.grupo='$grupo'
and asigna_prof.semestre = '$sem'";
$result=mssql_query($query,$link);
while($row = mssql_fetch_array($result))
{
    $nom_prof = $row["nombre"]." ".$row["pat"]." ".$row["mat"];
}

//division y departamento
$query="select division.nom_division nombrediv, deptos.desc_depto nombredep
from division,asignaturas,deptos
where division.id_division = asignaturas.id_division
and deptos.id_depto = asignaturas.id_depto
and asignaturas.clave = '$clave'";
$result=mssql_query($query,$link);
while($row = mssql_fetch_array($result))
{
    $div = $row["nombrediv"];
    $dep = $row["nombredep"];
}

//fecha
$fecha = date("d/m/Y");

//asignatura
$query="select nom_asig from asignaturas where clave = '$clave'";
$result=mssql_query($query,$link);
while($row = mssql_fetch_array($result))
```

```

    {
        $asig = $row["nom_asig"];
    }

//ni
$query="select count(*) conteo from asigna_alumno
where asig_alu = '$clave' and grupo_alu = '$grupo'
and sem_alu = '$sem'";
$result=mssql_query($query,$link);
while($row = mssql_fetch_array($result))
{
    $ni = $row["conteo"];
}

//ne
$query="select q1#1,q1#2,q1#3,q1#4 from actualiza
where clave = '$clave' and grupo = '$grupo'
and semestre = '$sem'";
$result=mssql_query($query,$link);
while($row = mssql_fetch_array($result))
{
    $ne = $row["q1#1"]+$row["q1#2"]+$row["q1#3"]+$row["q1#4"];
}
//e
$e = ($ne / $ni)*100;

```

Reporte de profesor por pregunta respuesta

En el reporte de profesor por pregunta respuesta se presentan veinte tablas que cada una corresponde a los primeros veinte reactivos que son los que tratan temas en torno al profesor. En cada tabla se resume el número de encuestados que contestaron cada opción y el total que representan según los pesos específicos de cada opción que son 3, 2, 1, 0 para las opciones 1, 2, 3 y 4 respectivamente. Adicionalmente en la tabla se presenta el *Máximo* que es el número de encuestados por el peso máximo que en este caso es 3. *Suma* es la sumatoria de los totales de cada opción y finalmente se tiene *Total de preguntas* que se calcula como *suma* entre el *máximo* por cien. Todo lo anterior es codificado como:

```

$query="select q1#1,q1#2,q1#3,q1#4 from actualiza
where clave = '$clave' and grupo = '$grupo'
and semestre = '$sem'";
$result=mssql_query($query,$link);
while($row = mssql_fetch_array($result))
{
    $ne_p1_1 = $row["q1#1"];

```



```

$total_p1_1 = $ne_p1_1 * $peso1;
$ne_p1_2 = $row["q1#2"];
$total_p1_2 = $ne_p1_2 * $peso2;
$ne_p1_3 = $row["q1#3"];
$total_p1_3 = $ne_p1_3 * $peso3;
$ne_p1_4 = $row["q1#4"];
$total_p1_4 = $ne_p1_4 * $peso4;
$max_p1 = $ne * 3;
$suma_p1 = $total_p1_1 + $total_p1_2 + $total_p1_3 + $total_p1_4;
$totalI = ($suma_p1*100)/$max_p1;
}

```

Reporte de asignatura - profesor por pregunta

En este reporte se presenta información mediante dos tablas que contienen el grupo, número de inscritos al grupo, número de encuestados en el grupo, porcentaje de encuestado en el grupo, así como el número de respuestas para cada reactivo de la encuesta.

En las tablas se presenta información tanto de los profesores de la asignatura como de los alumnos del profesor.

Reporte de asignatura - profesor por concepto

Al igual que en los otros reportes se presenta información general propia del tipo de reporte del que se trate, adicionalmente se puede visualizar los profesores que imparten una asignatura en particular, sus grupos, el número de inscritos a cada grupo, número de encuestados en cada grupo, porcentaje de encuestado en el grupo. Se agruparon las respuestas de los reactivos en cuatro rubros que fueron:

Rubro I - Aprendizaje.- Correspondiente a los reactivos (5,7,8,9,10).

Rubro II - Motivación.- Correspondiente a los reactivos (1,2,3,4,6).

Rubro III - Evaluación.- Correspondiente a los reactivos (12,13,15).

Rubro IV - Responsabilidad.- Correspondiente a los reactivos (11,14,16).

Rubro V – Desempeño del alumno.- Correspondiente a los reactivos (24,25,26,27,28).

Rubro VI – Responsabilidad del alumno .- Correspondiente a los reactivos (21,22).

Rubro VII – Sin evaluación del profesor (17,18,19,20).

De los cuales se presentó la media de todos cada rubro por grupo y finalmente el promedio de estos. Así como la la media de la asignatura y del departamento para cada rublo.

Reporte de división - departamento por pregunta

En este reporte se contienen dos tipos de tablas; una correspondiente a los reactivos de los profesores del departamento (reactivos del 1 al 20) y la otra a los reactivos de los alumnos del departamento (reactivos del 21 al 29) En los cuales se sintetizan los resultados de cada respuesta a los reactivos para cada departamento de cada división en específico. Anotando la calificación total que recibe cada una y que se calcula como el número de redondeo truncado de

$$\sum_{i=1}^n \text{numres}_n$$

donde:

n = número total de reactivos (19)

numres = número de respuesta.

El número total de reactivos en este caso es de 19 ya que el reactivo número 20 se excluye por tener una respuesta discreta (si/no) y se presentan sus totales por separado.

Un tratamiento similar se aplica en la segunda tabla pero para el caso de los reactivos 21 al 29.

Reporte de división - departamento por concepto

En este tipo de reporte al igual que en los demás se separa la información en lo que se refiere a los diferentes rublos previamente definidos tanto para el profesor del departamento como para los alumnos del departamento. Este reporte es muy similar al de asignatura - profesor por concepto, a diferencia de que en éste el análisis es por departamento y no por asignatura.

Reporte general del profesor

Finalmente en este reporte se presenta información condensada que engloba todos los demás reportes.

Se presenta información de cada una de las 29 preguntas de la encuesta como lo es calificación del grupo, máximos, mínimos y la media obtenida en los grupos de la asignatura, la media del departamento, división y de la facultad.

Módulo de procesos

Por las características del sistema es necesario volver a ejecutar todos los procesos estadísticos que se llevaron a cabo en los reportes, ya que estos solo se ejecutan de manera parcial y para poder obtener datos globales como estadísticas por grupo, asignatura, departamento o división se crearon cuatro programas que vuelven a correr los procesos una vez terminada la etapa de levantamiento de encuestas, los cuales son:

Programa para procesos estadísticos por grupo.

Programa para procesos estadísticos por asignatura.

Programa para procesos estadísticos por departamento.

Programa para procesos estadísticos por división.

Pruebas finales del programa piloto

Las pruebas se realizaron en dos etapas, la primera en el semestre 2003-2 con un grupo, y la segunda en el semestre 2004-1 con cinco grupos, algunos de los resultados fueron los siguientes:

Primera Etapa:

Clave: 0134

Grupo: 03

Semestre: 2003 – 2

No. Inscritos: 40

No. Encuestas: 32

% Encuestas: 80 %



Análisis de Profesor por Pregunta - Respuesta

Profesor: Jorge Valenano Assam
División: División de Ingeniería Eléctrica
Departamento: Departamento de Ingeniería en Computación
Asignatura: Diseño de Sistemas Digitales
Clave: 0134
Grupo: 03

Semestre: 2004-1
Fecha: 16/02/2004
No. Inscritos (NI): 35
No. Encuestas (NE): 12
% Encuestados (%E): 34.29

Para consultar la relación de preguntas de la encuesta: [encuestas](#)

| Pregunta 1 | | | Pregunta 2 | | | Pregunta 3 | | | Pregunta 4 | | |
|------------|-----|------------|------------|-----|------------|------------|-----|------------|------------|-----|------------|
| Opción | NE | Peso Total | Opción | NE | Peso Total | Opción | NE | Peso Total | Opción | NE | Peso Total |
| 1 | 9 * | 9 = 27 | 1 | 5 * | 5 = 15 | 1 | 4 * | 4 = 12 | 1 | 7 * | 7 = 21 |
| 2 | 3 * | 2 = 6 | 2 | 7 * | 2 = 14 | 2 | 8 * | 2 = 16 | 2 | 5 * | 2 = 10 |
| 3 | 0 * | 1 = 0 | 3 | 0 * | 1 = 0 | 3 | 0 * | 1 = 0 | 3 | 0 * | 1 = 0 |

3.2. Ejemplo de reporte de profesor por pregunta

En esta etapa se probó la infraestructura del sistema respecto a la capacidad de presentación y cómputo de las encuestas, y las pruebas resultaron satisfactorias. Se recabaron observaciones de los alumnos, en cuanto a la experiencia de usuario y el desempeño del sistema.

Segunda Etapa:

Clave: 0134

Grupo: 03

Semestre: 2004-1

No. Inscritos 35

No. Encuestas: 12

% encuestados: 34.29

RESUMEN
Conceptos y Preguntas

| CONCEPTO | PUNTOS | | CALIFICACION CONCEPTO |
|---------------------------------|--------|--------|--------------------------|
| | GRUPO | MAXIMO | |
| I.- APRENDIZAJE | 33.00 | 36 | 91.67 |
| II.- MOTIVACION | 29.80 | 36 | 82.78 |
| III.- EVALUACION | 23.67 | 36 | 65.74 |
| IV.- RESPONSABILIDAD | 34.67 | 36 | 96.30 |
| V.- DESEMPEÑO DEL ALUMNO | 24.00 | 36 | 66.67 |
| VI.- RESPONSABILIDAD DEL ALUMNO | 32.25 | 36 | 89.58 |

Rubro I - APRENDIZAJE - Correspondiente a los reactivos (5,7,8,9,10).

Rubro II - MOTIVACION - Correspondiente a los reactivos (1,2,3,4,6).

Rubro III - EVALUACION - Correspondiente a los reactivos (12,13,15).

Rubro IV - RESPONSABILIDAD - Correspondiente a los reactivos (11,14,16).

Rubro V - DESEMPEÑO DEL ALUMNO - Correspondiente a los reactivos (24,25,26,27,28).

Rubro VI - RESPONSABILIDAD DEL ALUMNO - Correspondiente a los reactivos (21,22).

Rubro VII - SIN EVALUACIÓN DEL PROFESOR (17,18,19,20).

CALIFICACION DEL PROFESOR: 84.12

CALIFICACION DEL ALUMNO: 78.13

3.3. Resultados del grupo

Clave: 0434

Grupo: 03

Semestre: 2004-1

No. Inscritos 35

No. Encuestas: 4

% encuestados: 11.43

RESUMEN
Conceptos y Preguntas

| CONCEPTO | PUNTOS | | CALIFICACION CONCEPTO |
|---------------------------------|--------|--------|--------------------------|
| | GRUPO | MAXIMO | |
| I.- APRENDIZAJE | 11.40 | 12 | 95.00 |
| II.- MOTIVACION | 11.00 | 12 | 91.67 |
| III.- EVALUACION | 9.67 | 12 | 80.56 |
| IV.- RESPONSABILIDAD | 11.67 | 12 | 97.22 |
| V.- DESEMPEÑO DEL ALUMNO | 9.20 | 12 | 76.67 |
| VI.- RESPONSABILIDAD DEL ALUMNO | 11.50 | 12 | 95.83 |

Rubro I - APRENDIZAJE - Correspondiente a los reactivos (5,7,8,9,10).
 Rubro II - MOTIVACION - Correspondiente a los reactivos (1,2,3,4,6).
 Rubro III - EVALUACION - Correspondiente a los reactivos (12,13,15).
 Rubro IV - RESPONSABILIDAD - Correspondiente a los reactivos (11,14,16).
 Rubro V - DESEMPEÑO DEL ALUMNO - Correspondiente a los reactivos (24,25,26,27,28).
 Rubro VI - RESPONSABILIDAD DEL ALUMNO - Correspondiente a los reactivos (21,22).
 Rubro VII - SIN EVALUACIÓN DEL PROFESOR (17,18,19,20).

CALIFICACION DEL PROFESOR: 91.11

CALIFICACION DEL ALUMNO: 86.25

3.4. Resultados del grupo.

Clave: 0840

Grupo: 01

Semestre: 2004-1

No. Inscritos 40

No. Encuestas: 19

% encuestados: 47.50

RESUMEN
Conceptos y Preguntas

| CONCEPTO | PUNTOS | | CALIFICACION CONCEPTO |
|---------------------------------|--------|--------|--------------------------|
| | GRUPO | MAXIMO | |
| I - APRENDIZAJE | 49.60 | 57 | 87.02 |
| II - MOTIVACION | 48.60 | 57 | 85.26 |
| III - EVALUACION | 53.67 | 57 | 94.15 |
| IV - RESPONSABILIDAD | 55.67 | 57 | 97.66 |
| V - DESEMPEÑO DEL ALUMNO | 41.60 | 57 | 72.98 |
| VI - RESPONSABILIDAD DEL ALUMNO | 51.50 | 57 | 90.35 |

Rubro I - APRENDIZAJE - Correspondiente a los reactivos (5,7,8,9,10).
 Rubro II - MOTIVACION - Correspondiente a los reactivos (1,2,3,4,6).
 Rubro III - EVALUACION - Correspondiente a los reactivos (12,13,15).
 Rubro IV - RESPONSABILIDAD - Correspondiente a los reactivos (11,14,16).
 Rubro V - DESEMPEÑO DEL ALUMNO - Correspondiente a los reactivos (24,25,26,27,28).
 Rubro VI - RESPONSABILIDAD DEL ALUMNO - Correspondiente a los reactivos (21,22).
 Rubro VII - SIN EVALUACIÓN DEL PROFESOR (17,18,19,20).

CALIFICACION DEL PROFESOR: 91.02
CALIFICACION DEL ALUMNO: 81.66

Comentarios de los alumnos

En general la clase es buena

la clase esta muy bien, sólo a veces la maestra se desvia un poco del tema, tiempo que podria utilizarse mejor de otra manera es muy buena clase

Quisiera comenzar diciendole que como persona aunque no la he tratado mucho, se nota a leguas que es muy agradable y esto en lo particular se me hace genial porque así pienso yo que nos agrada asistir a clases, ya que hay otros profesores que son prepotentes y sangrónes y aparte de eso no son tan picudos, pienso que su clase es muy buena y ademas entretenida, definitivamente volveria a cursar con Ud. ATTE: Nazareth Sanchez Martínez

Creo que esta idea es excelente, sugiero tener un ambiente más amigable y sobre todo una sección de ayuda. También, esta última pregunta la debería de responder el sistema. Sobre todo la sección de validación se active con javascript al pulsar el boton de aceptar... y que indique cuales son las preguntas que no se respondieron (poniendo una viñeta junto a ellas por ejemplo). También creo que es prudente poner los datos del maestro (por lo menos el nombre) porque luego no saben ni a quien califican. Finalmente creo que esto es excelente, aunque dudo que si no hay sanción los alumnos llenen las encuestas.

Me parece bien que motive el uso de Linux, pero pienso que nos debería orientar un poco más acerca de su uso, como por ejemplo que distribución (de tantas que hay) nos conviene para empezar y a lo mejor hasta prestarnos un CD con la copia de la distribución que usted utiliza, de tal forma que si tenemos algún problema con el sistema operativo, usted nos pueda orientar. Porque el solo decir trabajen en Linux, pienso que distrae un poco la atención porque en los libros no viene descrito a detalle el uso del sistema operativo.

3.5. Pantalla de comentarios

Programa de Evaluación del Proceso Enseñanza - Aprendizaje
de la Facultad de Ingeniería

Análisis de Asignatura - Profesor por Pregunta

| | |
|---|---------------------------|
| División: División de Ingeniería Eléctrica | Semestre: 2003-2 |
| Departamento: Departamento de Ingeniería en Computación | Fecha: 18/02/2004 |
| Asignatura: Diseño de Sistemas Digitales | No. Insritos (NI): 42 |
| Clave: 0134 | No. Encuestas (NE): 34 |
| | % Encuestados (%E): 80.95 |

Para consultar la relación de preguntas de la encuesta [clicquea](#)

Profesores de la asignatura:

| CLAVE | PROFESOR | GRUPO | NI | NE | NÚMERO DE RESPUESTA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|----------------------|-------|----|----|---------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|
| | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | 7% | 7% | 7% | 6% | 8% | 5% | 8% | 8% | 7% | 7% | 5% | 7% | 4% | 3% | 7% | 4% | - | - | | | | | | | | | | | | | | | | | | | | | | |
| 1034 | Jorge Vale and Assem | 03 | 40 | 32 | 80 | 82 | 29 | 79 | 17 | 79 | 17 | 81 | 25 | 86 | 46 | 79 | 17 | 87 | 50 | 90 | 63 | 86 | 46 | 85 | 42 | 81 | 25 | 62 | 50 | 79 | 17 | 90 | 63 | 69 | 79 | 61 | 46 | 86 | 46 | 86 | 46 | 79 | | |
| 5544 | Christian Lopez | 03 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

3.6. Análisis de Asignatura profesor por pregunta



Universidad Nacional Autónoma de México
Facultad de Ingeniería



Programa de Evaluación del Proceso Enseñanza - Aprendizaje
de la Facultad de Ingeniería

Análisis de Asignatura - Profesor por Concepto

| | |
|---|---------------------------|
| División: División de Ingeniería Eléctrica | Semestre: 2003-2 |
| Departamento: Departamento de Ingeniería en Computación | Fecha: 18/02/2004 |
| Asignatura: Diseño de Sistemas Digitales | No. Insritos (NI): 42 |
| Clave: 0134 | No. Encuestas (NE): 34 |
| | % Encuestados (%E): 80.95 |

Para consultar la relación de preguntas de la encuesta: [encuesta](#)

Profesores de la asignatura:

| PROFESOR | GRUPO | NI | NE | %E | APRENDIZAJE* | MOTIVACION* | EVALUACION* | RESPONSABILIDAD* | CALIF FINAL |
|--------------------------|-------|----|----|----|--------------|-------------|-------------|------------------|-------------|
| Jorge Valeriano Assem | 03 | 40 | 32 | 80 | 87.28 | 80.21 | 70.49 | 77.76 | 76.9475 |
| Christian Luna Rodríguez | 57 | 0 | 0 | 0 | 0.00 | 0.00 | 0.00 | 0.00 | 0 |

3.7. Análisis de Asignatura profesor por concepto

En esta segunda etapa, las pruebas se concentraron en la capacidad del sistema en cuanto al cómputo de los resultados, así como de los procesos estadísticos del mismo, asimismo se obtuvieron más impresiones de los usuarios (los alumnos), las cuales en general fueron favorables.

Por lo que las pruebas se desarrollaron sin mayores problemas, en cuanto a la infraestructura del sistema, se probó asimismo la generación de los reportes generales.

Cabe destacar que los resultados obtenidos para los reportes por asignatura y división no son muy significativos, debido a la cantidad reducida de grupos encuestados, pero nos permitieron evaluar el comportamiento tanto de los reportes, como del proceso que se lanza para generar las estadísticas. Estas pruebas también fueron satisfactorias.

CAPÍTULO

4

Manual de usuario

Manual de usuario

Manual de instalación

Requerimientos de Hardware y Software

Funcionamiento del sistema

Convenciones generales del sistema

Descripción del ambiente de trabajo

Procesos principales del sistema

Opciones del administrador

Altas

Procesos estadísticos

Ver Reporte tipo

Encuesta del alumno

Reportes de profesor, jefe de departamento y división.

4.1 Manual de usuario

Manual de Instalación

Requerimientos de Hardware y Software

Hardware:

Procesador a 90 MHz o superior.

16 MB en RAM o superior.

Monitor VGA o superior, resolución mínima recomendada de 800 x 600.

Conexión a Internet a 28.8 Kbps o superior.

Software:

Microsoft Windows 95, 98, ME, 2000, ó XP.

Internet Explorer 5.x o superior.

Instalación del sistema.

No requiere de ninguna instalación especial, por tratarse de una aplicación Web.




4.3 Funcionamiento del sistema

Convenciones generales del sistema

El sistema contiene los elementos específicos de una aplicación Web, para su funcionamiento, como son botones, tablas y diversas formas para despliegue y captura de información.

4.4 Descripción del ambiente de trabajo

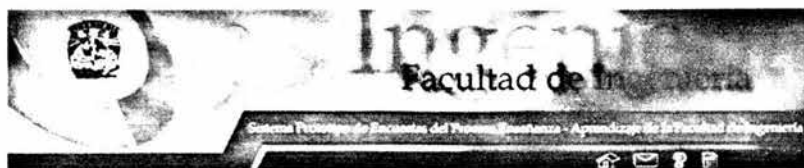
El sistema contiene un módulo de ingreso, que según el tipo de usuario que entra a la aplicación, presenta el entorno de alguno de los siguientes tipos de usuario:

-  Opciones del administrador.
-  Encuesta del alumno.
-  Reportes de profesor, jefe de departamento y división.

A continuación se describen los procesos que se llevan a cabo en cada uno de estos módulos:

4.5 Procesos principales del sistema

Al inicio de la aplicación, se presenta una pantalla de clave o número de cuenta y password o fecha de nacimiento, para el ingreso al sistema:



Bienvenido al Sistema

Ingresar los datos requeridos.

| | |
|---|---|
| C clave o número de cuenta | |
| Password o fecha de nacimiento | Ver Formato (ddmm/aaaa) |
| <input type="button" value="Ingresar"/> | |

4.1. Pantalla de ingreso al sistema

Dicha pantalla, contiene en la parte superior varios íconos con opciones de contacto, preguntas y notas acerca de sistema, así como un icono en forma de casa, que al presionarlo, retorna a la página de inicio. Estas opciones estarán disponibles en todo momento dentro de la aplicación.

En la parte central, se encuentran la forma de ingreso, donde se deben incluir los datos requeridos, así como un link llamado *ver formato*, que muestra información de ayuda.

Dependiendo del tipo de usuario, se presentará la pantalla del módulo correspondiente; en caso de existir un error en los datos, se presenta una pantalla de error.



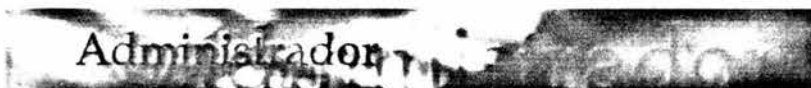
Hay un error, favor de verificar los datos de ingreso.

Corregir

4.2. Señal de error

Opciones del administrador

A continuación, se presenta el módulo de administración, en donde al ingresar se presenta el menú inicial con las opciones que tiene dicho tipo de usuario:



[Alta de Profesor](#)

[Alta de materias para el profesor](#)

[Alta de alumnos](#)

[Alta de materias de alumnos](#)

[Ejecutar procesos estadísticos](#)

[Ver reporte tipo](#)

4.3. Menú del administrador

Altas

Todas las opciones del administrador para altas, utilizan un mismo formato de 3 pantallas: captura, presentación preliminar y confirmación de datos recibidos.

A continuación se muestra el funcionamiento de alta de profesor, para ejemplificar el funcionamiento:

Opciones del Administrador

Alta de Profesor

Clave del profesor:

Nombre:

Apellido Paterno:

Apellido Materno:

Grado:

Ing. ▼

División:

División Educación Continua ▼

Aceptar

Limpiar

4.4. Pantalla de captura.

Al ingresar los datos y presionar aceptar, se presentarán en pantalla los datos capturados, y en su caso, mensajes de error en caso de hallarse un dato incorrecto.

Favor de confirmar los datos a registrar.

| | | |
|--------------------|------------------|--------------------|
| Clave del profesor | 9999 | |
| Nombre | Christian | |
| Apellido Paterno | Luna | |
| Apellido Materno | R)= | Apellido no válido |
| Grado | Ing. | |
| División | DEC | |

Corregir

4.5. Pantalla de presentación preliminar.

Cabe destacar que el sistema contiene validaciones para nombres, grupos, claves, semestres, fechas de nacimiento y números de cuenta.

Para el nombre y apellidos solo se permiten letras incluida la ñ, incluso vocales acentuadas.

La fecha debe ser en un formato de dd/mm/aaaa, en el que se verifican los años bisiestos.

En el caso de números de cuenta se permiten sólo caracteres numéricos, y debe ser de 9 dígitos. Asimismo sólo se permiten números en claves de asignatura y grupo.

Para el semestre, se valida que sea de 4 números en un año válido (entre 2000 y 2132), y el dígito del número de semestre debe ser entre 1 y 3.

Si alguno de los datos ingresados no cumple con las validaciones mencionadas, solo se presenta el botón de corregir que retorna a la página anterior. En el caso de que los datos sean válidos, se habilita el botón de aceptar. En este caso, los datos serán guardados, y se presentará una pantalla de confirmación.

Los datos proporcionados y registrados son los siguientes:

| | |
|--------------------|-----------|
| Clave del profesor | 9998 |
| Nombre | Christian |
| Apellido Paterno: | Luna |
| Apellido Materno: | Rodriguez |
| Grado: | Dr. |
| División: | DIE |

Alta otro

Menu

4.6. Pantalla de confirmación

En el caso de que alguno de los datos cause algún conflicto con la integridad de la base de datos, como pueden ser claves repetidas o grupos ya asignados, se presentará un mensaje como el siguiente:

La clave del profesor proporcionada ya existe en la Base de Datos, favor de verificarla

Corregir

4.7. Aviso

Por lo que, en este caso, se deberán verificar los datos.

Para los otros casos de altas de materias y alumnos, el funcionamiento sería similar al mostrado.

4.6 Procesos estadísticos

El link de ejecutar procesos estadísticos, ofrece la opción al administrador de finalizar la encuesta, una vez terminado el proceso, y así generar los datos necesarios para la generación de los reportes.

Ejecutar procesos de la encuesta.

Ejecutar

4.8 Procesos

Al oprimir, se mostrarán mensajes con los procesos que se van ejecutando, así como posibles mensajes de error.

Ver Reporte tipo

En esta opción, se puede visualizar un reporte de profesor, para un grupo específico, como muestra para el administrador. Para continuar, se deberán ingresar los datos requeridos.

Reporte tipo

Entrada

Clave:

Grupo:

Semestre:

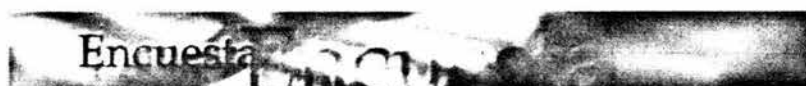
Aceptar

Limpiar

4.9. Reporte tipo

Encuesta del alumno

Para el caso de los alumnos, se presenta una pantalla de selección de materias, en la que, dependiendo si ya contestó o no, podrá proseguir a contestar la encuesta.



Captura de encuesta.

Numero de Cuenta: **094196250**
Nombre: **Christian Mauricio Luna Rodriguez**
Semestre: **2003-2**

Favor de seleccionar materia.

| Grupo | Clave | Materia | Encuesta |
|-------|-------|------------------------------|-----------------------|
| 99 | 0134 | Diseño de Sistemas Digitales | <input type="radio"/> |

Regresar

4.10. Selección de materia

Al seleccionar la encuesta de la materia deseada, se confirmarán los datos y se muestra la encuesta a contestar.

Inicio de encuesta

Bienvenido (a): Christian Mauricio Luna Rodriguez

Para comenzar la encuesta presiona el botón:

Continuar

4.11. Mensaje de confirmación

| | |
|------------|-----------|
| Num Cuenta | 094196250 |
| Clave | 0134 |
| Grupo | 99 |
| Semestre | 2003-2 |

a) SOBRE EL PROFESOR

1.- El interés del profesor para que los alumnos aprendan es:

muy grande grande escaso nulo

2.- La confianza que el profesor inspira en los alumnos para intervenir en clase es:

muy grande grande escaso nulo

3.- Las clases se desarrollan de manera:

muy interesante interesante aburrida muy aburrida

4.- El profesor expone la clase en forma clara:

siempre frecuentemente algunas veces nunca

5.- Los conocimientos del profesor sobre la asignatura son:

4.12. Encuesta

Aquí se deberán contestar todas las preguntas, de lo contrario se mostrará un mensaje de error; los comentarios son opcionales:

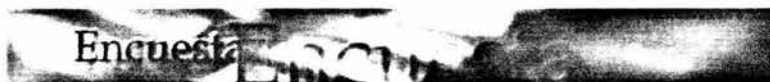


Falta uno o más reactivos por contestar.

Corregir

4.13. Mensaje de error

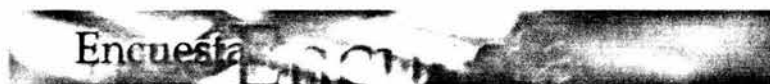
Al contestar todos los reactivos, se muestra una pantalla de confirmación, y se habrá finalizado el proceso.



La encuesta ha sido procesada...

4.14. Aviso

La aplicación cuenta con un sistema de validación de sesión HTTP, que en caso de acceso incorrecto, presentaría un mensaje como el siguiente:



Ingresa nuevamente por favor...

4.15. Aviso.

Reportes de profesor, jefe de departamento y división.

Para estos casos, el sistema identifica las asignaturas asignadas al profesor, o bien los reportes para los jefes de división y departamento; y se podrá entonces seleccionar el tipo de reporte a visualizar, en ese momento, se mostrará la página deseada. Un ejemplo se muestra a continuación:



Universidad Nacional Autónoma de México
Facultad de Ingeniería



Programa de Evaluación del Proceso Enseñanza-Aprendizaje
de la Facultad de Ingeniería

Análisis de Profesor por Pregunta - Respuesta

Profesor: Jorge Valenano Assem
División: División de Ingeniería Eléctrica
Departamento: Departamento de Ingeniería en Computación
Asignatura: Diseño de Sistemas Digitales
Clave: D134
Grupo: 03

Semestre: 2003-2
Fecha: 13/06/2003
No. Intentos (NI): 41
No. Encuestas (NE): 32
% Encuestados (%E): 78.05

Para consultar la relación de preguntas de la encuesta [aquí](#)

| | | | | | | | | | | | |
|----------------------------|----|------|-------------------|----------------------------|----|-------------------|-------|----------------------------|-------------------|------|-------|
| Pregunta 1 | | | Pregunta 2 | | | Pregunta 3 | | | Pregunta 4 | | |
| Opción | NE | Peso | Total | Opción | NE | Peso | Total | Opción | NE | Peso | Total |
| 1 | 15 | 3 | = 45 | 1 | 12 | 3 | = 36 | 1 | 12 | 3 | = 36 |
| 2 | 17 | 2 | = 34 | 2 | 20 | 2 | = 40 | 2 | 20 | 2 | = 40 |
| 3 | 0 | 1 | = 0 | 3 | 0 | 1 | = 0 | 3 | 0 | 1 | = 0 |
| 4 | C | C | = 0 | 4 | 0 | 0 | = 0 | 4 | 0 | 0 | = 0 |
| Máximo = 96 | | | | Máximo = 96 | | | | Máximo = 96 | | | |
| Suma = 79 | | | | Suma = 76 | | | | Suma = 76 | | | |
| Total de Preguntas = 62.29 | | | | Total de Preguntas = 79.17 | | | | Total de Preguntas = 79.17 | | | |
| Pregunta 5 | | | Pregunta 6 | | | Pregunta 7 | | | Pregunta 8 | | |
| Opción | NE | Peso | Total | Opción | NE | Peso | Total | Opción | NE | Peso | Total |
| Máximo = 96 | | | | Máximo = 96 | | | | Máximo = 96 | | | |
| Suma = 78 | | | | Suma = 75 | | | | Suma = 75 | | | |
| Total de Preguntas = 61.25 | | | | Total de Preguntas = 61.25 | | | | Total de Preguntas = 61.25 | | | |

4.16. Reporte

En este caso, se tiene la posibilidad de consultar el contenido de cada pregunta al presionar el link correspondiente. Asimismo es posible consultar la totalidad de las preguntas con el vínculo del inicio del reporte.

CAPÍTULO

5

Manual Técnico

- Manual Técnico
 - Convenciones del Manual
 - Presentación
- Introducción
- Objetivos
- Archivos del sistema
 - Componentes elementales del sistema
 - Extensión de los archivos
 - Programas principales
- Base de datos del sistema
 - Definición de la base de datos
- Estructura del sistema
 - Listado de tablas
 - Índices de la base de datos
- Diagramas de bloques.
 - Módulos y opciones
 - Descripción de los módulos
- Código Fuente
- Relación de Archivos
- Elementos de programación

5.1 Manual Técnico

Convenciones del Manual.

Presentación

Sistema prototipo para el control vía Web de encuestas para el programa de evaluación del proceso Enseñanza – Aprendizaje de la Facultad de Ingeniería.

El sistema es una propuesta de automatización del sistema actual de encuestas que se lleva a cabo en la Facultad de Ingeniería de la U.N.A.M., con el cual se pretende sustituir el mecanismo de llenado por hojas ópticas, con una aplicación Web, que otorgue mayor control, seguridad, conveniencia y mejores prestaciones que el sistema actual.

5.2 Introducción

Antecedentes

Situación actual

Actualmente el programa de evaluación del proceso enseñanza – aprendizaje cuenta con una encuesta impresa de tipo óptica donde el estudiante rellena alvéolos para contestar un total de 29 preguntas y adicionalmente se cuenta con una área abierta de comentarios y sugerencias al profesor que imparte alguna asignatura específica.

Dicho ejercicio de evaluación se realiza mediante la visita de la persona que va a aplicarla; al aula en un horario de clases, posteriormente se distribuyen entre el alumnado las hojas impresas de tipo óptica en donde cada uno llena datos particulares como lo son el nombre del profesor, asignatura, semestre, clave de asignatura, grupo, tipo de clase y número de profesor, una vez echo esto se procede a responde la encuesta en sus 20 preguntas que corresponden al apartado del profesor y 9 de autoevaluación sobre el alumno, además de posibles comentarios y sugerencias. Estas encuestas se guardan en un sobre cerrado para mandarlas al área de captura y procesamiento de datos, para distribuir los

resultados en la coordinación y posteriormente a los respectivos académicos.

5.3 Objetivos

Diseñar e implantar un sistema de información con carácter de programa piloto que facilite y optimice el actual proceso de evaluación de enseñanza–aprendizaje de la Facultad de Ingeniería.

Diseño y manejo de una base de datos histórica con la información recabada de los alumnos de la facultad mediante las encuestas.

Recabar, analizar y presentar los resultados de las encuestas vía Internet mediante una aplicación Web.

Ofrecer los reportes requeridos por parte de profesores, jefes de departamento y coordinadores de carrera, respecto a la información recabada en las encuestas, empleando dicha aplicación.

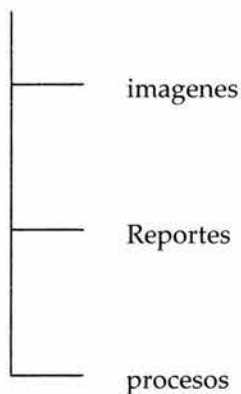
Generar estadísticas tanto del semestre en curso como del histórico de los semestres anteriores.

5.4 Archivos del sistema

Componentes elementales del sistema.

El sistema se compone de un paquete o componente llamado EncuestaDef2, en donde se encuentran fundamentalmente los archivos php que forman al sistema, este paquete tiene tres subpaquetes uno llamado *imagenes*, que contiene los elementos gráficos de la aplicación Web, otro llamado *Reportes* donde se encuentran los programas encargados de mostrar los reportes de distintos tipos, y el último subpaquete llamado *procesos*, que contiene los programas encargados de generar los procesos estadísticos para la encuesta. A continuación se muestra un diagrama con los elementos mencionados:

EncuestaDef2




5.1. Diagrama de paquetes.

Extensión de los archivos.

Los archivos del sistema son de los siguientes tipos por extensión:

- .php Tipos de archivo PHP, páginas dinámicas y programas que constituyen la funcionalidad de la aplicación Web.
- .htm Tipo de archivo HTML, páginas estáticas que constan algunas páginas de presentación y ayuda.
- .swf Tipo de archivo Flash, que son elementos gráficos de presentación en la aplicación.
- .gif
- .jpg Imágenes para la presentación de la aplicación en Web.
- .jpeg

Programas principales.*Paquete EncuestaDef2* Módulo de ingreso al sistema.

| | |
|----------------|---|
| ingreso.php | Página de ingreso al sistema. |
| ingresoPre.php | Resultado de la página anterior, si es correcto, redirecciona al módulo adecuado, caso contrario, muestra mensaje de error. |

 Módulo de administración.

| | |
|--|---|
| ingresoAdmin.php | Menú de ingreso al módulo. |
| admini.php | Alta de profesor. |
| adminiPre.php | Presentación preliminar de Alta de profesor. |
| adminiPro.php | Pantalla de resultado de Alta de profesor. |
| admini2.php | Alta de asignaturas que imparte el profesor. |
| admini2Pre.php imparte el profesor. | Presentación preliminar de alta de asignaturas que imparte el profesor. |
| admini2pro.php imparte el profesor. | Pantalla de resultado de alta de asignaturas que imparte el profesor. |
| ad_alum.php | Alta de alumnos. |
| ad_alumPre.php | Presentación preliminar de alta de alumnos. |
| ad_alumPro.php | Pantalla de resultado de alta de alumnos. |
| ad_alum2.php | Alta de asignaturas para alumnos. |

| | |
|----------------------------------|--|
| ad_alum2Pre.php para alumnos. | Presentación preliminar de alta de asignaturas para alumnos. |
| ad_alumPro2.php alumnos. | Pantalla de resultado de alta de asignaturas para alumnos. |
| reporte-intro.php | Pantalla de ingreso para ver reporte de muestra. |

Módulo de encuestas.

| | |
|-----------------------------|--|
| enc_intro1.php encuesta. | Pantalla de selección de materia a realizar la encuesta. |
| enc_intro0pro.php | Pantalla intermedia para iniciar encuesta. |
| encuestaTotal.php | Pantalla de captura de encuesta. |
| enc_intro_pro.php | Pantalla de resultado al contestar la encuesta. |

Otros programas.

| | |
|------------------|---|
| conec.php | Componente para la conexión a la base de datos. |
| validaciones.php | Componente para validaciones de datos. |

Paquete reportes

| | |
|-----------------|---|
| reporte-ppr.php | Reporte de profesor por pregunta - respuesta. |
| reporte-pxp.php | Reporte de asignatura - profesor por pregunta. |
| reporte-pxc.php | Reporte de asignatura - profesor por concepto . |

| | |
|-----------------|--|
| reporte-ddp.php | Reporte de división - departamento por pregunta. |
| reporte-ddc.php | Reporte de división - departamento por concepto. |
| reporte_tot.php | Reporte general del profesor. |

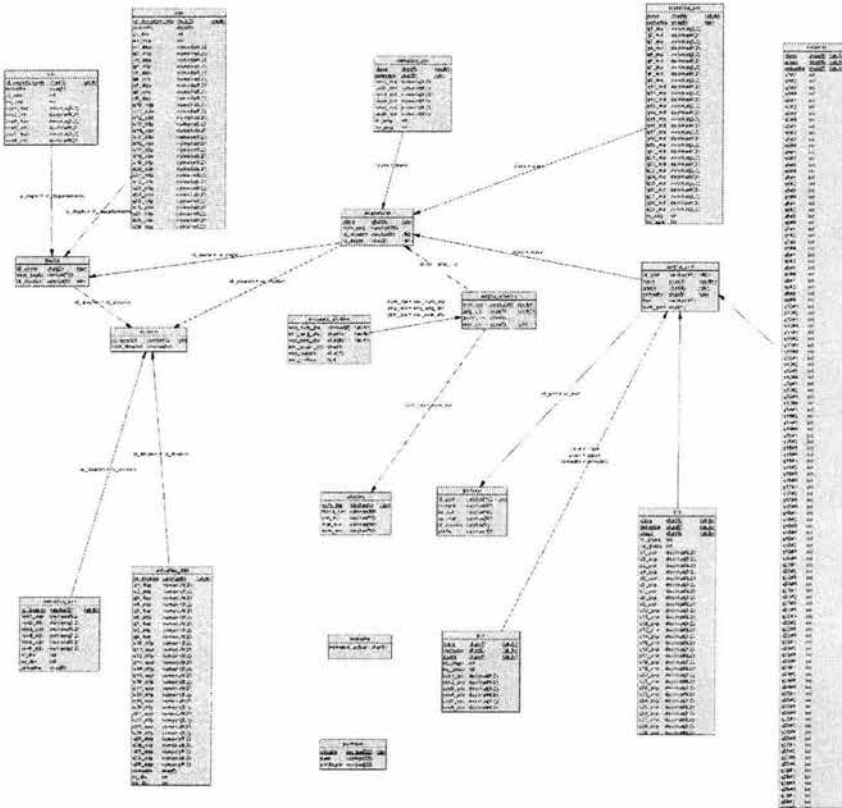
 *Paquete procesos.*

| | |
|--------------------------------|--|
| procesa_encuesta.php | Pantalla inicial para inicio de procesos estadísticos. |
| procesos_todos.php | Ejecuta todos los procesos estadísticos. |
| proceso01.php | Programa para procesos estadísticos por grupo. |
| proceso02.php asignatura. | Programa para procesos estadísticos por asignatura. |
| proceso03.php departamento. | Programa para procesos estadísticos por departamento. |
| proceso04.php | Programa para procesos estadísticos por división. |

5.5 Base de datos del sistema.

Definición de la base de datos.

Diagrama Entidad – Relación:



5.2. Diagrama Entidad – Relación

Nota: Ver diagrama Entidad –Relación anexo al final de este trabajo.

Área de trabajo.

La base de datos se encuentra alojada en un servidor de prueba ubicado en el departamento de Microsoft de la Facultad de Ingeniería, con las siguientes características:

Sistema operativo: Microsoft Windows 2000 server.

Manejador de Base de Datos: Microsoft SQL Server 2000.

5.6 Estructura del sistema.

El sistema esta compuesto de una base de datos única con las siguientes características:

19 Tablas de usuario.

19 Llaves primarias.

18 Índices.

17 Llaves foráneas.

El sistema no cuenta con vistas, triggers ni procedimientos almacenados.

Listado de tablas.

| Nombre | Código |
|-----------------|-----------------|
| division | division |
| encuesta_alumno | encuesta_alumno |
| deptos | deptos |
| ddp | ddp |
| ddc | ddc |
| pxp | pxp |
| permisos | permisos |

| Nombre | Código |
|---------------|---------------|
| pxc | pxc |
| profesor | profesor |
| semestre | semestre |
| actualiza_ddp | actualiza_ddp |
| actualiza_pxc | actualiza_pxc |
| actualiza_ddc | actualiza_ddc |
| actualiza | actualiza |
| asignaturas | asignaturas |
| asigna_prof | asigna_prof |
| actualiza_pxp | actualiza_pxp |
| asigna_alumno | asigna_alumno |
| alumno | alumno |

Creación de las tablas (DDL).

A continuación se presentan los scripts correspondientes a la creación de la base de datos:

```
create table alumno (
num_cta      varchar(9)          not null,
fecha_nac   varchar(10)         not null,
pat_alu     varchar(50)         null,
mat_alu     varchar(50)         null,
nom_alu     varchar(50)         not null,
constraint PK_ALUMNO primary key (num_cta)
)
```

```
create table profesor (
id_prof     varchar(15)         not null,
nombre     varchar(50)         not null,
ap_pat     varchar(50)         null,
ap_mat     varchar(50)         null,
id_division varchar(6)          not null,
```

```
grado          varchar(10)          null,  
constraint PK_PROFESOR primary key (id_prof)  
)
```

```
create table semestre (  
semestre_actual char(6)          null  
)
```

```
create table permisos (  
usuario        varchar(20)          not null,  
pass           varchar(20)          null,  
privilegio     varchar(20)          null,  
constraint PK_PERMISOS primary key (usuario)  
)
```

```
create table division (  
id_division    varchar(6)          not null,  
nom_division   varchar(80)         not null,  
constraint PK_DIVISION primary key (id_division)  
)
```

```
create table deptos (  
id_depto       char(2)            not null,  
desc_depto     varchar(70)         null,  
id_division    varchar(6)         not null  
              default '("DEC")',  
constraint PK_DEPTOS primary key (id_depto)  
)
```

```
alter table deptos  
  add constraint FK__deptos__id_divis__2704CA5F foreign key  
(id_division)  
  references division (id_division)
```

```
create table asignaturas (  
clave          char(4)            not null,
```



```

nom_asig      varchar(80)      not null,
id_division   varchar(6)      not null,
id_depto      char(2)        not null
              default '(6)',
constraint PK_ASIGNATURAS primary key (clave)
)

```

```

alter table asignaturas
add constraint FK__asignatur__id_de__160F4887 foreign key (id_depto)
references deptos (id_depto)

```

```

alter table asignaturas
add constraint FK__asignatur__id_di__1FCDBCEB foreign key
(id_division)
references division (id_division)

```

```

create table asigna_prof (
id_prof       varchar(15)      not null,
clave         char(4)          not null,
grupo         char(4)          not null,
semestre      char(6)          not null,
tipo          varchar(15)      not null,
num_prof      char(1)          not null,
constraint PK_ASIGNA_PROF primary key (clave, grupo, semestre)
)

```

```

alter table asigna_prof
add constraint FK__asigna_pr__clave__4CA06362 foreign key (clave)
references asignaturas (clave)

```

```

alter table asigna_prof
add constraint FK__asigna_pr__id_pr__4BAC3F29 foreign key (id_prof)
references profesor (id_prof)

```

```

create table actualiza (
clave         char(4)          not null,
grupo         char(4)          not null,
semestre      char(6)          not null,
)

```

| | | |
|-------|-----|-------|
| q1#1 | int | null, |
| q1#2 | int | null, |
| q1#3 | int | null, |
| q1#4 | int | null, |
| q2#1 | int | null, |
| q2#2 | int | null, |
| q2#3 | int | null, |
| q2#4 | int | null, |
| q3#1 | int | null, |
| q3#2 | int | null, |
| q3#3 | int | null, |
| q3#4 | int | null, |
| q4#1 | int | null, |
| q4#2 | int | null, |
| q4#3 | int | null, |
| q4#4 | int | null, |
| q5#1 | int | null, |
| q5#2 | int | null, |
| q5#3 | int | null, |
| q5#4 | int | null, |
| q6#1 | int | null, |
| q6#2 | int | null, |
| q6#3 | int | null, |
| q6#4 | int | null, |
| q7#1 | int | null, |
| q7#2 | int | null, |
| q7#3 | int | null, |
| q7#4 | int | null, |
| q8#1 | int | null, |
| q8#2 | int | null, |
| q8#3 | int | null, |
| q8#4 | int | null, |
| q9#1 | int | null, |
| q9#2 | int | null, |
| q9#3 | int | null, |
| q9#4 | int | null, |
| q10#1 | int | null, |
| q10#2 | int | null, |

| | | |
|-------|-----|-------|
| q10#3 | int | null, |
| q10#4 | int | null, |
| q11#1 | int | null, |
| q11#2 | int | null, |
| q11#3 | int | null, |
| q11#4 | int | null, |
| q12#1 | int | null, |
| q12#2 | int | null, |
| q12#3 | int | null, |
| q12#4 | int | null, |
| q13#1 | int | null, |
| q13#2 | int | null, |
| q13#3 | int | null, |
| q13#4 | int | null, |
| q14#1 | int | null, |
| q14#2 | int | null, |
| q14#3 | int | null, |
| q14#4 | int | null, |
| q15#1 | int | null, |
| q15#2 | int | null, |
| q15#3 | int | null, |
| q15#4 | int | null, |
| q16#1 | int | null, |
| q16#2 | int | null, |
| q16#3 | int | null, |
| q16#4 | int | null, |
| q17#1 | int | null, |
| q17#2 | int | null, |
| q17#3 | int | null, |
| q17#4 | int | null, |
| q18#1 | int | null, |
| q18#2 | int | null, |
| q18#3 | int | null, |
| q18#4 | int | null, |
| q19#1 | int | null, |
| q19#2 | int | null, |
| q19#3 | int | null, |
| q19#4 | int | null, |

| | | |
|-------|-----|-------|
| q20#1 | int | null, |
| q20#2 | int | null, |
| q20#3 | int | null, |
| q20#4 | int | null, |
| q21#1 | int | null, |
| q21#2 | int | null, |
| q21#3 | int | null, |
| q21#4 | int | null, |
| q22#1 | int | null, |
| q22#2 | int | null, |
| q22#3 | int | null, |
| q22#4 | int | null, |
| q23#1 | int | null, |
| q23#2 | int | null, |
| q23#3 | int | null, |
| q23#4 | int | null, |
| q24#1 | int | null, |
| q24#2 | int | null, |
| q24#3 | int | null, |
| q24#4 | int | null, |
| q25#1 | int | null, |
| q25#2 | int | null, |
| q25#3 | int | null, |
| q25#4 | int | null, |
| q26#1 | int | null, |
| q26#2 | int | null, |
| q26#3 | int | null, |
| q26#4 | int | null, |
| q27#1 | int | null, |
| q27#2 | int | null, |
| q27#3 | int | null, |
| q27#4 | int | null, |
| q28#1 | int | null, |
| q28#2 | int | null, |
| q28#3 | int | null, |
| q28#4 | int | null, |
| q29#1 | int | null, |
| q29#2 | int | null, |

```

q29#3      int          null,
q29#4      int          null,
constraint PK_ACTUALIZA primary key (clave, grupo, semestre)
)

```

```

alter table actualiza
  add constraint fk_actualiza foreign key (clave, grupo, semestre)
  references asigna_prof (clave, grupo, semestre)

```

```

create table asigna_alumno (
num_cta      varchar(9)      not null,
asig_alu     char(4)         not null,
grupo_alu    char(4)         not null,
sem_alu      char(6)         not null,
constraint PK_ASIGNA_ALUMNO primary key (num_cta, asig_alu,
sem_alu)
)

```

```

alter table asigna_alumno
  add constraint FK__asigna_al__num_c__75A278F5 foreign key
(num_cta)
  references alumno (num_cta)

```

```

alter table asigna_alumno
  add constraint FK__asigna_al__asig__74AE54BC foreign key (asig_alu)
  references asignaturas (clave)

```

```

create table encuesta_alumno (
enc_num_cta  varchar(9)      not null,
enc_asig_alu char(4)         not null,
enc_sem_alu  char(6)         not null,
enc_grupo_alu char(4)         not null,
enc_realizo  char(1)         not null,
enc_comen    text            null,
constraint PK_ENCUESTA_ALUMNO primary key (enc_num_cta,
enc_asig_alu, enc_sem_alu)
)

```

```
alter table encuesta_alumno
```

```
  add constraint fk_enc_alumno foreign key (enc_num_cta, enc_asig_alu,  
  enc_sem_alu)
```

```
  references asigna_alumno (num_cta, asig_alu, sem_alu)
```

```
create table pxp (
```

```
  clave          char(4)          not null,  
  semestre      char(6)          not null,  
  grupo         char(4)          not null,  
  ni_grupo      int              null,  
  ne_grupo      int              null,  
  q1_pxp        decimal(8,2)     null,  
  q2_pxp        decimal(8,2)     null,  
  q3_pxp        decimal(8,2)     null,  
  q4_pxp        decimal(8,2)     null,  
  q5_pxp        decimal(8,2)     null,  
  q6_pxp        decimal(8,2)     null,  
  q7_pxp        decimal(8,2)     null,  
  q8_pxp        decimal(8,2)     null,  
  q9_pxp        decimal(8,2)     null,  
  q10_pxp       decimal(8,2)     null,  
  q11_pxp       decimal(8,2)     null,  
  q12_pxp       decimal(8,2)     null,  
  q13_pxp       decimal(8,2)     null,  
  q14_pxp       decimal(8,2)     null,  
  q15_pxp       decimal(8,2)     null,  
  q16_pxp       decimal(8,2)     null,  
  q17_pxp       decimal(8,2)     null,  
  q18_pxp       decimal(8,2)     null,  
  q19_pxp       decimal(8,2)     null,  
  q20_pxp       decimal(8,2)     null,  
  q21_pxp       decimal(8,2)     null,  
  q22_pxp       decimal(8,2)     null,  
  q23_pxp       decimal(8,2)     null,  
  q24_pxp       decimal(8,2)     null,  
  q25_pxp       decimal(8,2)     null,  
  q26_pxp       decimal(8,2)     null,  
  q27_pxp       decimal(8,2)     null,
```

```
q28_pxp      decimal(8,2)      null,
q29_pxp      decimal(8,2)      null,
constraint PK_PXP primary key (clave, semestre, grupo)
)
alter table pxp
  add constraint fk_pxp foreign key (clave, grupo, semestre)
  references asigna_prof (clave, grupo, semestre)
```

```
create table pxc (
clave        char(4)          not null,
semestre     char(6)          not null,
grupo        char(4)          not null,
ni_grupo     int              null,
ne_grupo     int              null,
con1_pxc     decimal(8,2)     null,
con2_pxc     decimal(8,2)     null,
con3_pxc     decimal(8,2)     null,
con4_pxc     decimal(8,2)     null,
con5_pxc     decimal(8,2)     null,
con6_pxc     decimal(8,2)     null,
constraint PK_PXC primary key (clave, semestre, grupo)
)
```

```
alter table pxc
  add constraint fk_pxc foreign key (clave, grupo, semestre)
  references asigna_prof (clave, grupo, semestre)
```

```
create table actualiza_pxp (
clave        char(4)          not null,
semestre     char(6)          not null,
q1_md        decimal(8,2)     null,
q2_md        decimal(8,2)     null,
q3_md        decimal(8,2)     null,
q4_md        decimal(8,2)     null,
q5_md        decimal(8,2)     null,
```

```

q6_md          decimal(8,2)      null,
q7_md          decimal(8,2)      null,
q8_md          decimal(8,2)      null,
q9_md          decimal(8,2)      null,
q10_md         decimal(8,2)      null,
q11_md         decimal(8,2)      null,
q12_md         decimal(8,2)      null,
q13_md         decimal(8,2)      null,
q14_md         decimal(8,2)      null,
q15_md         decimal(8,2)      null,
q16_md         decimal(8,2)      null,
q17_md         decimal(8,2)      null,
q18_md         decimal(8,2)      null,
q19_md         decimal(8,2)      null,
q20_md         decimal(8,2)      null,
q21_md         decimal(8,2)      null,
q22_md         decimal(8,2)      null,
q23_md         decimal(8,2)      null,
q24_md         decimal(8,2)      null,
q25_md         decimal(8,2)      null,
q26_md         decimal(8,2)      null,
q27_md         decimal(8,2)      null,
q28_md         decimal(8,2)      null,
q29_md         decimal(8,2)      null,
ni_asig        int              null,
ne_asig        int              null,
constraint PK_ACTUALIZA_PXP primary key (clave, semestre)
)

```

```

alter table actualiza_pxp
  add constraint FK__actualiza__clave__14E61A24 foreign key (clave)
  references asignaturas (clave)

```

```

create table actualiza_pxc (
  clave          char(4)          not null,
  semestre       char(6)          not null,
  con1_md        numeric(8,2)     null,

```



```

con2_md      numeric(8,2)      null,
con3_md      numeric(8,2)      null,
con4_md      numeric(8,2)      null,
con5_md      numeric(8,2)      null,
con6_md      numeric(8,2)      null,
ni_asig      int          null,
ne_asig      int          null,
constraint PK_ACTUALIZA_PXC primary key (clave, semestre)
)

```

```

alter table actualiza_pxc
  add constraint FK_actualiza__clave__13F1F5EB foreign key (clave)
  references asignaturas (clave)

```

```

create table ddp (
id_departamento char(2)      not null,
semestre         char(6)      not null,
ni_dep           int          null,
ne_dep           int          null,
q1_ddp          numeric(8,2)  null,
q2_ddp          numeric(8,2)  null,
q3_ddp          numeric(8,2)  null,
q4_ddp          numeric(8,2)  null,
q5_ddp          numeric(8,2)  null,
q6_ddp          numeric(8,2)  null,
q7_ddp          numeric(8,2)  null,
q8_ddp          numeric(8,2)  null,
q9_ddp          numeric(8,2)  null,
q10_ddp         numeric(8,2)  null,
q11_ddp         numeric(8,2)  null,
q12_ddp         numeric(8,2)  null,
q13_ddp         numeric(8,2)  null,
q14_ddp         numeric(8,2)  null,
q15_ddp         numeric(8,2)  null,
q16_ddp         numeric(8,2)  null,
q17_ddp         numeric(8,2)  null,
q18_ddp         numeric(8,2)  null,
q19_ddp         numeric(8,2)  null,

```

```

q20_ddp      numeric(8,2)      null,
q21_ddp      numeric(8,2)      null,
q22_ddp      numeric(8,2)      null,
q23_ddp      numeric(8,2)      null,
q24_ddp      numeric(8,2)      null,
q25_ddp      numeric(8,2)      null,
q26_ddp      numeric(8,2)      null,
q27_ddp      numeric(8,2)      null,
q28_ddp      numeric(8,2)      null,
q29_ddp      numeric(8,2)      null,
constraint PK_DDP primary key (id_departamento)
)

```

```

alter table ddp
  add constraint FK_ddp__id_departam__47A6A41B foreign key
(id_departamento)
  references deptos (id_depto)

```

```

create table ddc (
id_departamento char(2)      not null,
semestre        char(6)      not null,
ni_dep          int          null,
ne_dep          int          null,
con1_ddc        decimal(8,2) null,
con2_ddc        decimal(8,2) null,
con3_ddc        decimal(8,2) null,
con4_ddc        decimal(8,2) null,
con5_ddc        decimal(8,2) null,
con6_ddc        decimal(8,2) null,
constraint PK_DDC primary key (id_departamento)
)

```

```

alter table ddc
  add constraint FK_ddc__id_departam__46B27FE2 foreign key
(id_departamento)
  references deptos (id_depto)

```

```

create table actualiza_ddp (
id_division      varchar(6)           not null,
q1_ddp          numeric(8,2)       null,
q2_ddp          numeric(8,2)       null,
q3_ddp          numeric(8,2)       null,
q4_ddp          numeric(8,2)       null,
q5_ddp          numeric(8,2)       null,
q6_ddp          numeric(8,2)       null,
q7_ddp          numeric(8,2)       null,
q8_ddp          numeric(8,2)       null,
q9_ddp          numeric(8,2)       null,
q10_ddp         numeric(8,2)       null,
q11_ddp         numeric(8,2)       null,
q12_ddp         numeric(8,2)       null,
q13_ddp         numeric(8,2)       null,
q14_ddp         numeric(8,2)       null,
q15_ddp         numeric(8,2)       null,
q16_ddp         numeric(8,2)       null,
q17_ddp         numeric(8,2)       null,
q18_ddp         numeric(8,2)       null,
q19_ddp         numeric(8,2)       null,
q20_ddp         numeric(8,2)       null,
q21_ddp         numeric(8,2)       null,
q22_ddp         numeric(8,2)       null,
q23_ddp         numeric(8,2)       null,
q24_ddp         numeric(8,2)       null,
q25_ddp         numeric(8,2)       null,
q26_ddp         numeric(8,2)       null,
q27_ddp         numeric(8,2)       null,
q28_ddp         numeric(8,2)       null,
q29_ddp         numeric(8,2)       null,
semestre        char(6)            null,
ni_div          int                 null,
ne_div          int                 null,
constraint PK_ACTUALIZA_DDP primary key (id_division)
)

alter table actualiza_ddp

```

```
add constraint FK__actualiza__id_di__45BE5BA9 foreign key
(id_division)
references division (id_division)
```

```
create table actualiza_ddc (
id_division      varchar(6)          not null,
con1_ddc        numeric(8,2)        null,
con2_ddc        numeric(8,2)        null,
con3_ddc        numeric(8,2)        null,
con4_ddc        numeric(8,2)        null,
con5_ddc        numeric(8,2)        null,
con6_ddc        numeric(8,2)        null,
ni_div          int                 null,
ne_div          int                 null,
semestre        char(6)             null,
constraint PK_ACTUALIZA_DDC primary key (id_division)
)
```

```
alter table actualiza_ddc
add constraint FK__actualiza__id_di__44CA3770 foreign key
(id_division)
references division (id_division)
```

Índices de la base de datos

Llaves:

| Nombre | Código | Tabla |
|---------------------------|---------------------------|-----------------|
| PK_division_1CF15040 | PK_division_1CF15040 | division |
| pk_enc_alumno | pk_enc_alumno | encuesta_alumno |
| PK_deptos_09A971A2 | PK_deptos_09A971A2 | deptos |
| PK_ddp | PK_ddp | ddp |
| PK_ddc | PK_ddc | ddc |
| pk_pxc | pk_pxc | pxc |
| tencuesta_alumno | tencuesta_alumno | encuesta_alumno |
| PK_profesor_164452B1 | PK_profesor_164452B1 | profesor |
| PK_permisos_06CD04F7 | PK_permisos_06CD04F7 | permisos |
| pk_pxp | pk_pxp | pxp |
| PK_actualiza_ddp_2BFE89A6 | PK_actualiza_ddp_2BFE89A6 | actualiza_ddp |
| pk_actpxc | pk_actpxc | actualiza_pxc |
| PK_actualiza_ddc_2B0A656D | PK_actualiza_ddc_2B0A656D | actualiza_ddc |
| pk_actualiza | pk_actualiza | actualiza |
| PK_asignaturas_182C9B23 | PK_asignaturas_182C9B23 | asignaturas |
| pk_AsProf | pk_AsProf | asigna_prof |
| pk_actpxp | pk_actpxp | actualiza_pxp |
| pk_alumno | pk_alumno | asigna_alumno |
| PK_alumno_4E88ABD4 | PK_alumno_4E88ABD4 | alumno |

Índices:

| Nombre | Código | Unique | Cluster | P K | F K | A K | Tabla |
|----------------------------|----------------------------|--------|---------|--------|--------|--------|-----------------|
| PK_deptos_09A971A2 | PK_deptos_09A971A2 | V | F | F | F | F | deptos |
| PK_division_1CF15040 | PK_division_1CF15040 | V | F | F | F | F | division |
| PK_ddp | PK_ddp | V | F | F | F | F | ddp |
| PK_ddc | PK_ddc | V | F | F | F | F | ddc |
| pk_pxc | pk_pxc | V | F | F | F | F | pxc |
| pk_enc_alumno | pk_enc_alumno | V | F | F | F | F | encuesta_alumno |
| PK_profesor_164452B1 | PK_profesor_164452B1 | V | F | F | F | F | profesor |
| PK_permisos_06CD04F7 | PK_permisos_06CD04F7 | V | F | F | F | F | permisos |
| pk_pxp | pk_pxp | V | F | F | F | F | pxp |
| PK_actualiza_ddp_2BF E89A6 | PK_actualiza_ddp_2BF E89A6 | V | F | F | F | F | actualiza_ddp |
| pk_actpxc | pk_actpxc | V | F | F | F | F | actualiza_pxc |
| PK_actualiza_ddc_2B0 A656D | PK_actualiza_ddc_2B0 A656D | V | F | F | F | F | actualiza_ddc |

| Nombre | Código | Unique | Cluster | P K | F K | A K | Tabla |
|---------------------------------|---------------------------------|--------|---------|--------|--------|--------|-------------------|
| pk_actualiza | pk_actualiza | V | F | F | F | F | actualiza |
| | | V | F | F | F | F | asignaturas |
| PK_asignat uras_182C9 B23 | PK_asignat uras_182C9 B23 | | | | | | |
| pk_AsProf | pk_AsProf | V | F | F | F | F | asigna_prof |
| pk_actpxp | pk_actpxp | V | F | F | F | F | actualiza_pxp |
| pk_alumno | pk_alumno | V | F | F | F | F | asigna_alumn o |
| | | V | F | F | F | F | alumno |
| PK_alumno _4E88ABD 4 | PK_alumno _4E88ABD 4 | | | | | | |

Relación entre tablas:

| Nombre | Código | Tabla padre | Tabla hija |
|---------------------------------|---------------------------------|-------------|-------------|
| FK_asignatur_id _di_1FCDBCEB | FK_asignatur_id _di_1FCDBCEB | division | asignaturas |
| FK_ddc_id_dep artam_46B27FE2 | FK_ddc_id_dep artam_46B27FE2 | deptos | ddc |
| FK_asignatur_id _de_160F4887 | FK_asignatur_id _de_160F4887 | deptos | asignaturas |
| FK_asigna_pr_i d_pr_4BAC3F29 | FK_asigna_pr_i d_pr_4BAC3F29 | profesor | asigna_prof |

| Nombre | Código | Tabla padre | Tabla hija |
|-------------------------------------|-------------------------------------|-------------------|---------------------|
| fk_pxc | fk_pxc | asigna_prof | pxc |
| | | deptos | ddp |
| FK_ddp_id_dep artam_47A6A41 B | FK_ddp_id_dep artam_47A6A41 B | | |
| fk_enc_alumno | fk_enc_alumno | asigna_alum no | encuesta_alu mno |
| | | division | deptos |
| FK_deptos_id_d ivis_2704CA5F | FK_deptos_id_d ivis_2704CA5F | | |
| fk_pxp | fk_pxp | asigna_prof | pxp |
| | | division | |
| FK_actualiza_id _di_44CA3770 | FK_actualiza_id _di_44CA3770 | | actualiza_ddc |
| | | division | |
| FK_actualiza_id _di_45BE5BA9 | FK_actualiza_id _di_45BE5BA9 | | actualiza_ddp |
| fk_actualiza | fk_actualiza | asigna_prof | actualiza |
| | | asignaturas | asigna_prof |
| FK_asigna_pr_c lave_4CA06362 | FK_asigna_pr_c lave_4CA06362 | | |
| | | asignaturas | |
| FK_asigna_al_a sig__74AE54BC | FK_asigna_al_a sig__74AE54BC | | asigna_alumn o |
| | | asignaturas | |
| FK_actualiza_cl ave_13F1F5EB | FK_actualiza_cl ave_13F1F5EB | | actualiza_pxc |
| | | alumno | |
| FK_asigna_al_n um_c_75A278F5 | FK_asigna_al_n um_c_75A278F5 | | asigna_alumn o |
| | | asignaturas | |
| FK_actualiza_cl | FK_actualiza_cl | | actualiza_pxp |

| Nombre | Código | Tabla padre | Tabla hija |
|--------------|--------------|-------------|------------|
| ave_14E61A24 | ave_14E61A24 | | |

Lenguaje de manipulación de datos (DML)

A continuación se presentan algunos ejemplos de estos elementos de cada uno de los módulos de la aplicación:

Módulo de ingreso al sistema:

```
select privilegio from permisos where usuario='$login' and pass='$pass'
```

Módulo de administración:

```
insert into profesor (id_prof,nombre,ap_pat,ap_mat,grado,id_division)
values ('$id_prof','$nombre','$ap_pat','$ap_mat','$grado','$division')
```

```
insert into asigna_prof (id_prof,clave,grupo,semestre,tipo,num_prof)
values ('$id_prof','$clave','$gpo','$semestre','$tipo','$nump')
```

```
insert into actualiza (clave,grupo,semestre) values
('$clave','$gpo','$semestre')
```

```
update actualiza set $campo=0 where clave='$clave' and grupo='$gpo' and
semestre='$semestre'
```

```
insert into alumno (num_cta,fecha_nac,nom_alu,pat_alu,mat_alu) values
('$num_cta','$fecha_nac','$nom_alu','$pat_alu','$mat_alu')
```

```
insert into permisos (usuario,pass,privilegio) values
('$num_cta','$fecha_nac','$usuario')
```

```
insert into asigna_alumno (num_cta,asig_alu,grupo_alu,sem_alu) values
('$num_cta','$asig_alu','$grupo_alu','$sem_alu')
```

```
insert into encuesta_alumno
(enc_num_cta,enc_asig_alu,enc_grupo_alu,enc_sem_alu,enc_realizo)
values ('$num_cta','$asig_alu','$grupo_alu','$sem_alu','0')
```

Módulo de encuestas:

```
select aa.asig_alu, aa.grupo_alu, a.nom_asig, ea.enc_realizo
      from asigna_alumno aa, asignaturas a,
encuesta_alumno ea
      where a.clave = aa.asig_alu
      and (ea.enc_num_cta = aa.num_cta and
ea.enc_asig_alu = aa.asig_alu and ea.enc_sem_alu = aa.sem_alu)
      and num_cta = '$num_cta'
      and sem_alu = '$sem'
select enc_realizo from encuesta_alumno where enc_num_cta='$num_cta'
AND enc_asig_alu='$clave' and enc_grupo_alu='$grupo' and
enc_sem_alu='$semestre'

update actualiza set $q1=$q1+1 where clave='$clave' and grupo='$grupo'
and semestre='$semestre'
```

Módulo reportes:

```
select profesor.nombre nombre,profesor.ap_pat pat,profesor.ap_mat mat
from profesor,asigna_prof
where asigna_prof.id_prof=profesor.id_prof
and asigna_prof.clave = '$clave' and asigna_prof.grupo='$grupo'
and asigna_prof.semestre = '$sem'

select division.nom_division nombrediv, deptos.desc_depto nombredep
from division,asignaturas,deptos
where division.id_division = asignaturas.id_division
and deptos.id_depto = asignaturas.id_depto
and asignaturas.clave = '$clave'

select count(*) conteo from asigna_alumno
where asig_alu = '$clave' and grupo_alu = '$grupo'
```

```
and sem_alu = '$sem'
```

```
select q1#1,q1#2,q1#3,q1#4 from actualiza
where clave = '$clave' and grupo = '$grupo'
and semestre = '$sem'
```

```
select (select p.id_prof
        from profesor p, asigna_prof ap
        where p.id_prof = ap.id_prof and
              (ap.grupo = pxp.grupo and ap.clave = pxp.clave and ap.semestre =
pxp.semestre)) id_prof,
        (select p.nombre
        from profesor p, asigna_prof ap
        where p.id_prof = ap.id_prof and
              (ap.grupo = pxp.grupo and ap.clave = pxp.clave and ap.semestre =
pxp.semestre)) nombre,
        (select p.ap_pat
        from profesor p, asigna_prof ap
        where p.id_prof = ap.id_prof and
              (ap.grupo = pxp.grupo and ap.clave = pxp.clave and ap.semestre =
pxp.semestre)) ap_pat,
        (select p.ap_mat
        from profesor p, asigna_prof ap
        where p.id_prof = ap.id_prof and
              (ap.grupo = pxp.grupo and ap.clave = pxp.clave and ap.semestre =
pxp.semestre)) ap_mat,
        pxp.grupo, pxp.ni_grupo, pxp.ne_grupo, pxp.q1_pxp,
        pxp.q2_pxp, pxp.q3_pxp, pxp.q4_pxp, pxp.q5_pxp,
        pxp.q6_pxp, pxp.q7_pxp, pxp.q8_pxp, pxp.q9_pxp,
        pxp.q10_pxp, pxp.q11_pxp, pxp.q12_pxp,
        pxp.q13_pxp, pxp.q14_pxp, pxp.q15_pxp, pxp.q16_pxp,
        pxp.q17_pxp, pxp.q18_pxp, pxp.q19_pxp
        from pxp
        where pxp.clave = '$clave' and pxp.semestre = '$sem'
```

```
select id_departamento, (select d.desc_depto from deptos d, ddp where
d.id_depto = ddp.id_departamento) nom_dep,
```

```

ni_dep, ne_dep, q1_ddp, q2_ddp, q3_ddp, q4_ddp, q5_ddp, q6_ddp,
q7_ddp, q8_ddp,
q9_ddp, q10_ddp, q11_ddp, q12_ddp, q13_ddp, q14_ddp, q15_ddp,
q16_ddp, q17_ddp,
q18_ddp, q19_ddp
from ddp, deptos d
where d.id_depto = ddp.id_departamento
and d.id_division = '$div'

```

Módulo procesos.

```

select sum(ni_grupo) ni_asig, sum(ne_grupo) ne_asig, sum(q1_pxp)
suma1, sum(q2_pxp) suma2,
  sum(q3_pxp) suma3, sum(q4_pxp) suma4, sum(q5_pxp) suma5,
sum(q6_pxp) suma6, sum(q7_pxp) suma7, sum(q8_pxp) suma8,
  sum(q9_pxp) suma9, sum(q10_pxp) suma10, sum(q11_pxp) suma11,
sum(q12_pxp) suma12, sum(q13_pxp) suma13,
  sum(q14_pxp) suma14, sum(q15_pxp) suma15, sum(q16_pxp) suma16,
sum(q17_pxp) suma17, sum(q18_pxp) suma18,
  sum(q19_pxp) suma19, sum(q20_pxp) suma20, sum(q21_pxp) suma21,
sum(q22_pxp) suma22, sum(q23_pxp) suma23,
  sum(q24_pxp) suma24, sum(q25_pxp) suma25, sum(q26_pxp) suma26,
sum(q27_pxp) suma27, sum(q28_pxp) suma28,
  sum(q29_pxp) suma29 from pxp where clave = '$clave' and semestre =
'$semestre'

```

```

insert into actualiza_pxp (semestre, clave, ni_asig, ne_asig, q1_md, q2_md,
q3_md, q4_md, q5_md, q6_md, q7_md, q8_md, q9_md,
  q10_md, q11_md, q12_md, q13_md, q14_md, q15_md, q16_md, q17_md,
q18_md, q19_md,
  q20_md, q21_md, q22_md, q23_md, q24_md, q25_md, q26_md, q27_md,
q28_md, q29_md)
values ('$semestre', '$clave', $ni_asig, $ne_asig, $suma1, $suma2, $suma3,
$suma4, $suma5, $suma6, $suma7, $suma8, $suma9, $suma10,
  $suma11, $suma12, $suma13, $suma14, $suma15, $suma16, $suma17,
$suma18, $suma19, $suma20,
  $suma21, $suma22, $suma23, $suma24, $suma25, $suma26, $suma27,
$suma28, $suma29)

```

```

select sum(ni_asig) ni_dep, sum(ne_asig) ne_dep,
       sum(q1_md) suma1, sum(q2_md) suma2, sum(q3_md) suma3,
       sum(q4_md) suma4, sum(q5_md) suma5,
       sum(q6_md) suma6, sum(q7_md) suma7, sum(q8_md) suma8,
       sum(q9_md) suma9, sum(q10_md) suma10,
       sum(q11_md) suma11, sum(q12_md) suma12, sum(q13_md) suma13,
       sum(q14_md) suma14, sum(q15_md) suma15,
       sum(q16_md) suma16, sum(q17_md) suma17, sum(q18_md) suma18,
       sum(q19_md) suma19, sum(q20_md) suma20,
       sum(q21_md) suma21, sum(q22_md) suma22, sum(q23_md) suma23,
       sum(q24_md) suma24, sum(q25_md) suma25,
       sum(q26_md) suma26, sum(q27_md) suma27, sum(q28_md) suma28,
       sum(q29_md) suma29
from actualiza_pxp app, asignaturas a
where app.clave = a.clave
and app.semestre = '$semestre'
and a.id_depto = '$id_depto'

```

```

insert into ddp (id_departamento, semestre, ni_dep, ne_dep, q1_ddp,
q2_ddp, q3_ddp, q4_ddp, q5_ddp, q6_ddp, q7_ddp, q8_ddp, q9_ddp,
q10_ddp, q11_ddp, q12_ddp, q13_ddp, q14_ddp, q15_ddp, q16_ddp,
q17_ddp, q18_ddp, q19_ddp,
q20_ddp, q21_ddp, q22_ddp, q23_ddp, q24_ddp, q25_ddp, q26_ddp,
q27_ddp, q28_ddp, q29_ddp)
values ('$id_depto', '$semestre', $ni_dep, $ne_dep, $suma1, $suma2,
$suma3, $suma4, $suma5, $suma6, $suma7, $suma8, $suma9, $suma10,
$suma11, $suma12, $suma13, $suma14, $suma15, $suma16, $suma17,
$suma18, $suma19, $suma20,
$suma21, $suma22, $suma23, $suma24, $suma25, $suma26, $suma27,
$suma28, $suma29)

```

```

select sum(ni_dep) ni_div, sum(ne_dep) ne_div,
       sum(q1_ddp) suma1, sum(q2_ddp) suma2, sum(q3_ddp) suma3,
       sum(q4_ddp) suma4, sum(q5_ddp) suma5,
       sum(q6_ddp) suma6, sum(q7_ddp) suma7, sum(q8_ddp) suma8,
       sum(q9_ddp) suma9, sum(q10_ddp) suma10,

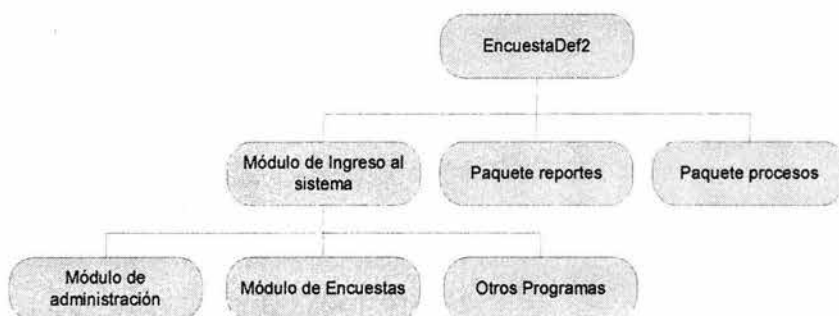
```

```
sum(q11_ddp) suma11, sum(q12_ddp) suma12, sum(q13_ddp) suma13,  
sum(q14_ddp) suma14, sum(q15_ddp) suma15,  
sum(q16_ddp) suma16, sum(q17_ddp) suma17, sum(q18_ddp) suma18,  
sum(q19_ddp) suma19, sum(q20_ddp) suma20,  
sum(q21_ddp) suma21, sum(q22_ddp) suma22, sum(q23_ddp) suma23,  
sum(q24_ddp) suma24, sum(q25_ddp) suma25,  
sum(q26_ddp) suma26, sum(q27_ddp) suma27, sum(q28_ddp) suma28,  
sum(q29_ddp) suma29  
from ddp, deptos d  
where ddp.id_departamento = d.id_depto  
and d.id_division = '$id_division'  
and ddp.semestre = '$semestre'
```

5.7 Diagramas de bloques

Módulos y opciones

A continuación se muestra esquemáticamente el sistema de archivos del sistema. Posteriormente se presenta una descripción de cada uno de los módulos.



Descripción de los módulos

EncuestaDef2

Paquete principal de la aplicación.

Módulo de ingreso al sistema

Contiene programas de validación para distinguir el tipo de usuario del sistema.

Módulo de administración

Contiene programas para el uso exclusivo del administrador, con opciones como altas, asignaciones y generación de reportes.

Módulo de encuestas

Contiene los programas php que constituyen el acceso, y la presentación de la encuesta correspondiente a cada alumno, para su resolución.

Otros programas

Contiene programas que son, en general procesos internos para la generación de los reportes estadísticos, así como programas utilitarios para validaciones y conexión a la base de datos.

Paquete Reportes

Contiene los programas php correspondientes para la generación y presentación de reportes estadísticos.

Paquete procesos

Contiene los programas php que generan los procesos estadísticos que se llevan a cabo cada semestre al finalizar el periodo de encuestas.

Código Fuente

Se considera conveniente omitir el código fuente dentro del manual técnico, para evitar que la tesis sea demasiado extensa.

Relación de Archivos

Paquete EncuestaDef2

Módulo de ingreso al sistema

| | |
|----------------|---|
| ingreso.php | Página de ingreso al sistema. |
| ingresoPre.php | Resultado de la página anterior, si es correcto, redirecciona al módulo adecuado, caso contrario, muestra mensaje de error. |

Módulo de administración

| | |
|------------------|----------------------------|
| ingresoAdmin.php | Menú de ingreso al módulo. |
|------------------|----------------------------|

| | |
|----------------------------------|---|
| admini.php | Alta de profesor. |
| adminiPre.php | Presentación preliminar de Alta de profesor. |
| adminiPro.php | Pantalla de resultado de Alta de profesor. |
| admini2.php | Alta de asignaturas que imparte el profesor. |
| admini2Pre.php | Presentación preliminar de alta de asignaturas que imparte el profesor. |
| admini2pro.php | Pantalla de resultado de alta de asignaturas que imparte el profesor. |
| ad_alum.php | Alta de alumnos. |
| ad_alumPre.php | Presentación preliminar de alta de alumnos. |
| ad_alumPro.php | Pantalla de resultado de alta de alumnos. |
| ad_alum2.php | Alta de asignaturas para alumnos. |
| ad_alum2Pre.php para alumnos. | Presentación preliminar de alta de asignaturas para alumnos. |
| ad_alumPro2.php alumnos. | Pantalla de resultado de alta de asignaturas para alumnos. |
| reporte-intro.php | Pantalla de ingreso para ver reporte de muestra. |

Módulo de encuestas

| | |
|-----------------------------|--|
| enc_intro1.php encuesta. | Pantalla de selección de materia a realizar la encuesta. |
| enc_intro0pro.php | Pantalla intermedia para iniciar encuesta. |

| | |
|-------------------|---|
| encuestaTotal.php | Pantalla de captura de encuesta. |
| enc_intro_pro.php | Pantalla de resultado al contestar la encuesta. |

Otros programas

| | |
|------------------|---|
| conec.php | Componente para la conexión a la base de datos. |
| validaciones.php | Componente para validaciones de datos. |

Paquete reportes

| | |
|-----------------|--|
| reporte-ppr.php | Reporte de profesor por pregunta - respuesta. |
| reporte-pxp.php | Reporte de asignatura - profesor por pregunta. |
| reporte-pxc.php | Reporte de asignatura - profesor por concepto . |
| reporte-ddp.php | Reporte de división - departamento por pregunta. |
| reporte-ddc.php | Reporte de división - departamento por concepto. |
| reporte_tot.php | Reporte general del profesor. |

Paquete procesos.

| | |
|----------------------|--|
| procesa_encuesta.php | Pantalla inicial para inicio de procesos estadísticos. |
| procesos_todos.php | Ejecuta todos los procesos estadísticos. |
| proceso01.php | Programa para procesos estadísticos por grupo. |

| | |
|---------------|---|
| proceso02.php | Programa para procesos estadísticos por asignatura. |
| proceso03.php | Programa para procesos estadísticos por departamento. |
| proceso04.php | Programa para procesos estadísticos por división. |

Sistema de seguridad

Dentro del sistema se encuentran dos principales sistemas de seguridad:

Sistema de validación. Que mediante el uso de usuarios y contraseñas, se controla el acceso para los diferentes tipos de usuario del sistema que son:

Alumnos.

Profesores.

Jefes de departamento y división.

Administrador.

Sistema de sesiones HTTP. Que controla que la sesión sea válida dentro de la aplicación Web, y evitar que se realicen procesos no autorizados, o bien, que se dupliquen procesos.

Además de la integridad referencial que provee la base de datos, impide la duplicidad de información, y facilita la validación de los procesos de la aplicación

Elementos de programación

Documentación de programas.

ingresoPre.php

| Variables | Funciones | Descripción |
|--------------|-----------------------------|---|
| \$link | Conectarse() | Resultado de la página de ingreso, si es correcto, redirecciona al módulo adecuado, caso contrario, muestra mensaje de error. |
| \$login | mssql_query(\$query,\$link) | |
| \$pass | mssql_fetch_array(\$result) | |
| \$query | \$row[] | |
| \$result | \$_POST[] | |
| \$privilegio | \$_SESSION[] | |
| | mssql_close(\$link) | |

adminiPre.php

| Variables | Funciones | Descripción |
|-------------|-------------|--|
| \$id_prof | formato() | Presentación preliminar de Alta de profesor. |
| \$nombre | es_nombre() | |
| \$ap_pat | \$_POST[] | |
| \$ap_mat | | |
| \$grado | | |
| \$division | | |
| \$hay_error | | |
| | | |

adminiPro.php

| Variables | Funciones | Descripción |
|------------|-----------------------------|--|
| \$link | Conectarse() | Pantalla de resultado de Alta de profesor. |
| \$query | mssql_query(\$query,\$link) | |
| \$result | mssql_fetch_array(\$result) | |
| \$id_prof | \$row[] | |
| \$nombre | mssql_close(\$link) | |
| \$ap_pat | \$_POST[] | |
| \$ap_mat | | |
| \$grado | | |
| \$division | | |
| | | |

admini2Pre.php

| Variables | Funciones | Descripción |
|--|--|---|
| \$id_prof \$clave \$gpo \$semestre \$tipo \$nump \$hay_error | \$_POST[] formato() es_grupo() es_semestre() | Presentación preliminar de alta de asignaturas que imparte el profesor. |

admini2pro.php

| Variables | Funciones | Descripción |
|---|---|---|
| \$link \$query \$result \$clave \$gpo \$semestre \$tipo \$nump | Conectarse() mssql_query(\$query,\$link) mssql_fetch_array(\$result) \$row[] mssql_close(\$link) \$_POST[] | Pantalla de resultado de alta de asignaturas que imparte el profesor. |

ad_alumPre.php

| Variables | Funciones | Descripción |
|--|--|---|
| \$num_cta \$fecha_nac \$nom_alu \$pat_alu \$mat_alu \$hay_error | \$_POST[] formato() es_num_cta() es_fecha() es_nombre() | Presentación preliminar de alta de alumnos. |

ad_alumPro.php

| Variables | Funciones | Descripción |
|-------------------|---|----------------------------------|
| \$link \$query | Conectarse() mssql_query(\$query,\$link) | Pantalla de resultado de alta de |

| | | |
|---|--|----------|
| \$result1 \$result2 \$num_cta \$fecha_nac \$nom_alu \$pat_alu \$mat_alu \$num_cta_base | mssql_fetch_array(\$result) \$row[] mssql_close(\$link) \$_POST[] | alumnos. |
|---|--|----------|

ad_alum2Pre.php

| Variables | Funciones | Descripción |
|---|--|--|
| \$num_cta \$sig_alu \$grupo_alu \$sem_alu \$hay_error | \$_POST[] formato() es_num_cta() es_grupo() es_semestre() | Presentación preliminar de alta de asignaturas para alumnos. |

ad_alumPro2.php

| Variables | Funciones | Descripción |
|---|---|--|
| \$link \$query \$result \$result2 \$result3 \$result4 \$num_cta \$sig_alu \$grupo_alu \$sem_alu \$nom_alu \$pat_alu \$mat_alu | Conectarse() mssql_query(\$query,\$link) mssql_fetch_array(\$result) \$row[] mssql_close(\$link) \$_POST[] | Pantalla de resultado de alta de asignaturas para alumnos. |

enc_intro1.php

| Variables | Funciones | Descripción |
|---|--|--|
| \$link \$query1 \$result1 \$sem \$num_cta | Conectarse() mssql_query(\$query,\$link) mssql_fetch_array(\$result) \$row1[] mssql_close(\$link) \$_SESSION [] | Pantalla de selección de materia a realizar la encuesta. |

enc_intro0pro.php

| Variables | Funciones | Descripción |
|--|---|--|
| \$link \$query \$query2 \$result \$result2 \$num_cta \$clave \$grupo \$semestre \$dato \$dato2 \$nom_completo | Conectarse() mssql_query(\$query,\$link) mssql_fetch_array(\$result) \$row[] mssql_close(\$link) \$_POST[] \$_SESSION [] | Pantalla intermedia para iniciar encuesta. |

encuestaTotal.php

| Variables | Funciones | Descripción |
|---|---|----------------------------------|
| \$num_cta \$clave \$grupo \$semestre | \$_POST[] \$_SESSION [] session_is_registered() | Pantalla de captura de encuesta. |

enc_intro_pro.php

| Variables | Funciones | Descripción |
|-------------------|------------------------------|---------------------------------------|
| \$link \$query | \$_POST[] \$_SESSION [] | Pantalla de resultado al contestar la |

| | | |
|------------|-----------------------------|-----------|
| \$result | Conectarse() | encuesta. |
| \$num_cta | mssql_query(\$query,\$link) | |
| \$clave | mssql_fetch_array(\$result) | |
| \$grupo | \$row[] | |
| \$semestre | mssql_close(\$link) | |
| \$comenta | session_is_registered() | |
| \$falta | isset() | |
| \$q1 | | |
| \$q2 | | |
| \$q3 | | |
| ... | | |
| \$q29 | | |

conec.php

| Variables | Funciones | Descripción |
|-----------|---|---|
| \$link | mssql_connect() mssql_select_db() Conectarse() mssql_close() | Componente para la conexión a la base de datos. |

validaciones.php

| Variables | Funciones | Descripción |
|---------------------|---------------|--|
| \$anio | bisiestro() | Componente para validaciones de datos. |
| \$cadena | es_fecha() | |
| \$cadena2 | ereg() | |
| \$expresion_regular | es_nombre() | |
| \$coincidencias | es_num_cta() | |
| | es_semestre() | |
| | formato() | |
| | es_grupo() | |
| | chop() | |
| | strtolower() | |
| | ucwords() | |

reporte-ppr.php

| Variables | Funciones | Descripción |
|---------------|-----------------------------|---|
| \$link | \$_POST[] | Reporte de profesor por pregunta - respuesta. |
| \$query | Conectarse() | |
| \$result | mssql_query(\$query,\$link) | |
| \$clave | mssql_fetch_array(\$result) | |
| \$grupo | \$row[] | |
| \$sem | mssql_close(\$link) | |
| \$nom_prof | number_format() | |
| \$div | | |
| \$dep | | |
| \$asig | | |
| \$ni | | |
| \$ne | | |
| \$e | | |
| \$peso1 | | |
| \$peso2 | | |
| \$peso3 | | |
| \$peso4 | | |
| \$ne_p1_1 | | |
| \$total_p1_1 | | |
| \$ne_p1_2 | | |
| \$total_p1_2 | | |
| \$ne_p1_3 | | |
| \$total_p1_3 | | |
| \$ne_p1_4 | | |
| \$total_p1_4 | | |
| \$max_p1 | | |
| \$suma_p1 | | |
| \$total1 | | |
| ... | | |
| \$ne_p29_1 | | |
| \$total_p29_1 | | |
| \$ne_p29_2 | | |
| \$total_p29_2 | | |
| \$ne_p29_3 | | |
| \$total_p29_3 | | |

| | | |
|--|--|--|
| \$ne_p29_4 \$total_p29_4 \$max_p29 \$suma_p29 \$total29 \$pungrupo_1 ... \$pungrupo_6 \$maxgrupo_1 ... \$maxgrupo_6 \$calif_1 ... \$calif_6 \$calif_prof \$calif_alum | | |
|--|--|--|

reporte-pxc.php

| Variables | Funciones | Descripción |
|-----------|-----------------------------|----------------------------------|
| \$link | \$_POST[] | Reporte de profesor por concepto |
| \$query | Conectarse() | |
| \$result | mssql_query(\$query,\$link) | |
| \$clave | mssql_fetch_array(\$result) | |
| \$ni | \$row[] | |
| \$ne | mssql_close(\$link) | |
| \$e | number_format() | |
| \$fecha | | |
| \$div | | |
| \$dep | | |
| \$sem | | |

reporte-pxp.php

| Variables | Funciones | Descripción |
|-----------|-----------------------------|----------------------------------|
| \$link | \$_POST[] | Reporte de profesor por pregunta |
| \$query | Conectarse() | |
| \$result | mssql_query(\$query,\$link) | |
| \$slave | mssql_fetch_array(\$result) | |
| \$ni | \$row[] | |
| \$ne | mssql_close(\$link) | |
| \$e | number_format() | |
| \$fecha | | |
| \$div | | |
| \$dep | | |
| \$sem | | |

reporte-ddc.php

| Variables | Funciones | Descripción |
|-----------|-----------------------------|----------------------------------|
| \$link | \$_POST[] | Reporte de división por concepto |
| \$query | Conectarse() | |
| \$result | mssql_query(\$query,\$link) | |
| \$slave | mssql_fetch_array(\$result) | |
| \$ni | \$row[] | |
| \$ne | mssql_close(\$link) | |
| \$e | number_format() | |
| \$fecha | | |
| \$div | | |
| \$nomdiv | | |
| \$sem | | |

reporte-ddp.php

| Variables | Funciones | Descripción |
|-----------|-----------------------------|----------------------------------|
| \$link | \$_POST[] | Reporte de división por pregunta |
| \$query | Conectarse() | |
| \$result | mssql_query(\$query,\$link) | |
| \$slave | mssql_fetch_array(\$result) | |
| \$ni | \$row[] | |
| \$ne | mssql_close(\$link) | |
| \$e | number_format() | |
| \$fecha | | |

| | | |
|----------|--|--|
| \$div | | |
| \$nomdiv | | |
| \$sem | | |

proceso01.php

| Variables | Funciones | Descripción |
|---------------|-----------------------------|--|
| \$link | Conectarse() | Programa para procesos estadísticos por grupo. |
| \$query | mssql_query(\$query,\$link) | |
| \$result | mssql_fetch_array(\$result) | |
| \$row[] | mssql_close(\$link) | |
| \$query1 | number_format() | |
| \$result1 | | |
| \$clave | | |
| \$grupo | | |
| \$sem | | |
| \$ni | | |
| \$ne | | |
| \$e | | |
| \$peso1 | | |
| \$peso2 | | |
| \$peso3 | | |
| \$peso4 | | |
| \$ne_p1_1 | | |
| \$total_p1_1 | | |
| \$ne_p1_2 | | |
| \$total_p1_2 | | |
| \$ne_p1_3 | | |
| \$total_p1_3 | | |
| \$ne_p1_4 | | |
| \$total_p1_4 | | |
| \$max_p1 | | |
| \$suma_p1 | | |
| \$total1 | | |
| ... | | |
| \$ne_p29_1 | | |
| \$total_p29_1 | | |
| \$ne_p29_2 | | |

| | | |
|--|--|--|
| \$total_p29_2 \$ne_p29_3 \$total_p29_3 \$ne_p29_4 \$total_p29_4 \$max_p29 \$suma_p29 \$total29 \$pungrupo_1 ... \$pungrupo_6 \$maxgrupo_1 ... \$maxgrupo_6 \$calif_1 ... \$calif_6 \$calif_prof \$calif_alum | | |
|--|--|--|

proceso02.php

| Variables | Funciones | Descripción |
|---|---|---|
| \$link \$query \$result \$row[] \$row1[] \$query1 \$result1 \$semestre \$slave \$ni_asig \$ne_asig \$suma1 | Conectarse() mssql_query(\$query,\$link) mssql_fetch_array(\$result) mssql_close(\$link) | Programa para procesos estadísticos por asignatura. |

| | | |
|----------|--|--|
| ... | | |
| \$suma29 | | |

proceso03.php

| Variables | Funciones | Descripción |
|---|---|---|
| \$link \$query \$result \$row[] \$row1[] \$query1 \$result1 \$semestre \$id_depto \$ni_dep \$ne_dep \$suma1 ... \$suma29 | Conectarse() mssql_query(\$query,\$link) mssql_fetch_array(\$result) mssql_close(\$link) | Programa para procesos estadísticos por departamento. |

proceso04.php

| Variables | Funciones | Descripción |
|--|---|---|
| \$link \$query \$result \$row[] \$row1[] \$query1 \$result1 \$semestre \$id_division \$ni_div \$ne_div \$suma1 ... \$suma29 | Conectarse() mssql_query(\$query,\$link) mssql_fetch_array(\$result) mssql_close(\$link) | Programa para procesos estadísticos por división. |

CONCLUSIONES

Conclusiones

El diseño y desarrollo del sistema de información con carácter de programa piloto para las encuestas del denominado “Programa de evaluación del proceso enseñanza – aprendizaje de la Facultad de Ingeniería”, se consiguió de manera satisfactoria utilizando tecnología Web, mediante una aplicación hecha en el lenguaje PHP soportada sobre el servidor Web IIS (Internet Information Service) y la base de datos SQL Server 2000.

Unas de las principales contribuciones de éste trabajo, fue la creación de una aplicación Web que hace posible contar con mecanismos o sistemas de evaluación, que permitan determinar el grado de cumplimiento de sus objetivos que fueron fijados en el proceso enseñanza - aprendizaje en la Facultad de Ingeniería, con el fin de promover el mejoramiento continuo, que incluya los resultados de las evaluaciones de los alumnos hacia los docentes, así como del plan de estudios y la manera en que este se aplica.

El sistema permite una toma de decisiones más pronta ya que los resultados, una vez concluido el periodo de levantamiento de encuestas, pueden obtenerse en línea y mediante un acceso remoto gracias a Internet y tampoco es necesario esperar a que la interpretación de los datos se lleve a cabo ya que el proceso se puede ejecutar prácticamente de manera inmediata.

La aplicación ofrece a los distintos tipos de usuarios, las prestaciones requeridas para cada uno de ellos, en un ambiente que facilita cada proceso y ofrece una interfaz más conveniente y eficiente que pueda sustituir, en su momento, al procedimiento con el que actualmente se realizan las encuestas.

La base de datos diseñada e implantada, utilizando el manejador de base de datos MS – SQL Server 2000, cumple con las diversas necesidades del sistema, como son las de almacenar los datos de los alumnos, profesores y grupos correspondientes a cada asignatura, así como permitir generar y controlar toda la información estadística referente a las encuestas. Brindando también la posibilidad de mantener históricos de los ejercicios estadísticos semestrales para su futura comparación.

Asimismo, la base de datos cumple con las condiciones necesarias de integridad referencial que garantizan la confiabilidad y seguridad en la información almacenada para las encuestas.

Se tuvo especial cuidado en lo referente a la asegurar y mantener la confidencialidad de las respuestas emitidas por parte de los alumnos, así como la integridad de dichos datos y que estas respuestas al final del proceso, de forma que no puedan ser asociadas con cada uno de los alumnos que fueron encuestados. Ya que una vez realizada la encuesta, no existe correlación entre las respuestas y el usuario que las realizó, sin embargo, existen banderas que se habilitan para saber si cada uno de los alumnos ya ha contestado alguna encuesta específica.

Las posibilidades que brinda una aplicación Web dinámica nos permitieron personalizar los reportes estadísticos a cada tipo de los destinatarios finales, ya sean estos profesores, jefes de departamento, jefes de división, dependiendo de los requerimientos y filtración de los datos requerida, basándonos en los reportes que genera el sistema actual.

La estructura del Sistema de Información creado, permite que se puedan generar en un futuro, otros reportes según sean requeridos, considerando también históricos o comparativos con otros semestres.

Con esta versión electrónica de la encuesta se aspira obtener resultados de toda la población encuestada y no solo de un subconjunto o muestra como se hace actualmente, esto debido a que no todos los alumnos se encuentran al momento de realizar dichas encuestas. La visión es convertir a la encuesta vía Web en un censo, con la capacidad de recopilar información sobre toda la población, abriendo una brecha para otros ejercicios estadísticos de utilidad para la Facultad.

Permite una mejor eficacia en los procesos estadísticos de la Facultad ya que, entre otras cosas, ayuda a disminuir el costo de manejo de recursos tanto materiales como humanos. Ahorro de recursos materiales en el sentido de que por ser un sistema totalmente electrónico no se requiere de papelería, sistemas de impresión y análisis de los datos. Así mismo ahorro en recursos humanos ya que la distribución y levantamiento de encuestas ya no sería necesaria.

Por otra parte se vuelve más eficiente la notificación de resultados y reportes ya que todo se puede hacer vía electrónica mediante la consulta del sitio previa identificación mediante una validación de contraseñas.

En resumen, la información requerida se presenta al usuario de manera completa y analizada, incluyendo páginas de confirmación, páginas de notificación, avisos de errores.

El sistema resultó ser de uso y consulta rápida, ya que la interfaz con el usuario es accesible por su diseño amigable y de sencillo entendimiento, como consecuencia es eficiente y práctica.

Asimismo la aplicación cuenta con proceso de interacción administrativa con el usuario, que se puede acceder vía remota previa identificación mediante contraseña y permite dar de alta profesores, grupos, y alumnos por semestre.

Utilización inteligente de la tecnología en el sentido que hecha mano de entornos abiertos (como el lenguaje PHP por lo que no hay que pagar el uso de licencias), herramientas probadas, tecnología estándar, compatibilidad continua ya que es soportada por la mayoría de los navegadores actuales de Internet, sencillez para realizar modificaciones y actualizaciones al sistema.

Además de brindar un importante ahorro administrativo efectivo mediante ahorro de recursos de la Facultad, el cual puede eventualmente llegar a ser extensivo a la mayoría de las instancias educativas de nuestra Universidad.

Finalmente se puede concluir que el sistema cumple con las reglas de aseguramiento de calidad del software como lo son:

Corrección: Ya que se pudieron realizar de manera satisfactoria las especificaciones establecidas por los clientes.

Fiabilidad: Ya que se garantiza la integridad de los resultados recabados en el sistema.

Eficiencia: Debido a que los requerimientos en cuanto a recursos de computadora, de código y ancho de banda para realizar a cabo sus funciones son mínimos.

Seguridad: Tiene alto grado de control al acceso del sistema y de los datos por personal no autorizado.

Facilidad de uso: Por que el esfuerzo requerido para aprender, trabajar, preparar la entrada e interpretar la salida del sistema es mínimo, ya que las características de uso de las aplicaciones Web son intuitivas y reconocidas por casi cualquier usuario.

Facilidad de mantenimiento: La aplicación requiere poco esfuerzo para la administración, mantenimiento y control de los datos.

Facilidad de prueba: Es de gran facilidad la realización de pruebas en el sistema con la finalidad de asegurar que se realicen las funciones requeridas.



Portabilidad. El sistema es de gran portabilidad en cuanto a código ya que está basado en un lenguaje multiplataformas en lo que se refiere al servidor que es donde reside el código.

Reusabilidad. La infraestructura del sistema permite incorporar nueva funcionalidad respecto a los requerimientos necesarios hacia su implantación en toda la facultad.



GLOSARIO

Terminos usados



A

Active Server Page - ASP (Página de Servidor Activo) Una página ASP es un tipo especial de página HTML que contiene unos pequeños programas (también llamados scripts) que son ejecutados en servidores Microsoft Internet Information Server antes de ser enviados al usuario para su visualización en forma de página HTML. Habitualmente esos programas realizan consultas a bases de datos y los resultados de esas consultas determinan la información que se envía a cada usuario específico. Los ficheros de este tipo llevan el sufijo .asp. No confundir con Application Service Provider

ActiveX (ActiveX) Lenguaje desarrollado por Microsoft para la elaboración de aplicaciones exportables a la red y capaces de operar sobre cualquier plataforma a través, normalmente, de navegadores WWW. Permite dar dinamismo a las páginas web.

American Standard Code for Information Interchange - ASCII (Estándar Americano de Codificación para el Intercambio de Información) Conjunto de normas de codificación de caracteres mediante caracteres numéricos, de amplia utilización en informática y telecomunicaciones.

Apache (Apache) Servidor HTTP de dominio público basado en el sistema operativo Linux. Apache fue desarrollado en 1995 y es actualmente uno de los servidores HTTP más utilizados en la red.

application (aplicación) Un programa informático que lleva a cabo una función con el objeto de ayudar a un usuario a realizar una determinada actividad. WWW, FTP, correo electrónico y Telnet son ejemplos de aplicaciones en el ámbito de Internet

Application Program Interface - API (Interfaz para programas de aplicación) Conjunto de convenciones de programación que definen cómo se invoca un servicio desde un programa.

Asymmetrical Digital Subscriber Line - ADSL (Línea de Suscripción Asimétrica Digital) Tecnología de transmisión de tipo xDSL, que permite a los hilos telefónicos de cobre convencionales una alta velocidad de transmisión. Es la más utilizada actualmente y se denomina asimétrica porque permite más velocidad en la recepción de datos por parte del usuario que en la emisión de datos por éste.

Asynchronous Transfer Mode - ATM (Modo de Transferencia Asíncrona)

Estándar que define la conmutación de paquetes (*cells*- celdas o células) de tamaño fijo con alta carga, alta velocidad (entre 1,544 Mbps. y 1,2 Gbps) y asignación dinámica de ancho de banda. ATM es conocido también como paquete rápido (*fast packet*).

bandwidth (ancho de banda,anchura de banda) Técnicamente es la diferencia en hertzios (Hz) entre la frecuencia más alta y la más baja de un canal de transmisión. Sin embargo, este término se usa mucho más a menudo para definir la cantidad de datos que puede ser enviada en un periodo de tiempo determinado a través de un circuito de comunicación dado, por ejemplo, 33,6 Kbps (miles de bits por segundo).

baud (baudio) Cuando se transmiten datos, un baudio es el numero de veces que cambia el 'estado' del medio de transmisión en un segundo. Como cada cambio de estado puede afectar a más de un bit de datos, la tasa de bits de datos transferidos (por ejemplo, medida en bits por segundo) puede ser superior a la correspondiente tasa de baudios.

bit (bitio,bit) Unidad mínima de información digital que puede ser tratada por una computadora. Proviene de la contracción de la expresión *binary digit*(dígito binario).

bits per second - bps (bits por segundo) Unidad de medida de la capacidad de transmisión de una línea de telecomunicación.

byte (octeto,byte) Conjunto significativo de ocho bits que representan un carácter, por ejemplo la letra "a", en un sistema informático.

C

cache (caché,copia de visitas) En Internet es la copia mantenida por un navegador de las páginas web visitadas últimamente, de manera que, si el usuario vuelve a solicitarlas, son leídas desde el disco duro sin necesidad de tener que conectarse de nuevo a la red, consiguiéndose así una mejora muy apreciable del tiempo de respuesta.

Cascade Style Sheet - CSS (Hoja de Estilo en Cascada) Es un conjunto de instrucciones HTML que definen la apariencia de uno o más elementos de un conjunto de páginas web con el objetivo de uniformizar su diseño.

cgi-bin (cgi-bin) Directorio de un servidor web donde suelen almacenarse los programas CGI. *bin* es una contracción de *binario*.

client (cliente) Un sistema o proceso que solicita a otro sistema o proceso que le preste un servicio. Una estación de trabajo que solicita el contenido de un fichero a un servidor de ficheros es un cliente de este servidor.

client-server model (modelo cliente-servidor) Modelo de comunicación entre computadoras conectadas a una red en el cual hay uno, llamado cliente, que satisface las peticiones realizadas por otro llamado servidor.

Common Gateway Interface - CGI (Interfaz Común de Pasarela) Interfaz de intercambio de datos estándar en WWW a través del cual se organiza el envío y recepción de datos entre navegador y programas residentes en servidores WWW.

cookie (espía,cookie,cuqui ,figón) Conjunto de caracteres que se almacenan en el disco duro o en la memoria temporal de la computadora de un usuario cuando accede a las páginas de determinados sitios web. Se utilizan para que el servidor accedido pueda conocer las preferencias del usuario al volver éste a conectarse. Dado que pueden ser un peligro para la intimidad de los usuarios, éstos deben saber que los navegadores permiten desactivar los (¿o las?) caquis.

Copyright (derecho de copia) Derecho que tiene un autor, incluido el autor de un programa informático, sobre todas y cada una de sus obras y que le permite decidir en qué condiciones han de ser éstas reproducidas y distribuidas. Aunque este derecho es legalmente irrenunciable puede ser ejercido de forma tan restrictiva o tan generosa como el autor decida. El símbolo de este derecho es ©.

D

domain (dominio) Conjunto de caracteres que identifica un sitio de la red accesible por un usuario. Así, por ejemplo, el nombre de dominio .es identifica a los usuarios dados de alta en el registro español de nombres de dominio.

Domain Name System - DNS (Sistema de Nombres de Dominio) El DNS un servicio de búsqueda de datos de uso general, distribuido y multiplicado. Su utilidad principal es la búsqueda de direcciones IP de sistemas anfitriones (*hosts*) de Internet basándose en los nombres de éstos. El estilo de los nombres de host utilizado actualmente en Internet es llamado nombre de dominio. Los dominios originarios, a los que se ha añadieron algunos más en el año 2000, son: .com (comercial, empresas), .edu (educación, centros docentes), .org (organización sin ánimo de lucro), .net (operación de la red), gov (gobierno o administración pública) y .mil (ejército de los EE.UU.). La mayoría de los países tienen un dominio propio. Por ejemplo, .mx (México), .es (España), .au (Australia).

dotcom (punto.com) Término que se aplica a las empresas nacidas en diferentes sectores productivos como consecuencia del desarrollo de Internet. El nombre viene del dominio .com que estas empresas utilizan en sus sitios web.

dynamic HTML o DHTML (HTML dinámico) Extensiones del lenguaje HTML que permiten crear páginas web más animadas y expresivas.

E

eXtensible Markup Language - XML (Lenguaje Extensible de Marcado) Lenguaje desarrollado por el W3 Consortium para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red. El formato de XML es muy parecido al del HTML aunque no es una extensión ni un componente de éste.

F

File Transfer Protocol - FTP (Protocolo de Transferencia de Ficheros) Protocolo que permite a un usuario de un sistema acceder a, y transferir desde, otro sistema de una red. FTP es también habitualmente el nombre del programa que el usuario invoca para ejecutar el protocolo.

firewall (cortafuegos) Dispositivo que se coloca entre una red local e Internet y cuyo objetivo es asegurar que todas las comunicaciones entre los usuarios de dicha red e Internet se realicen conforme a las normas de seguridad de la organización que lo instala.

Free Software (software libre) Programas desarrollados y distribuidos según la filosofía de dar al usuario la libertad de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar dichos programa (Linux es un ejemplo de esta filosofía). El software libre no es siempre software gratuito (equivocación bastante habitual que tiene su origen en que la palabra inglesa *free* significa tanto "libre" como "gratuito").

Free Software Foundation - FSF (Fundación para el Software Libre) Fundación norteamericana creada en 1996 por Richard M. Stallman cuyo objetivo es promover el desarrollo y el uso de software libre en todas las áreas de la Informática.

freeware (programas de dominio público, programas de libre distribución, programas gratuitos) Programas informáticos que se distribuyen a través de la red de forma gratuita.

G

gateway Punto de una red que actúa como punto de entrada a otra red.

General Public License - GPL (Licencia Pública General) Licencia de regulación de los derechos de autor de los programas de software libre (*free software*). Esta licencia, promovida por la Free Software Foundation (FSF) en el marco de la iniciativa GNU, permite la distribución de copias de programas (e incluso cobrar por ello), así como modificar el código fuente de los mismos o utilizarlo en otros programas.

Gigabyte - GB (*gigabyte, gigaocteto*) Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, DVD, etc.). Un GB corresponde a 1.024 millones de bytes.

Graphical User Interface - GUI (Interfaz Gráfica de Usuario) Componente de una aplicación informática que el usuario visualiza y a través de la cual opera con ella. Está formada por ventanas, botones, menús e iconos, entre otros elementos.

Graphics Interchange Format - GIF (Formato de Intercambio Gráfico) Formato gráfico utilizado para representar ficheros en la red, creado y patentado por la empresa norteamericana Unisys. Los gráficos GIF, que pueden ser incluidos en páginas HTML, son reconocidos y visualizados por los programas navegadores. Hay un tipo especial de GIFs llamados *Animated GIF* o GIF animados.

H

host system (*sistema anfitrión, sistema principal*) Computadora que, mediante la utilización de los protocolos TCP/IP, permite a los usuarios comunicarse con otros sistemas anfitriones de una red. Los usuarios se comunican utilizando programas de aplicación, tales como el correo electrónico, Telnet, WWW y FTP. La acepción verbal (*to host*) describe el hecho de almacenar algún tipo de información en un servidor ajeno.

host address (*dirección de sistema anfitrión*) Es la dirección Internet de un sistema anfitrión. Puede ser un nombre o una serie de números.

host name (*nombre de sistema anfitrión*) Nombre dado a un sistema anfitrión. Por ejemplo ati.es.

host number (número de sistema anfitrión) Identificación numérica dada a una máquina anfitriona. Por ejemplo 132.248.59.84

hyperlink (hipervínculo, nexa, hiperenlace) Puntero existente en un documento hipertexto que apunta a (enlaza con) otro documento que puede ser o no otro documento hipertexto.

hypermedia (hipermedia) Acrónimo de los términos "hipertexto" y "multimedia", que se refiere a las páginas web que integran información en distintos tipos de formatos: texto, gráficos, sonidos y vídeo, principalmente.

hypertext (hipertexto) Aunque el concepto en sí es muy anterior al WWW (fue creado por el físico norteamericano Vannevar Bush en 1945), en Internet el término se aplica a los enlaces existentes en las páginas escritas en HTML, enlaces que llevan a otras páginas que pueden ser a su vez páginas de hipertexto. Las páginas hipertextuales son accedidas normalmente a través de navegadores WWW.

HyperText Markup Language - HTML (Lenguaje de Marcado de Hipertexto) Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores WWW. Admite componentes hipertextuales y multimedia.

HyperText Transfer Protocol - HTTP (Protocolo de Transferencia de Hipertexto) Protocolo usado para la transferencia de documentos WWW.

I

information (información) Agregación de datos que tiene un significado específico más allá de cada uno de éstos. Un ejemplo: 2, 0, 0 y 3 son datos; 2003 es una información. La información ha sido siempre un recurso muy valioso, revalorizado hoy más aún por el desarrollo y la expansión de las Tecnologías de la Información y de las comunicaciones.

Information Technologies - IT (Tecnologías de la Informació) Una forma de denominar al conjunto de herramientas, habitualmente de naturaleza electrónica, utilizadas para la recogida, almacenamiento, tratamiento, difusión y transmisión de la información.

Integrated Services Digital Network - ISDN (RDSI, Red Digital de Servicios Integrados) Tecnología de transmisión que combina servicios de voz y digitales a través de la red en un solo medio, haciendo posible ofrecer a los clientes servicios digitales de datos así como conexiones de voz a través de un solo 'cable', con una

velocidad máxima de transmisión de 128 Kbps. Los estándares de la ISDN los especifica la ITU-TSS.

interface (interfase, interfaz) Zona de contacto, conexión entre dos componentes de "hardware", entre dos aplicaciones o entre un usuario y una aplicación. También apariencia externa de una aplicación informática.

International Organization for Standardization - ISO (Organización Internacional para la Normalización) Organización de carácter voluntario fundada en 1946 que es responsable de la creación de estándares internacionales en muchas áreas, incluyendo la informática y las comunicaciones. Está formada por las organizaciones de normalización de sus países miembro.

internet Una internet (con "i" minúscula) es un conjunto de redes conectadas entre sí. Red de telecomunicaciones nacida en 1969 en los EE.UU. a la cual están conectadas centenares de millones de personas, organismos y empresas en todo el mundo, mayoritariamente en los países más desarrollados, y cuyo rápido desarrollo está teniendo importantes efectos sociales, económicos y culturales, convirtiéndose de esta manera en uno de los medios más influyentes de la llamada Sociedad de la Información y en la Autopista de la Información por excelencia. Fue conocida como ARPANET hasta 1974.

Internet Protocol - IP (Protocolo Internet) Conjunto de reglas que regulan la transmisión de paquetes de datos a través de Internet. La versión actual es IPv4 mientras que en el proyecto Internet2 se intenta implementar la versión 6 (IPv6), que permitiría mejores prestaciones dentro del concepto QoS (*Quality of Service*).

IP address (dirección IP) Número compuesto por 32 dígitos binarios que identifica a todo emisor o receptor de información en Internet.

J

Java (Java) Lenguaje de programación desarrollado por la empresa Sun para la elaboración de pequeñas aplicaciones exportables a la red (*applets*) y capaces de operar sobre cualquier plataforma a través, normalmente, de navegadores WWW. Permite dar dinamismo a las páginas web.

Java Server Page - JSP (Página de Servidor Java) Una página JSP es un tipo especial de página HTML que contiene unos pequeños programas (también llamados *scripts*) que son ejecutados en servidores Netscape antes de ser enviados al usuario para su visualización en forma de página HTML. Habitualmente esos

programas realizan consultas a bases de datos y los resultados de esas consultas determinan la información personalizada que se envía a cada usuario específico. Los ficheros de este tipo llevan el sufijo *.jsp*.

JavaScript (JavaScript) Lenguaje de programación para WWW desarrollado por Netscape. Al igual que VBScript, pertenece a la familia Java pero se diferencia de este último en que los programas están incorporados en el fichero HTML.

Joint Photographers Expert Group - JPEG (Grupo Conjunto de Fotógrafos Expertos) Formato gráfico comprimido utilizado para representar ficheros en la red. Los gráficos JPEG, que pueden ser incluidos en páginas HTML, son reconocidos y visualizados por los programas navegadores.

K

Kilobits per second - Kbps (kilobits por segundo) Unidad de medida de la capacidad de transmisión de una línea de telecomunicación. Un Kbps corresponde a 1.000 bits por segundo.

Kilobyte - KB (Kilobyte ,kilocteto) Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, DVD, etc.). Un KB corresponde a 1.024 bytes.

L

link (liga, vínculo, enlace) Apuntadores hipertexto que sirven para saltar de una información a otra, o de un servidor a otro, cuando se navega por Internet o bien la acción de realizar dicho salto.

Local Area Network - LAN (Red de Area Local) Red de datos para dar servicio a un área geográfica máxima de unos pocos kilómetros cuadrados, por lo cual pueden optimizarse los protocolos de señal de la red para llegar a velocidades de transmisión de Gbps (gigabits por segundo).

Megabits per second - Mbps (megabits por segundo) Unidad de medida de la capacidad de transmisión por una línea de telecomunicación. Un Mbps corresponde a un millón de bits por segundo.

Megabyte - MB (megabyte ,megaocteto) Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, DVD, etc.). Un MB corresponde a 1.024.000 bytes.

Multipurpose Internet Mail Extensions - MIME (Extensiones Multipropósito del Correo Internet) Conjunto de especificaciones Internet de libre distribución que permiten tanto el intercambio de texto escrito en lenguajes con diferentes juegos de caracteres como el intercambio de ficheros de diversos formatos entre computadoras y aplicaciones que sigan los estándares de correo Internet. Las especificaciones MIME están recogidas en numerosos RFCs, entre los que se encuentran los RFC1521 y 1848.

O

octet (octeto) Un octeto está formado por 8 unidades de información (llamadas bits). Este término se usa a veces en vez de *byte* en la terminología de redes porque algunos sistemas tienen *bytes* que no están formados por 8 bits.

Open Systems Interconnection - OSI (Interconexión de Sistemas Abiertos) Modelo de referencia diseñado por comités ISO con el objetivo de convertirlos en estándares internacionales de arquitectura de redes de computadoras.

P

password (palabra de paso, contraseña) Conjunto de caracteres alfanuméricos que permite a un usuario el acceso a un determinado recurso o la utilización de un servicio dado

Personal Home Page Tools - PHP (Herramientas para Páginas Iniciales Personales) Lenguaje de programación tipo *script* para entornos Web utilizado sobre todo en servidores para personalizar la información que se envía a los usuarios que acceden a un sitio web. Es un programa de software libre con unas funciones muy semejantes a las de ASP y JSP.

Petabyte - PB (*petabyte* ,*petaocteto*) Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, DVD, etc.). Un PB corresponde a 1.024 billones (2^{50}) de bytes. Todavía no se han desarrollado memorias ni dispositivos de almacenamiento de esta capacidad.

plug in (*pluguín,conector ,plug In*) Pequeño programa que añade alguna función a otro programa, habitualmente de mayor tamaño. Un programa puede tener uno o más conectores. Son muy utilizados en los programas navegadores para ampliar sus funcionalidades.

port (puerta,puerto) En los protocolos TCP/IP es un punto de conexión lógica. También es un punto de conexión física de una computadora para enlazar con otros dispositivos como, por ejemplo, módems o impresoras.

Portable Document Format - PDF (Formato de Documento Portable) Formato gráfico creado por la empresa Adobe que reproduce cualquier tipo de documento en forma digital idéntica, facsímil, permitiendo así la distribución electrónica de los mismos a través de la red en forma de ficheros PDF. El programa gratuito Acrobat Reader, de Adobe, permite la visualización de los mismos

Practical Extraction and Report Language - PERL (Lenguaje Práctico de Extracción e Informes) Lenguaje de programación muy utilizado para la elaboración de aplicaciones CGI.

protocol (protocolo) Descripción formal de formatos de mensaje y de reglas que dos computadoras deben seguir para intercambiar dichos mensajes. Un protocolo puede describir detalles de bajo nivel de las interfaces máquina-a-máquina o intercambios de alto nivel entre programas de asignación de recursos.

proxy (apoderado ,proxy) Servidor especial encargado, entre otras cosas, de centralizar el tráfico entre Internet y una red privada, de forma que evita que cada una de las máquinas de la red interior tenga que disponer necesariamente de una conexión directa a la red. Al mismo tiempo contiene mecanismos de seguridad (*firewall* o cortafuegos) que impiden accesos no autorizados desde el exterior hacia la red privada.

public domain (dominio público) Espacio, información o programa a disposición libre de los usuarios.

R

Request For Comments - RFC (Petición de Comentarios) Serie de documentos iniciada en 1967 que describe el conjunto de protocolos de Internet y experimentos similares. No todos los RFC's (en realidad muy pocos de ellos) describen estándares de Internet pero todos los estándares Internet están escritos en forma de RFC's. La serie de documentos RFC es inusual en cuanto los protocolos que describen son elaborados por la comunidad Internet que desarrolla e investiga, en contraste con los protocolos revisados y estandarizados formalmente que son promovidos por organizaciones como CCITT y ANSI.

RFC 822 Formato estándar Internet para cabeceras de mensajes de correo electrónico. Los expertos hablan a menudo de "mensajes 822". El nombre viene del "RFC 822", que contiene esa especificación (STD 11, RFC 822). El formato 822 era conocido antes como formato 733.

RFC 822 - RFC 822 Formato estándar Internet para cabeceras de mensajes de correo electrónico. Los expertos hablan a menudo de mensajes 822. El nombre viene del RFC 822, que contiene esa especificación (STD 11, RFC 822). El formato 822 era conocido antes como formato 733.

root (raíz) Dícese del directorio inicial de un sistema de ficheros. En entornos Unix se refiere también al usuario principal.

router (encaminador,direccionador,enrutador) Dispositivo que distribuye tráfico entre redes. La decisión sobre a donde enviar los datos se realiza en base a información de nivel de red y tablas de direccionamiento.

S

server (servidor) Sistema que proporciona recursos (por ejemplo, servidores de ficheros, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la Red.

Simple Mail Transfer Protocol - SMTP (Protocolo Simple de Transferencia de Correo) Protocolo definido en STD 10, RFC 821, que se usa para transferir correo electrónico entre computadoras. Es un protocolo de servidor a servidor, de tal manera que para acceder a los mensajes es preciso utilizar otros protocolos.

T

Terabyte - TB (terabyte ,teraocteto) Unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático (disquete, disco duro, CD-ROM, etc.). Un TB corresponde a algo más de mil billones de bytes, concretamente 1,024 (2^{40}). Todavía no se han desarrollado memorias de esta capacidad aunque sí dispositivos de almacenamiento.

top level domain (dominio de alto nivel) En una dirección Internet son los caracteres que figuran en último lugar, precedidos de un punto (.). Por ejemplo, en <http://www.unam.com> el dominio de alto nivel es "es".

Transmission Control Protocol/Internet Protocol - TCP/IP (Protocolo de Control de Transmisión/Protocolo Internet) Sistema de protocolos, definidos en RFC 793, en los que se basa buena parte de Internet. El primero se encarga de dividir la información en paquetes en origen, para luego recomponerla en destino, mientras que el segundo se responsabiliza de dirigirla adecuadamente a través de la red.

U

Uniform Resource Locator/Universal Resource Identifier - URL/URI (Localizador Uniforme de Recursos/Identificador Universal de Recursos) Sistema unificado de identificación de recursos en la red (el URI todavía no está implantado). Las direcciones se componen de protocolo, FQDN y dirección local del documento dentro del servidor. Este tipo de direcciones permite identificar objetos WWW, Gopher, FTP, News,... Ejemplos de URL son: <http://www.ingenieria.unam> o <ftp://ftp.unam.>

W

Web Servidor de información WWW. Se utiliza también para definir el universo WWW en su conjunto. En el primer caso quizás debería ir en minúscula; en el segundo, en mayúscula.

web server (servidor web) Máquina conectada a la red en la que están almacenadas físicamente las páginas que componen un sitio web. Dícese también del programa que sirve dichas páginas.

Web User Interface - WUI (Interfaz de Usuario basada en Web) Interfaz gráfica de usuario con la apariencia típica de una página web

Webmaster, webmaster (administrador de Web) Persona que se encarga de la gestión y mantenimiento de un servidor web, fundamentalmente desde el punto de vista técnico.

Wide Area Information Servers - WAIS (Servidores de Información de Area Amplia) Servicio de información distribuida, anterior al WWW, que permitía hacer preguntas en lenguaje simple, la búsqueda indexada para obtener información con rapidez y un mecanismo de retroalimentación de información relevante para permitir que los resultados de una búsqueda inicial repercutiesen en búsquedas subsiguientes.

Wide Area Network - WAN (Red de Área Amplia) Red de conectadas entre sí en un área geográfica relativamente extensa. Este tipo de redes suelen ser públicas, es decir, compartidas por muchos usuarios.

World Wide Web - WWW o W3 Sistema de información distribuido, basado en hipertexto, creado a principios de los años 90 por Tim Berners-Lee, investigador en el CERN, Suiza. La información puede ser de cualquier formato (texto, gráfico, audio, imagen fija o en movimiento) y es fácilmente accesible a los usuarios mediante los programas navegadores. Es preciso destacar el hecho poco habitual de que tanto Berners-Lee como el CERN renunciaron a la explotación comercial de este extraordinario invento.

Fuentes

"Guía Práctica para Usuarios de HTML. Creación de Páginas Web". Anaya Multimedia, Madrid, 1996.

"Internetworking with TCP/IP: Principles, Protocols and Architectures", Prentice Hall, Englewood Cliffs, NJ, 1995.

Diccionario de la Lengua Española", Vigésima Primera Edición

Free On-Line Dictionary of Computing.

Sección "VOCABULARIO" de Cyberp@ís y sitio web personal.

"The Internet Companion: A Beginner's Guide to Global Networking", Addison-Wesley, Reading, MA, Segunda Edición, 1994.

Sección "Diccionario", Números 1 a 40. Mundo Revistas S.A. (Grupo Godó), Barcelona.



BIBLIOGRAFÍA

Morán Oviedo
La Evaluación en los aprendizajes y sus Implicaciones Sociales
Perfiles Educativos
No. 3
México
1980

Gil Rubio, Francisco Javier, et. al.
Creación de sitios web con PHP 4
Mc Graw Hill
España
2001
547 p.

Welling, Luke
PHP and MySQL Web development
Estados Unidos
Sams
2001
867 p.

Medinets, David
PHP3 programming browser-based applications
McGraw-Hill
Mexico
2000
551 p.

Trigos Garcia, Esteban
PHP 4
España
Anaya Multimedia
2000
284 p.

Talmage, Ron
Microsoft SQL server 7.0 administrator's guide
Estados Unidos
Rocklin
1999
770 p.

Linsenbardt, Mark A.
SQL server 7 administration
Mexico
Osborne
1999
680 p.

Gunderloy, Mike
SQL Server 7
España
Anaya Multimedia
1999
640 p.

Gui-based design and development for client/ Server applicaations : Using
powerbuilder, sqlwindows, visual basic, parts workbench
Jonathan s. sayles [et al.]
New york
J. Wiley
1994
310 p

Gruber, Martin
Understanding sql
Estados Unidos
Sybex
1990
434 p

Schmuller, Joseph
Aprendiendo UML en 24 horas
Pearson Educación
México
2000
448 p.

Sitio oficial del lenguaje PHP
<http://www.php.net/>

sitio de microsoft sql server
<http://www.microsoft.com/latam/sql/>

sitio de microsoft de windows 2000 server
<http://www.microsoft.com/latam/windows2000/>

sitio de programación
<http://www.elguruprogramador.com.ar/zonas/ver.asp?cod=98>

sitio de instalación de IIS en php
<http://www.ciberteca.net/articulos/webmaster/phpwindows/iis.asp>

sitio de programación web
<http://www.desarrolloweb.com/php/>

sitio de programación web
<http://www.webestilo.com/php/>

