



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA
DE PUEBLA**

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

**“Recursos Lingüísticos de la Recuperación de
Información en la FCC”**

TESIS PROFESIONAL

**QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN**

PRESENTA:

Rigoberto Andrade Castillo

Asesor:

Dr. David Eduardo Pinto Avendaño.

Coasesor:

M.C Beatriz Beltrán Martínez.

Puebla, Puebla.

Noviembre de 2009

AGRADECIMIENTOS.

A mis padres:

Cándida y Rufino (†)

Por haberme dado la oportunidad de poder superarme.

A mis hermanos.

A mi esposa: Eloísa.

Por el apoyo incondicional que me ha brindado.

INDICE:

Introducción.....	iii
CAPITULO I. Antecedentes y conceptos.....	1
1.1.- Conectividad y páginas web.....	2
1.1.1.- Servidores, Clientes y Enrutadores.....	4
1.1.2.- Páginas web.....	6
1.2.--Lenguajes de programación.....	7
1.2.1.- Lenguajes de programación para páginas web.....	9
1.3.- Bases de Datos.....	11
1.3.1- MySQL.....	14
1.3.2- Modelo Entidad-Relación y Modelo Relacional.....	15
1.4.- Sistemas de recuperación de información.....	18
CAPÍTULO II. Planificación.....	24
2.1.- Requerimientos del sistema.....	25
2.2.- Análisis del sistema.....	26
2.3- Diseño del sistema.....	41
2.3.1.- Diseño de datos.....	41
2.3.2.- Diseño de interfaz.....	49
2.3.3- Diagrama Funcional del sistema.....	57
CAPÍTULO III. Desarrollo.....	60
3.1.- Desarrollo e implementación del sistema.....	60
3.2.- Pruebas y depuración.....	86
CAPÍTULO IV. Resultados y conclusiones.....	92
4.1.-Resultados.....	92
4.2.-Conclusiones.....	92
Bibliografía.....	94

INTRODUCCIÓN.

En la Facultad de Ciencias de la Computación (FCC) de la BUAP, específicamente en el Laboratorio de Recuperación de Información, se ha visto la necesidad de un sitio de internet, dónde los alumnos y profesores de la maestría en Base de Datos y Recuperación de Información, puedan acceder para realizar sus prácticas correspondientes.

El presente trabajo intenta cubrir esa necesidad con la creación de una página web, donde los alumnos ya mencionados puedan realizar consultas en relación a cursos, eventos, seminarios, etc., que se realicen en el Cuerpo Académico de Bases de Datos y Recuperación de Información.

Además, el sistema debe ser administrable vía web, es decir, un administrador del sistema no necesita saber programar para administrar el contenido del sitio.

Así pues, en los próximos capítulos se describe todo el proceso tanto teórico como práctico del desarrollo de dicho sistema, esperando que sea de gran utilidad para los estudiantes y profesores de la maestría en ciencias de la computación.

CAPÍTULO I.

ANTECEDENTES Y CONCEPTOS.

Los tres últimos siglos han sido dominados cada, uno por una tecnología distinta. El siglo XVIII fue de los grandes sistemas mecánicos acompañados por la Revolución Industrial, El siglo XIX fue de la máquina de vapor. El siglo XX se centró en el procesamiento y distribución de la información. Entre otros avances, vimos crear redes complejas de telefonía, se inventó la radio y la tv, y a su vez, el nacimiento y crecimiento de la industria de la computación y los satélites de comunicación [*Computer Networks*, 4th Edt. Tanenbaum; Andrew S, Prentice Hall, 2003].

Con la llegada de las computadoras, muchas actividades en el procesamiento de información que en aquel entonces se realizaban manualmente, se facilitaron de manera vertiginosa; ahora se hacían de manera más rápida y eficiente. Las distintas necesidades de realizar cálculos y procesos de información se vieron satisfechas en el campo de la investigación, con el uso de una computadora se podían automatizar procesos, éstas tareas que anteriormente requerían mucho tiempo, ahora se podían realizar en solo unos minutos.

En las Industria los cambios no se dejaron esperar, las maquinarias ahora eran manejadas por programas de computadora y en lugar de ser manejadas por varios hombres, ahora sólo se necesitaba una persona que se encargara de monitorizar estos procesos de producción automatizados a través de la computadora.

En la Educación también hubo cambios, los libros y enciclopedias ahora ya no solo estaban en papel, si no que gracias a empresas que se dedican al desarrollo de software todos estos libros ahora podrían consultarse electrónicamente a través de un programa software en donde estaba contenida una gran cantidad de información facilitando así las investigaciones sobre cualquier área de trabajo.

Como resultado del rápido progreso en la tecnología, estas áreas convergen de tal forma que la diferencia entre recolección, transportación, almacenamiento y procesamiento de la información es prácticamente nula. Grandes organizaciones con cientos de oficinas en un área geográfica amplia pueden obtener información acerca del estado de cada oficina con solo presionar un botón [*Computer Networks*, 4th Edt. Tanenbaum; Andrew S, Prentice Hall, 2003].

Además, debido a las crecientes necesidades de tecnologías que nos facilitaran el acceso a la información empezaron a desarrollarse sistemas de conectividad que nos permitieran trabajar en grupo. Fue así, como aparecieron las llamadas Redes de Computadoras; mismas que a su vez, empezaron a conectarse entre sí, es decir, se empezaron a crear Redes de Redes; y así sucesivamente hasta que se llegó a formar una gran red mundial que es la que ahora conocemos como **Internet**.

1.1. Conectividad y Páginas Web.

Una **Red de Computadoras** es un conjunto de computadoras independientes interconectadas entre sí, éstas pueden estar conectadas a través de distintos medios físicos de comunicación como fibra óptica, cable coaxial, cable de par trenzado, línea telefónica, vía satélite, etc., los cuales nos permiten el tráfico de datos [*Guía Completa de Microsoft Windows 2000 Profesional*, Peter Norton, Prentice Hall, 2001]. Las redes de computadoras pueden tener distinto tamaño, forma y figura, de acuerdo al tamaño y al área geográfica que ocupan, se clasifican en:

- ◆ *Redes de Área Local*.- generalmente conocidas como LANs, son redes de propiedad privada que se encuentran en un solo edificio o en un campus de pocos kilómetros de longitud, se utilizan principalmente para conectar computadoras personales y estaciones de trabajo en oficinas de una empresa y de fábricas para compartir recursos

como impresoras e intercambiar información.

- ◆ *MAN*.- Red de Área Metropolitana.- ocupa una ciudad, el ejemplo más conocido de este tipo de red es el de televisión por cable disponible en varias ciudades.
- ◆ *WAN*.- Red de Área Ampla.- abarca una gran área geográfica como un país o un continente, contiene un gran conjunto de maquinas y dispositivos para la conexión de varias subredes.

Aunque parezca raro, ni *Internet* ni *Web* son una red de computadoras, Internet es una red de redes, Web es un sistema distribuido que se ejecuta sobre Internet [*Guía Completa de Microsoft Windows 2000 Profesional*, Peter Norton, Prentice Hall, 2001].

Además, estas computadoras pueden tener distintos Sistemas Operativos como por ejemplo Microsoft Windows, Linux, o Unix. Un **sistema operativo** es un conjunto de programas que administran los recursos de una computadora., tales como el acceso a la información, el acceso a la impresora, etc.

A pesar de que las computadoras tengan distintos sistemas operativos, es posible la comunicación de datos entre éstas gracias a ciertos protocolos como TCP/IP (Transmission Control Protocol / Internet Protocol) [*Guía Completa de Microsoft Windows 2000 Profesional*, Peter Norton, Prentice Hall, 2001], por mencionar alguno.

Cada computadora conectada a una red tiene una dirección IP (IP Address). Ésta dirección no es más que un identificador de una computadora en una red, la dirección debe ser única y es del tipo 200.66.0.4, ó 192.168.1.1 que es generalmente usada en subredes. Éstas direcciones pueden a su vez, relacionarse a un nombre alfanumérico, como por ejemplo cs.buap.mx gracias a los Servidores DNS (explicado en el apartado “*Servidores, Clientes y Enrutadores*”) que son computadoras encargadas de traducir ese nombre a notación numérica.

1.1.1. Servidores, Clientes y Enrutadores.

Las primeras redes de computadoras estaban basadas en el modelo Centralizado. Generalmente en éstas redes un servidor (mainframe o supercomputador) manejaba todos los aspectos de la red, mientras que cada usuario accedía al servidor principal desde una terminal; como el equipo central gestionaba todas las tareas de la red, generalmente las terminales eran económicas, equipos de bajo rendimiento [*Guía Completa de Microsoft Windows 2000 Profesional*, Peter Norton, Prentice Hall, 2001].

Actualmente gracias a los avances tecnológicos y a la rápida evolución de los computadores personales el modelo centralizado ha sido prácticamente sustituido por el modelo Cliente/Servidor. Hoy día, los usuarios tienen toda la potencia de un gran equipo al alcance de sus dedos, con la ventaja añadida de disponer de una red interconectada.

Una gran mayoría de las redes trabajan bajo el modelo cliente/servidor también conocidas como redes basadas en servidor, una estación cliente realiza una petición de datos que están almacenados en un servidor. La estación cliente procesa los datos en su propio cpu. Los resultados pueden ser almacenados entonces en el servidor para su uso futuro. Los datos también pueden ser almacenados en la estación cliente a los que pueden acceder otras estaciones cliente en la red. En las *Redes de Trabajo en Grupo* o *Redes de Igual a Igual*, en las que no hay un servidor, cada estación actúa a su vez como cliente y como servidor.

En un entorno cliente/servidor el servidor no contiene el software de interfaz de usuario; el cliente es responsable de mostrar los datos de una manera que resulte útil. El usuario introduce las instrucciones desde el equipo cliente, el equipo cliente prepara la información y envía una petición de información específica al servidor a través de la red. El servidor procesa la petición, localiza la información adecuada y la devuelve al cliente a través de la red. El equipo cliente presenta la información al usuario. El equipo cliente puede procesar adicionalmente esta información usando su propia cpu y software.

Generalmente, en un entorno cliente/servidor el servidor se dedica a almacenar y

gestionar datos. El servidor recibe las peticiones estructuradas de los clientes, las procesa y devuelve la información al cliente a través de la red.

En realidad el concepto cliente/servidor es muy genérico que puede ser entendido incluso en el ámbito de una sola máquina, donde unas aplicaciones pueden prestar servicio a otras. Sin embargo su significado puede presuponer la existencia de varias máquinas (al menos dos) conectadas en una red:

- Un servidor es cualquier máquina que dispone un recurso para ser compartido
- Un cliente es cualquier máquina que necesita un recurso externo
- Un servidor de un recurso puede ser cliente de otros y a su vez, un cliente puede ser servidor de un recurso.

Los servidores están clasificados desde un punto de vista funcional como se muestra a continuación:

- **Servidores DNS** (Domain Name Server o Servidor de Nombres de Dominio): Son computadoras que traducen los nombres alfanuméricos de una computadora en una dirección IP, es decir, en números o viceversa. Esto es logrado a través de tablas, y en caso de no tener en sus tablas la dirección numérica correspondiente a lo solicitado entonces la petición es redireccionada a otro servidor DNS que si contenga esa información. Los servidores que contienen esas tablas se denominan NS (Name Server o Servidor de Nombres).
- **Servidores de Correo** (Mail Server). Almacenan y reexpiden los mensajes de correo electrónico (e-mail). Funcionan exactamente igual que el servicio de correos convencional, la información que manejan son mensajes de correo, con la única diferencia que lo manejan electrónicamente.
- **Servidores Web** (Web Server). Almacenan “Sitios” Web, inicialmente páginas de Hipertexto en formato HTML, actualmente guardan también imágenes, archivos de sonido y de vídeo, e incluso archivos ejecutables; para su consulta o visualización desde cualquier lugar del mundo a través de un computador conectado a Internet. Los sitios se refieren a

una dirección IP también conocida como **url** (universal resource locator o localizador universal de recursos).

- **Servidores FTP** (File Transfer Protocol o Protocolo de transferencia de archivos). Son almacenes de archivos listos para ser descargados o consultados mediante técnicas FTP. Inicialmente casi toda la información habida en el Internet se encontraba almacenada en este tipo de servidores; y a pesar de que siguen siendo usados por varias instituciones y universidades, su utilización ha descendido en favor del “formato” web.
- **Servidores de Noticias.** (News Server) Almacenan una gran cantidad de mensajes hacia y desde decenas de miles de grupos de noticias que existen en la red.
- **Servidores de Terminales** (Terminal Server). Un servidor encargado de atender las llamadas telefónicas de los módem y pasar la comunicación al host (un nodo Internet) por la línea de alta velocidad. Esta función se conoce con el nombre de Front-End.

Cuando la operación entre clientes y servidores se realiza a través de una red (como es el caso de Internet) la información viaja codificada a través de redes que pueden ser del tamaño de un edificio o redes que pueden atravesar continentes alrededor del mundo. En caso de redes grandes se necesita un tercer tipo de máquinas para gestionar las transmisiones de datos, denominados **Enrutadores** (Routers).

Los enrutadores ó ruteadores (routers) tienen una funcionalidad exclusivamente transmisora de la información sin importar su uso o contenido. Su función principal es, como su nombre lo indica, buscar las mejores rutas para el tránsito de los paquetes (información).

1.1.2. Páginas Web

Una **página web** es un documento en el cual se puede mostrar información general o específica. Ésta página puede ser consultada o visualizada desde cualquier parte del mundo, a través de una computadora conectada a Internet, y está construido o escrito en algún lenguaje de programación como HTML (HyperText Markup Language).

Un documento escrito en HTML es interpretado y visualizado por un browser (navegador) como Microsoft Internet Explorer, Netscape Navigator o Mozilla Firefox por ejemplo.

Las páginas web están almacenadas en un servidor web dentro de la WWW (World Wide Web: nuestra red mundial). El protocolo usado para la consulta de este tipo de documentos es el HTTP (HyperText Transfer Protocol o protocolo de transferencia de hipertexto).

Inicialmente, las páginas web contenían únicamente texto, en la actualidad pueden contener imágenes, sonidos o vídeos que pueden ser visualizados o escuchados en el navegador, e incluso pueden ejecutarse aplicaciones con una interfaz a través del web.

Cuando una página web es solicitada desde un navegador, se especifica que se usará el protocolo http (protocolo predeterminado en los navegadores), seguido de la dirección IP o el nombre de dominio alfanumérico, además de esto, también puede especificarse exactamente que archivo solicitar. La petición es enviada al servidor web, éste, procesa esa petición y envía de regreso el archivo solicitado por el navegador que, a su vez, interpreta el código HTML y visualizará al contenido del documento.

1.2. Lenguajes de Programación.

Con el desarrollo de las primeras computadoras, vino la necesidad de programarlas, es decir, almacenar en memoria la información de la tarea que iban a ejecutar. Las primeras se usaban como calculadoras simples; se les indicaban los pasos del cálculo, uno por uno. Esto causó la necesidad de crear lenguajes que el computador pudiera “entender”.

Un lenguaje de programación permite al usuario crear programas que serán “entendidos” por la computadora (directa o indirectamente) con el objetivo de realizar una

tarea.

De una manera general, podemos clasificar los lenguajes de programación en tres categorías: máquina, de bajo nivel, de alto nivel.

El lenguaje máquina es aquel que puede ser entendido directamente por la computadora, las instrucciones en este lenguaje no son más que distintas combinaciones de 0's y 1's. Éstas especifican la operación a realizar, los registros del procesador y las celdas de la memoria a utilizar. La programación en lenguaje máquina es sumamente tediosa y complicada, por esta razón, se crearon los lenguajes de bajo nivel, que facilitaban este proceso. Este lenguaje está formado de un conjunto de instrucciones específicas de tipo alfabético comúnmente llamados mnemónicos o mnemotécnicos (generalmente abreviaturas de palabras en inglés). En los lenguajes de alto nivel, las instrucciones o sentencias son escritas con palabras similares a los de los lenguajes humanos (en la mayoría de los casos, inglés). Esta facilita la escritura y comprensión del código al programador.

Existen muchos lenguajes de alto nivel, como ejemplo podríamos citar a Ada, Basic, Cobol, Fortran, C, Pascal, Modula, Java, Perl, Php, HTML, y algunos lenguajes que incluyen un entorno gráfico y que sirven para crear programas muy vistosos y que incluso, facilitan la programación de aplicaciones, estos son los llamados “lenguajes visuales” como Visual Basic, Visual C, Delphi y C++ Builder por mencionar algunos.

Los programas escritos en lenguaje de alto nivel (programas fuente) no son entendibles directamente por la computadora, si no que necesitan de un programa traductor (compilador) que se encarga de transformar el código fuente a otro lenguaje que la computadora pueda entender.

Cada lenguaje de programación es usado de acuerdo a la finalidad o uso que vaya a tener la aplicación. Por ejemplo, existen lenguajes que generalmente son usados para crear programas de oficina. Otros que requieren que haya conectividad entre un servidor y algunos clientes (generalmente programas que incluyen el manejo de bases datos), o programas que

simplemente sean ejecutados a través de una página web en el internet.

1.2.1 Lenguajes de programación para páginas web.

Existe una gran cantidad de lenguajes usados para el desarrollo de páginas web, cada uno de éstos con sus propias características a favor o en contra de acuerdo al tipo de página que se desee crear, entre éstas páginas podemos mencionar aquellas que solo contienen texto y tal vez una que otra imagen (páginas estáticas), éstas páginas son enviadas al cliente (previa petición de la misma) tal cual se encuentra en el servidor. Otro tipo de páginas son las llamadas páginas dinámicas, éstas son generadas al momento en que el cliente la solicita; generalmente cuando se necesita mostrar alguna información contenido en bases de datos. Cuando el cliente solicita una página dinámica, el servidor ejecuta un programa que es el que generará la información que va a ser enviada al cliente como una página html.

Algunos lenguajes utilizados en la creación de páginas web son las siguientes:

- HTML: (HyperText Markup Language). Es el lenguaje básico para la creación de páginas web que es interpretado directamente por cualquier navegador.
- Java: (<http://java.sun.com/>) Es un lenguaje creado por Sun Microsystems y es parecido el lenguaje C. El lenguaje java es a la vez compilado e interpretado. Con el compilador se convierte el código fuente que reside en archivos cuya extensión es **.java**, a un conjunto de instrucciones que recibe el nombre de *bytecodes* que se guardan en un archivo cuya extensión es **.class**. Estas instrucciones son independientes del tipo de ordenador. El intérprete ejecuta cada una de estas instrucciones en un ordenador específico (Windows, Macintosh, etc). Solamente es necesario, por tanto, compilar una vez el programa, pero se interpreta cada vez que se ejecuta en un ordenador. Los tipos de programas más comunes que se pueden hacer con Java son los applets (se ejecutan en el navegador de la máquina cliente) y las aplicaciones (programas que se ejecutan directamente en la JVM – Java Virtual Machine). Otro tipo especial de programa se denomina servlet que es similar a los applets pero se ejecutan en los servidores Java.

- JavaScript: Es un lenguaje de programación desarrollado por Netscape Corporation para su navegador Netscape Navigator 2.0, para permitir la ejecución de código dentro de las páginas en HTML. Gracias a los programas (llamados guiones) escritos en este lenguaje y embebidos en las páginas HTML, se pueden conseguir interesantes efectos en las páginas web, comprobar la validez de la entrada de formularios, abrir y cerrar ventanas, cambiar dinámicamente el aspecto y los contenidos de una página, cálculos matemáticos sencillos y mucho más.
- ASP: (Active Server Pages) es la tecnología para la creación de páginas dinámicas del lado del servidor desarrollada por Microsoft. El tipo de servidores que emplean este lenguaje son aquellos que funcionan con sistema operativo de la familia de Windows.
- PHP: Acrónimo de “PHP: Hypertext Preprocessor”, es un lenguaje interpretado de alto nivel para páginas HTML, y ejecutado en el servidor. Nacido en 1994, se trata de un lenguaje de creación relativamente creciente que ha tenido una gran aceptación en la comunidad de desarrolladores de páginas web debido sobre todo a la potencia y simplicidad que lo caracterizan. PHP nos permite embeber sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados íntegramente en un lenguaje distinto al HTML. Por otra parte, y es aquí donde reside su mayor interés con respecto a los lenguajes pensados para los CGI, PHP ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana, sin complicaciones. PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones. ASP, siendo una tecnología Microsoft, está orientado hacia sistemas Windows, especialmente de la familia NT.

PHP es el lenguaje que se pretende usar en el presente trabajo, por las características que ya se han mencionado.

1.3. Bases de Datos.

Una Base de Datos (BD) es una colección de información ordenada e interrelacionada que es de importancia para una empresa [*Fundamentos de Bases de Datos*, Silberschatz, Abraham; Korth, Henry F; Sudarshan, S, McGraw-Hill].

Las bases de datos son ampliamente usadas. Las siguientes son algunas de sus aplicaciones más representativas:

- *Banca*. Para información de los clientes, cuentas y préstamos, y transacciones bancarias.
- *Líneas aéreas*. Para reservas e información de planificación. Las líneas aéreas fueron de los primeros en usar las bases de datos de forma distribuida geográficamente (los terminales situados en todo el mundo accedían al sistema de bases de datos centralizado a través de las líneas telefónicas y otras redes de datos).
- *Universidades*. Para información de los estudiantes, matrículas de las asignaturas y cursos.
- *Transacciones de tarjetas de crédito*. Para compras con tarjeta de crédito y generación mensual de extractos.
- *Telecomunicaciones*. Para guardar un registro de las llamadas realizadas, generación mensual de facturas, manteniendo el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.
- *Finanzas*. Para almacenar información sobre grandes empresas, ventas y compras de documentos formales financieros, como bolsa y bonos.
- *Ventas*. Para información de clientes, productos y compras.
- *Producción*. Para la gestión de la cadena de producción y para el seguimiento de la producción de elementos en las factorías, inventarios de elementos en almacenes y pedidos de elementos.
- *Recursos humanos*. Para información sobre los empleados, salarios, impuestos y beneficios, y para la generación de las nóminas.

Como esta lista ilustra, las bases de datos forman una parte esencial de casi todas las empresas actuales.

La creación de bases de datos debe ser realizada cuidadosamente de tal forma que puedan facilitar el acceso a la información y con buena rapidez, cuidando también que no haya redundancia en los datos (datos repetidos), ya que esto provocaría inconsistencias en la información.

La mayoría de los sistemas informáticos en la actualidad manejan una gran cantidad de datos y aquellos que están accesibles a través de internet no son una excepción.

Anteriormente se utilizaba el sistema de procesamiento de archivos en el cual toda la información de la empresa se almacenaba en archivos permanentes y cada archivo se asociaba con un programa que permitieran el acceso a la información; es decir, cada sistema o programa podía almacenar u obtener la información de los archivos para la cual estaba programada, y otras aplicaciones no podían acceder a esta información. Este tipo de estructura tenía los inconvenientes asociados al hecho de que existiese una redundancia no controlada, con la que se podían producir inconsistencias, muy peligrosas para la integridad de los datos. Así, por ejemplo, los datos de un cliente podrían aparecer en el archivo del departamento de facturación y estar repetidos o diferentes en el departamento de publicidad de una determinada identidad. Al figurar dichos datos en más de un sitio, su actualización resultaba complicada, y se corría el riesgo de modificarla en uno de los dos sitios y en el otro no. Cuando tiene lugar este tipo de operaciones, se puede dar el caso en que no se puede diferenciar cuál es la información correcta (datos inconsistentes).

Para evitar estas situaciones se diseñaron los sistemas gestores de bases de datos. Un Sistema de Gestión de Bases de Datos (SGBD) consiste de una colección de datos interrelacionados y un conjunto de programas que nos permiten acceder a dichos datos [*Fundamentos de Bases de Datos*, Silberschatz, Abraham; Korth, Henry F; Sudarshan, S, McGraw-Hill]. El primer objetivo de un SGBD es proporcionar un entorno que sea tanto práctico como eficiente de usar en el almacenamiento y recuperación de información de la

BD. Éstos sistemas “envuelven” y “protegen” los datos, en la medida de lo posible, frente a manipulaciones indebidas, al mismo tiempo que integran una serie de herramientas que gestionan, entre otras cosas, la manipulación completa de los datos, los accesos concurrentes, las copias de seguridad, la “integración” con programas en lenguaje de uso general, etc. Pero, sobre todo, los SGBD proporcionan a los datos una característica básica, que es la independencia respecto de los programas que los usan y que lleva a evitar la redundancia.

Los SGBD pueden representar la información de diferentes maneras: modelo jerárquico, modelo de red, etc; aunque por el momento nos centraremos mas en el Modelo Entidad–Relación y el Modelo Relacional como se verá más adelante.

Para poder atender a las necesidades de los diferentes tipos de usuarios, los gestores de bases de datos deben proporcionar distintas formas de trabajar. Parece claro que no es lo mismo que un usuario desee, en un momento dado, realizar una pequeña consulta del estilo de “deseo conocer todos los clientes que viven en la zona centro” a que dicho usuario esté realizando una gestión informática donde se registre y recupere información de una forma masiva. La mayoría de los gestores comerciales proporcionan herramientas de desarrollo y lenguajes propios de acceso que permiten desde una consulta sencilla hasta el acceso desde el programa más complicado. Sin embargo, todos los gestores de bases de datos relacionales permiten la utilización en sus entornos de un lenguaje estándar que facilita enormemente el trabajo: el lenguaje de consulta estructurado o SQL (Structured Query Language). Este lenguaje es enormemente fácil, intuitivo y potente y se puede usar desde los dos entornos tradicionales: como lenguaje interpretado para la realización de pequeñas consultas y como lenguaje embebido en otro para la realización de sistemas completos.

Un Sistema de Bases de Datos (SBD) es una colección de archivos interrelacionados y un conjunto de programas que permitan a los usuarios acceder y modificar estos archivos. Uno de los propósitos principales de un sistema de un SBD es proporcionar al usuario una visión abstracta de los datos. Es decir el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

1.3.1.- MySQL

MySQL es un gestor de bases de datos accesible de forma gratuita (su licencia se puede consultar en su propia documentación) en <http://www.mysql.com/> y está disponible para las principales plataformas del mercado. Al margen de algunas herramientas y pequeños programas para diferentes usos, básicamente se compone de un programa que actúa de servidor y que atiende las peticiones de los clientes (principalmente vía TCP). El software servidor tiene numerosas opciones para ejecutarse.

Los programas cliente se encuentran también disponibles en la misma url y sirven para establecer la conexión con el servidor y solicitar la ejecución de las consultas que se estimen oportunas.

Por lo antes expuesto, MySQL es el SGBD que usaremos en el presente trabajo.

Un sistema gestor de bases de datos debe proporcionar un lenguaje de definición de datos, y un lenguaje de manipulación de datos, mismos que MySQL sí proporciona.

El lenguaje de definición de datos se utiliza para especificar el esquema de la base de datos. Por ejemplo, la siguiente instrucción en lenguaje SQL define la tabla *cuenta*

Create table cuenta (numero-cuenta **char(10)**, saldo **integer**);

El lenguaje de manipulación de datos permite a los usuarios acceder o manipular la información en nuestra base de datos; esto es: La recuperación de información almacenada en la base de datos, la inserción de información nueva en la base de datos, el borrado de información en la base de datos ó, la modificación de información almacenada en la base de datos.

Así, una consulta es una instrucción de solicitud para recuperar información. Por

ejemplo:

Select * from cliente where id-cliente="19.283.746";

Nos devolverá toda la información de “González” contenida en la tabla *cliente*.

1.3.2.- Modelo Entidad-Relación y Modelo Relacional.

El **Modelo de Datos Entidad-Relación (E-R)** está basado en una percepción del mundo real que consta de una colección de objetos básicos llamados *entidades*, y de *relaciones* entre estos objetos [*Fundamentos de Bases de Datos*, Silberschatz, Abraham; Korth, Henry F; Sudarshan, S, McGraw-Hill]. Una entidad es una “cosa” u “objeto” en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad y las cuentas bancarias pueden ser consideradas entidades.

Las entidades se describen en una base de datos mediante un conjunto de atributos. Por ejemplo, los atributos *numero-cuenta* y *saldo* describen una cuenta particular de un banco y pueden ser atributos de la entidad *cuenta*. De igual forma, los atributos *nombre-cliente*, *calle-cliente* y *ciudad-cliente* pueden describir la entidad *Cliente*. Además un atributo extra *id-cliente* podría usarse para identificar unívocamente a los clientes, ya que podría haber dos clientes con el mismo nombre, calle y ciudad.

Una relación es una asociación entre varias entidades. Por ejemplo, una relación *impositor* asocia un cliente con cada cuenta que tiene. El conjunto de todas las entidades del mismo tipo y el conjunto de todas las relaciones del mismo tipo se denominan *conjunto de entidades* y *conjunto de relaciones* respectivamente.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama E-R, que consta de los siguientes componentes: Rectángulos, que representan conjuntos de entidades; Elipses, que representan atributos; Rombo, que

representan relaciones entre conjuntos de entidades; Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

Cada componente se etiqueta con la entidad o relación que representa.

Como ejemplo, consideremos como parte de un sistema bancario que consista en clientes y las cuentas que tienen los clientes; así, tendríamos el diagrama Entidad-Relación que se muestra en la siguiente figura.

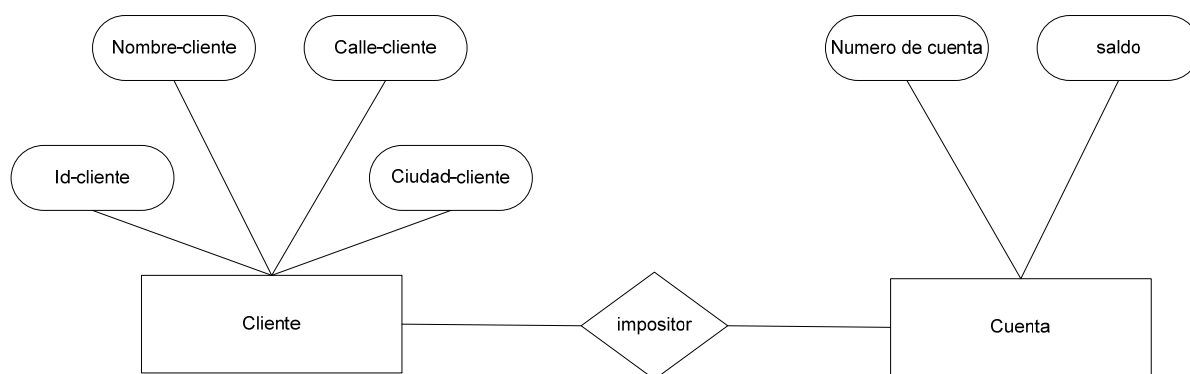


Fig 1.3.1 .- Ejemplo de Diagrama E-R

El diagrama E-R indica que hay dos conjuntos de entidades cliente y cuenta, con los atributos descritos anteriormente. El diagrama también muestra la relación impositor entre cliente y cuenta.

En el **modelo relacional** se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único. En la Figura siguiente se presenta un ejemplo de base de datos relacional consistente en tres tablas: la primera muestra los clientes de un banco, la segunda, las cuentas, y la tercera, las cuentas que pertenecen a cada cliente.

<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda

67.789.901	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsaín
32.112.312	Santos	Mayor	Peguerinos
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

(a) La tabla *cliente*

número-cuenta	saldo
C-101	500
C-215	700
C-102	400
C-305	350
C-201	900
C-217	750
C-222	700

(b) La tabla *cuenta*

id-cliente	número-cuenta
19.283.746	C-101
19.283.746	C-201
01.928.374	C-215
67.789.901	C-102
18.273.609	C-305
32.112.312	C-217
33.666.999	C-222
01.928.374	C-201

(c) La tabla *impositor*

La primera tabla, la tabla *cliente*, muestra, por ejemplo, que el cliente cuyo identificador es 19.283.746 se llama González y vive en la calle Arenal sita en La Granja.

La segunda tabla, *cuenta*, muestra que las cuentas C-101 tienen un saldo de 500 y la C-201 un saldo de 900 respectivamente.

La tercera tabla, *impositor*, muestra las cuentas que pertenecen a cada cliente. Por ejemplo, la cuenta C-101 pertenece al cliente cuyo identificador es 19.283.746 (González), y los clientes 19.283.746 (González) y 01.928.374 (Gómez) comparten el número de cuenta A-201 (pueden compartir un negocio).

El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de

registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro.

No es difícil ver cómo se pueden almacenar las tablas en archivos. Por ejemplo, un carácter especial (como una coma) se puede usar para delimitar los diferentes atributos de un registro, y otro carácter especial (como un carácter de nueva línea) se puede usar para delimitar registros. El modelo relacional oculta tales detalles de implementación de bajo nivel a los desarrolladores de bases de datos y usuarios.

El modelo de datos relacional es el modelo de datos más ampliamente usado, y una amplia mayoría de sistemas de bases de datos actuales se basan en el modelo relacional. El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos E-R.

Los diseños de bases de datos a menudo se realizan en el modelo E-R, y después se traducen al modelo relacional.

Nótese también que es posible crear esquemas en el modelo relacional que tengan problemas tales como información duplicada innecesariamente. Por ejemplo, supongamos que se almacena número-cuenta como un atributo del registro cliente. Entonces, para representar el hecho de que las cuentas C-101 y C-201 pertenecen ambas al cliente González (con identificador de cliente 19.283.746) sería necesario almacenar dos filas en la tabla cliente. Los valores de nombre-cliente, calle-cliente y ciudad-cliente de González estarían innecesariamente duplicados en las dos filas

1.4. Sistemas de Recuperación de Información.

Un Sistema de Recuperación de Información (SRI) es, como su nombre lo indica, un programa software que nos permite la recuperación de la información, previamente

almacenada, por medio de la realización de una serie de consultas (“queries”) a los documentos contenidos en la base de datos. Esta serie de preguntas o interrogaciones se conceptúan como *sentencias formales de expresión de necesidades de información*, y suelen venir expresadas por medio de un lenguaje de interrogación.

Un documento es un objeto de datos, textual tradicionalmente, aunque la evolución tecnológica propicia la profusión de documentos multimedia, incorporándose fotografías, ilustraciones gráficas, vídeo animado, audio, etc.,

Un SRI debe soportar una serie de operaciones básicas sobre los documentos almacenados en el mismo, como son: introducción de nuevos documentos, modificación de los documentos almacenados y eliminación de los mismos.

Debemos también contar con algún método de localización de los documentos (o con varios generalmente), para presentárselos posteriormente al usuario.

En la Facultad de Ciencias de la Computación, particularmente en el cuerpo de bases de datos y recuperación de información, se han elaborado diversos trabajos relacionados con la Recuperación de Información (RI). Cada uno de estos trabajos es referido y explicado a continuación.

En enero de 2003, se realizó **Cálculo de la Subsunción entre Términos** esta herramienta nos permite determinar la relación léxica, si existe, entre dos palabras; un sentido de una y un sentido de la otra. Permite obtener jerarquías de términos: de los más generales a los más específicos. El sistema requiere un corpus de un dominio y la elección de dos términos para calcular su grado de subsunción. Arroja el resultado como una relación de hiponimia, sinonimia, relación o sin relación.

Actualmente se está empleando en los cursos del postgrado: Recuperación de Información y Procesamiento de Lenguaje Natural. Así como en algunas tesis de maestría y proyectos de investigación.

En julio de 2003, se desarrolló un **Sistema de Recuperación de Información con Base en Extractos** el cual permite experimentar con un modelo alternativo de representación textual. El propósito es lograr búsquedas semánticas más que por palabras, esto es, búsqueda por el significado más que por cadenas de caracteres. Búsqueda de documentos dentro de una colección. El sistema se basa en la representación de un texto mediante un extracto (usando el generador de extractos). Una vez representado el texto de esta forma es incluido en un sistema de recuperación que usa el modelo booleano para efectuar la búsqueda considerando expansiones de consultas y documentos mediante un corpus.

Actualmente se está empleando en los cursos del posgrado: Recuperación de Información y Procesamiento de Lenguaje Natural. Así como en algunas tesis de maestría y proyectos de investigación.

En septiembre de 2003, se desarrolló un **Generador de Palabras Relacionadas** cuya aportación se basa en la gran necesidad de trabajar con volúmenes grandes de textos, como en los resultados de la búsqueda de material en Internet, obliga a desarrollar herramientas que seleccionen elementos de la búsqueda para ser procesada, o bien elijan de las respuestas encontradas las mejores. Este procedimiento se basa en diccionarios de diferente tipo. El programa desarrollado es un elemento indispensable en las tareas antes mencionadas, por lo cual nos permite realizar investigación en estos temas.

Se está utilizando el sistema que genera palabras relacionadas en la construcción de un tesoro para el dominio de Economía. Se pretende que este tesoro sea utilizado para enriquecer las búsquedas en los sistemas de recuperación de información o sistemas bibliotecarios; como aplicaciones inmediatas.

El sistema toma como base un texto grande (corpus de un dominio particular) de al menos un MB. Efectúa un análisis de frecuencias de las palabras de su vocabulario, para determinar los vecinos cercanos y formar listas de palabras relacionadas. Parte de una lista de sustantivos previamente extraída del vocabulario.

Actualmente se está empleando en los cursos del posgrado: Recuperación de Información y Procesamiento de Lenguaje Natural. Así como en algunas tesis de maestría y proyectos de investigación.

En octubre de 2003, se elaboró **Categorización Binaria de Textos** el cual permite experimentar con una representación novedosa de textos. La elección de las palabras índice que representan un texto, son evaluadas según su cercanía a una frecuencia distinguida que separa las palabras de alta y baja frecuencia. Su aplicación en sistemas de recuperación de información o procesamiento de lenguaje natural es múltiple por tratarse de una representación de textos.

El uso más común es en la clasificación de noticias según su tema. Pero una aplicación importante es en la clasificación de mensajes, encuestas (cuyas respuestas está en lenguaje natural). Toma un texto y decide si pertenece o no a un genero noticioso. Está en proceso la categorización de textos (no binaria), con y sin traslapes entre géneros.

Actualmente se está empleando en los cursos del posgrado: Recuperación de Información y Procesamiento de Lenguaje Natural. Así como en algunas tesis de maestría y proyectos de investigación.

También en octubre de 2003, se elaboró **Aproximación de Conceptos** que permite la experimentación con diccionarios onomasiológicos. Está en etapa de pruebas su empleo en sistemas de recuperación de información. El objetivo central de este sistema es partir de una descripción y obtener los nombres de los conceptos que son definidos equivalentemente a la descripción.

La principal aplicación es, por ahora, en la experimentación con diccionarios onomasiológicos. El grupo de Ingeniería Lingüística de la UNAM realiza este trabajo desde hace tiempo y participa en su evaluación. Toma una consulta que describe cierto fenómeno o entidad de la Física y responde con los nombres de los conceptos cuya definición "más se

acercan a tal descripción".

Actualmente se está empleando en los cursos del postgrado: Recuperación de Información y Procesamiento de Lenguaje Natural. Así como en algunas tesis de maestría y proyectos de investigación.

En noviembre de 2003, se desarrolló **Extractor de Textos** el cual facilita la elección de materiales sea para lectura o representación textual. Está en etapa de pruebas para su mejoramiento. Se pretende ponerlo en una página Internet para su empleo por cualquier usuario. Toma un texto de un dominio (ahora maneja solamente tres: sociedad, política y economía) y obtiene las oraciones más representativas. Puede ser empleado también en la representación de textos para sistemas de recuperación de información.

El sistema realiza un preprocesado del texto original para trabajar con pseudoraíces de las palabras, eliminando previamente las palabras cerradas. Enseguida representa estas nuevas oraciones con ayuda de un corpus y calcula la similitud entre cada oración y el texto completo. Elige las oraciones de mayor similitud para representar el extracto.

Actualmente se está empleando en los cursos del posgrado: Recuperación de Información y Procesamiento de Lenguaje Natural. Así como en algunas tesis de maestría y proyectos de investigación.

En diciembre de 2003, se liberó un **Etiquetador de Textos en Español** y el aporte principal es que con este sistema es posible llevar a cabo aplicaciones como las señaladas en "Uso" pero el aporte más importante es que el sistema se ha desarrollado aquí y con ello es posible tanto actualizar el sistema como reutilizar su código para otros propósitos en la investigación y la docencia.

Es una herramienta especializada y poco difundida, sobre todo para el español. Su función principal es ayudar en el tratamiento de textos: Extracción de información, parsing parcial, desambiguación del sentido de una palabra, etc.

El etiquetador con partes de la oración desarrollado, es un sistema que toma como entrada un archivo con texto plano, previamente preprocesado (identificación de tokens), y obtiene como resultado un archivo que contiene cada una de las palabras del textos original con su categoría gramatical.

Actualmente se está empleando en los cursos del postgrado: Recuperación de Información y Procesamiento de Lenguaje Natural. Así como en algunas tesis de maestría y proyectos de investigación.

En abril de 2004, se desarrolló un **Sistema de Recuperación de Información con Expansión de Consultas** en donde permite experimentar con un modelo alternativo para la extracción de documentos de Internet, usando expansión de consultas basada en relaciones de hiperonímia-hiponímia. Búsqueda de documentos en Internet.

El sistema se basa en la expansión de conceptos por medio de relaciones semánticas de hiperonímia-hiponímia. Usa expansión para enriquecer los resultados obtenidos en una consulta, usando como interfaz de búsqueda una aplicación basada en el Web.

Actualmente, el sistema está en experimentación, pero se piensa liberar al público en general.

CAPITULO II.

PLANIFICACIÓN.

Es ampliamente aceptado que el costo de corregir-modificar un sistema después de su instalación, o incluso después de las primeras etapas de diseño, es alto y quizá mayor que el costo de preparar una especificación de requerimientos antes que todo.

Por lo tanto, la preparación de una especificación adecuada de requerimientos reduce los costos y el riesgo general asociado con el desarrollo.

No existe en concreto una clara definición de lo que debe incluirse en la especificación de requerimientos. La distinción entre requerimientos y diseño ha sido siempre fácil de establecer y difícil de clarificar:

- *Requerimiento*: Lo que el sistema debe hacer.
- *Diseño*: Cómo debe hacerlo el sistema.

Estas definiciones son algo arbitrarias y necesitan ser refinadas en el futuro, puesto que es difícil hacer una clara distinción entre clarificación de requerimientos y diseño de alto nivel. Una especificación ideal de requerimientos debe decir lo que el usuario necesita (en términos del sistema y su ambiente); es decir relacionar entradas y salidas. No dice nada sobre la estructura del sistema pero deben establecerse restricciones (como tiempo de entrega).

De manera desafortunada, la estructura es a menudo una clave para el entendimiento. Una especificación de requerimientos sin estructura es a menudo oscura y anula su propósito.

La preparación de una especificación de requerimientos, debe describir cada aspecto posible del comportamiento del sistema, de tal forma que se necesita un entendimiento total de

los problemas del usuario, deseos y ambiente en el cual el sistema será instalado. [*Ingeniería de Software Explicada*, Norris, Mark; Rigby Peter, Megabyte-Noriega Editores. 1994].

2.1. Requerimientos del sistema.

En el Cuerpo Académico de Bases de Datos y Recuperación de Información de la Facultad de Ciencias de la Computación se han elaborado diversos programas software que nos permiten la recuperación de información. La necesidad de poder acceder a éstos ha motivado la realización de un sistema que nos facilite la utilización de dichos programas de una manera fácil, a través del Internet. Además del acceso a estos programas también se podrá acceder a otro tipo de información que sea útil para el Cuerpo Académico de Bases de Datos y Recuperación de Información.

El sistema que se requiere consta de una página web, cien por ciento administrable desde un navegador; ya que no será necesario algún lenguaje de programación para poder realizar cambios en el contenido de la información. Debe ser una página dinámica, porque todo lo que se visualice a través de un navegador será generado al momento de consultar la página.

En el apartado de consulta, que será la página inicial al acceder desde un navegador podrá seleccionar la siguiente información:

- **Información.**- Información de carácter general tanto para los miembros del cuerpo académico como para los visitantes a través del web. Aquí encontraremos avisos, noticias e información sobre eventos. La información será generada y subida al sitio por algún profesor del cuerpo académico.
- **Profesores.**- Información de los profesores que colaboran en el cuerpo académico.
- **Estudiantes.**- Información de los estudiantes que colaboran en el cuerpo académico.
- **Cursos.**- Aquí podremos ver los cursos que se imparten en el cuerpo académico. Y estos

cursos a su vez, tendrán lecturas recomendadas lo que implica generar un elemento más: “**lectura**”.

- **Seminarios.**- Información acerca de seminarios que se hayan realizado dentro del cuerpo académico.
- **Recursos.**- Recursos que serán útiles en la ejecución de las herramientas de recuperación de información que se han generado en el cuerpo académico.
- **Herramientas.**- Las herramientas de recuperación de información que se han generado en el cuerpo académico.
- **Proyectos.**- Los proyectos que se han realizado o se tienen en proceso en el cuerpo académico,
- **Artículos.**- Artículos que han sido publicados ya sea por parte de profesores o alumnos que participan en el cuerpo académico.

En lo sucesivo, nos referiremos a cada uno de los elementos enlistados previamente, y a los nuevos que se vayan a generar durante el diseño del sistema como “**objetos de datos**”. Un objeto de datos es una representación de cualquier composición de información compuesta que deba comprender el software. Por composición de información entendemos todo aquello que tienen un número de propiedades o atributos diferentes. Por mencionar un ejemplo: *Profesor* es un objeto de datos ya que se compone de propiedades como *nombre, dirección, teléfono, etc.*, también llamados atributos.

Así mismo, habrá un apartado donde se podrá administrar todo el contenido del sitio. Para ingresar a esta parte se tendrá que iniciar sesión, tecleando un nombre de usuario y su contraseña correspondiente. Aquí, a través de menús, se irá seleccionando los objetos de datos a administrar, y a través de ventanas de captura, modificación, etc., se podrán realizar las distintas tareas de administración del sitio. Al término de las actividades de administración se tendrá que cerrar la sesión para regresar nuevamente a la ventana inicial de consultas.

2.2 Análisis del sistema.

Antes de crear una base de datos que nos permita almacenar toda la información de

nuestra página web, debemos analizar qué información nos interesa que se pueda consultar, así que tomamos los elementos mencionados en la lista que se encuentra en la sección 2.1 (requerimientos del sistema), y de esta forma crear un Diagrama basado en el modelo Entidad-Relación, como se muestra en la figura 2.1.

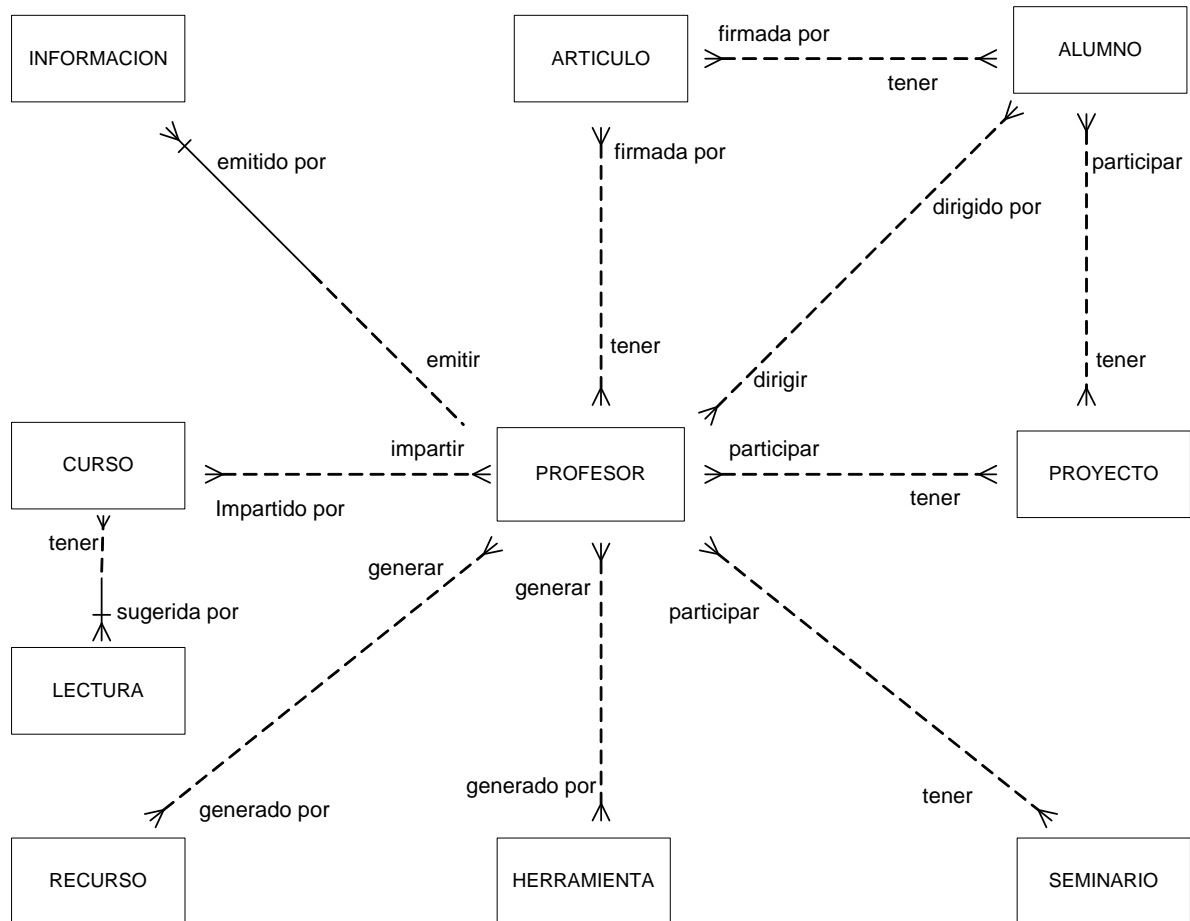


Figura 2.1.- Diagrama Entidad-Relación inicial

Las relaciones que se encuentran en la figura 2.1 son las siguientes:

1. **Profesor-Información.**- Un profesor **puede emitir** uno o más elementos de información, pero cada elemento de información **debe ser emitido** por uno y solo un profesor.
2. **Profesor-Curso.**- Un profesor **puede impartir** uno o más cursos y, un curso **puede ser impartido** por uno o más profesores.

3. *Curso-Lectura.*- Un curso **puede sugerir** una o más lecturas, una lectura **debe ser sugerida** para uno o varios cursos.
4. *Profesor-Recurso.*- Un profesor **puede generar** uno o más recursos, y un recurso **puede ser generado** por uno o más profesores.
5. *Profesor-Herramienta.*- Un profesor **puede generar** una o más herramientas, y una herramienta **puede ser generada** por uno o más profesores.
6. *Profesor-Seminario.*- Un profesor **puede participar** en un seminario, y un seminario **puede tener** uno o más profesores como participantes.
7. *Profesor-Proyecto.*- Un profesor **puede participar** en uno o más proyectos, y un proyecto **puede tener** uno o más profesores como participantes.
8. *Profesor-Alumno.*- Un profesor **puede dirigir** a uno o más alumnos, y un alumno **puede ser dirigido** por uno más profesores.
9. *Profesor-Artículo.*- Un profesor **puede tener** uno o más artículos, y un artículo **puede ser firmado** por uno o más profesores.
10. *Alumno-Artículo.*- Un alumno **puede tener** uno o más artículos, un artículo **puede ser firmado** por uno o más alumnos.
11. *Alumno-Proyecto.*- Un alumno **puede participar** en uno o más proyectos, un proyecto **puede tener** uno o más alumnos como participantes.

Como podemos ver, todas las relaciones representadas en la Figura 2.1 excepto *Profesor-Información*, son relaciones de **muchos a muchos**, lo cual no es conveniente ya que este tipo de relaciones siempre tienen por lo menos una entidad perdida, por lo tanto entre las dos entidades que forman la relación, debe interponerse una tercera entidad, como se muestra en la Figura 2.2, y así, eliminar esas relaciones de muchos a muchos.

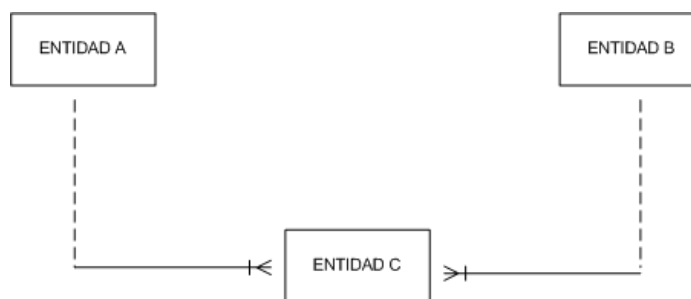


Figura 2.2.- Una tercera entidad resuelve el problema de la Fig. 2.1

Así, las relaciones de muchos a muchos que tenemos en la figura 2.1, podrían arreglarse de la siguiente manera:

Profesor-Curso

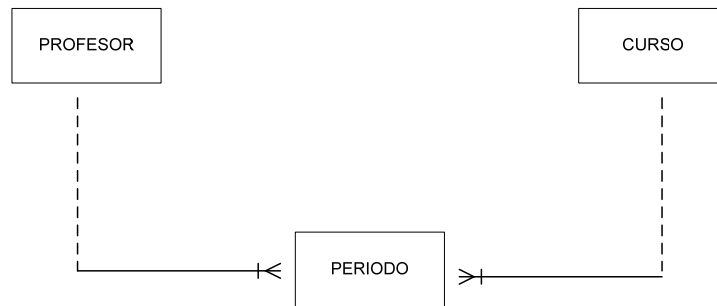


Figura 2.3.- Solución a relación Profesor-Curso

La figura 2.3 muestra como la entidad *Periodo* resuelve el problema de la relación muchos a muchos que existía entre Profesor y Curso. Así, un Profesor puede impartir en distintos periodos, en un Periodo debe impartir un Profesor; en un Periodo debe ser impartido un Curso, y un Curso puede ser impartido en uno o más Periodos.

Curso-Lectura

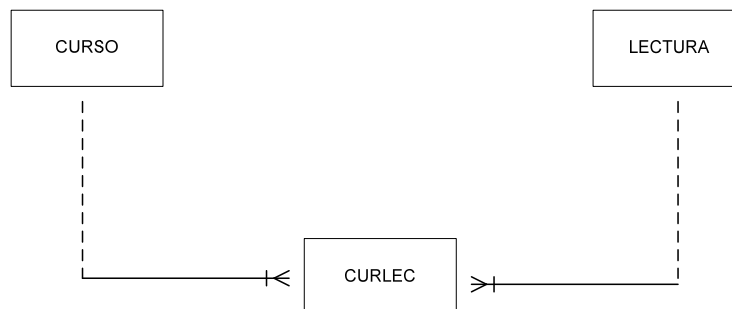


Figura 2.4.- Solución a la relación Curso-Lectura

La figura 2.4 muestra como la entidad *Curlec* resuelve el problema de la relación de muchos a muchos que existía entre Curso y Lectura. Así, un Curso puede sugerir distintas lecturas y, una Lectura puede ser sugerida en varios cursos.

Profesor-Recurso

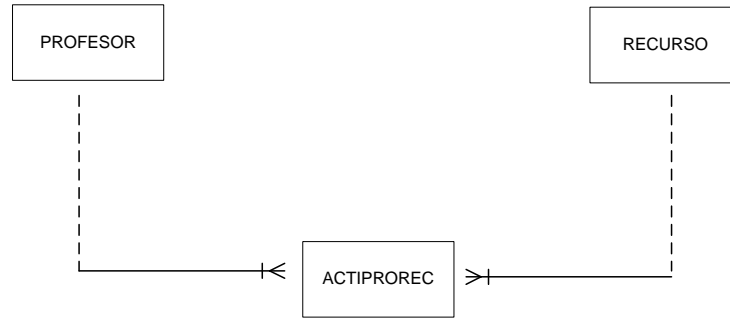


Figura 2.5.- Solución a relación Profesor-Recurso

En la Figura 2.5, la nueva entidad *Actiprorec* que entenderemos como *Actividad de un profesor en la generación de un recurso*, soluciona la relación de muchos a muchos que tenían las entidades Profesor y Recurso. Así, un profesor puede tener asignada una o más actividades, una actividad debe ser asignada a un Profesor; una Actividad debe generar un Recurso y un recurso puede ser generado por una o más actividades.

Profesor-Herramienta

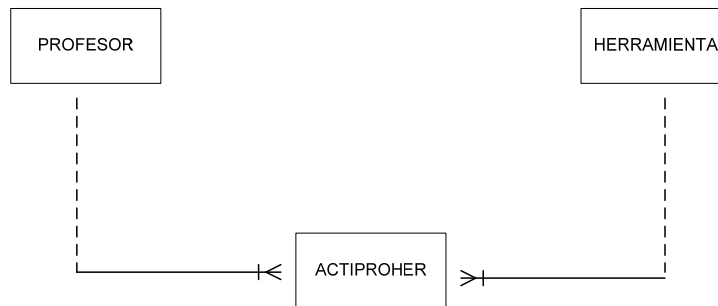


Figura 2.6.- Solución a relacion Profesor-Herramienta

En la Figura 2.6, la nueva entidad *Actiproher* que entenderemos como *Actividad de un profesor en la generación de una herramienta*, soluciona la relación de muchos a muchos que tenían las entidades Profesor y Herramienta. Así, un profesor puede tener asignada una o más actividades, una actividad debe ser asignada a un Profesor; una Actividad debe generar una Herramienta y una Herramienta puede ser generada por una o más actividades.

Profesor-Seminario

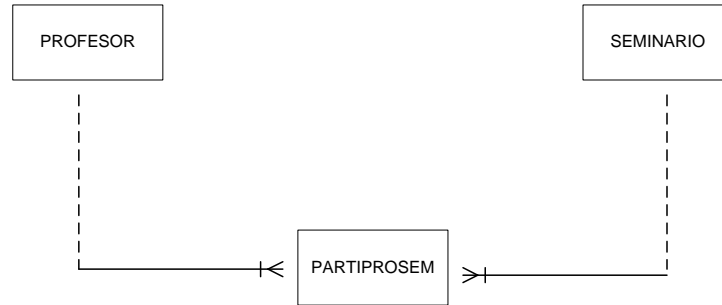


Figura 2.7.- Solución a relacion Profesor-Seminario

La figura 2.7 muestra la solución que se le da a la relación de muchos a muchos que existía entre las entidades Profesor y Seminario, la entidad *Partiprosem* la entenderemos como *Tipo de participación de un profesor en un seminario*. Un Profesor puede tener uno o más Tipos de participación, un Tipo de participación debe estar asignada a un Profesor; un Tipo de participación debe pertenecer a un Seminario y, un Seminario puede tener uno o más Tipos de Participación.

Profesor-Proyecto

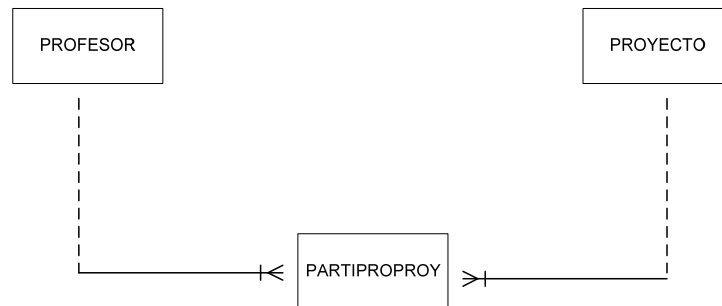


Figura 2.8.- Solución a relacion Profesor-Proyecto

La figura 2.8 presenta la forma en que se soluciona la relación de muchos a muchos entre las entidades Profesor y Proyecto, con la entidad *PARTIPROPROY* que entenderemos como *Tipo de participación*. Un Profesor puede tener uno o más Tipos de participación, un Tipo de participación debe estar asignado a un Profesor; un Tipo de participación debe pertenecer a un Proyecto y, un Proyecto puede tener uno o más Tipos de participación.

Profesor-Alumno

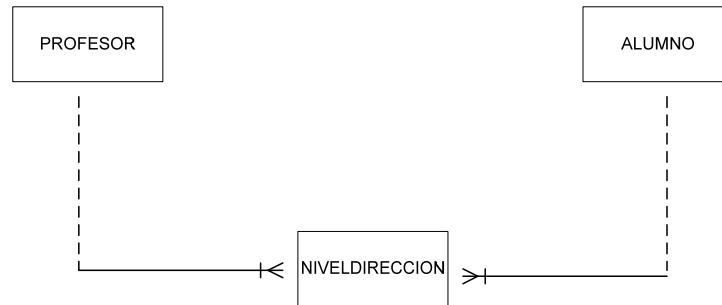


Figura 2.9.- Solución a relacion Profesor-Alumno

En la figura 2.9 la entidad *Niveldirección* resuelve la relación de muchos a muchos entre las entidades Profesor y Alumno. Un Profesor puede tener uno o más niveles de dirección, un Nivel de dirección debe estar ocupado solo por un profesor; un Nivel de dirección debe tener un alumno, y un alumno puede estar dirigido en uno o más niveles de dirección. En nuestro caso y por conveniencia, solo manejaremos dos Niveles de dirección: Asesor y Coasesor.

Profesor-Artículo

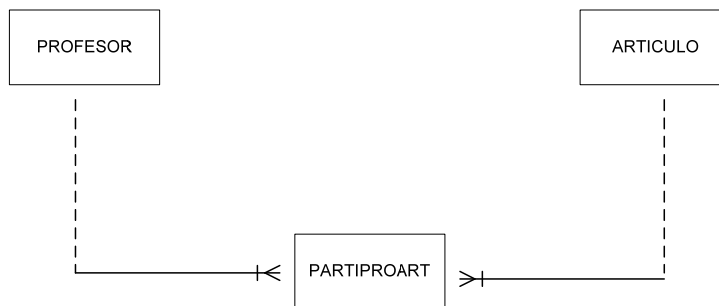


Figura 2.10.- Solución a relacion Profesor-Articulo

De la misma forma, en la figura 2.10 se resuelve el problema de muchos a muchos entre las entidades Profesor – Artículo con la nueva entidad *Partiproart* que entenderemos como Nivel de participación. Un Profesor puede tener uno o más Niveles de participación, un Nivel de participación debe estar ocupado por un Profesor; un Nivel de participación debe tener un Artículo y, un Artículo puede tener uno o más Niveles de participación.

Alumno-Artículo

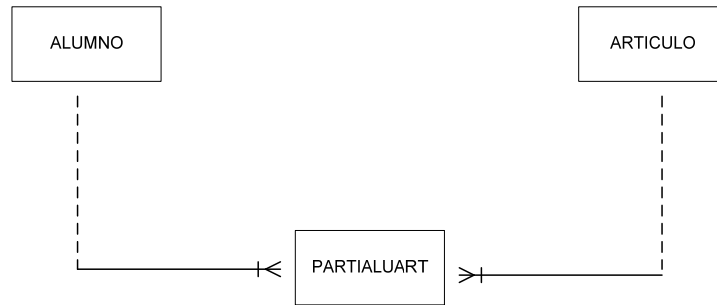


Figura 2.11.- Solución a relacion Alumno-Artículo

La figura 2.11 muestra la solución al problema de muchos a muchos entre las entidades Alumno y Artículo. La nueva entidad *Partialuart* la entenderemos como Nivel de participación. Un Alumno puede tener uno o más Niveles de participación, un Nivel de participación debe estar ocupado por un Alumno; un Nivel de participación debe tener un Artículo y, un Artículo puede tener uno o más Niveles de participación.

Alumno-Proyecto

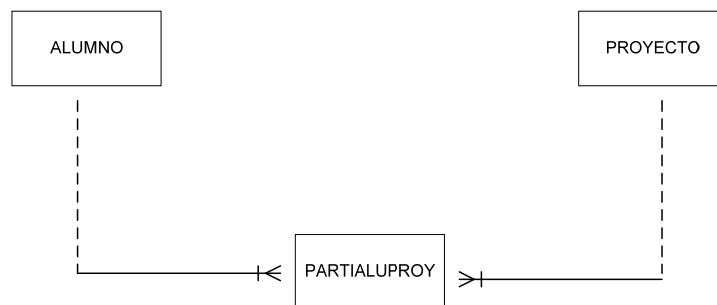


Figura 2.12.- Solución a relacion Alumno-Proyecto

De igual forma, en la figura 2.12 se muestra la solución al problema de la relación de muchos a muchos entre las entidades Alumno y Proyecto. La nueva entidad *PARTIALUPROY* la entenderemos como *Tipo de participación*. Un Alumno puede tener uno o más Tipos de participación, un Tipo de participación debe estar asignada a un Alumno; un Tipo de participación debe pertenecer a un Proyecto y, un Proyecto puede tener uno o más Tipos de participación.

Después de resolver los problemas de las relaciones de muchos a muchos, mostrados

A diferencia de las relaciones que teníamos en la Figura 2.1, ahora encontramos en la figura 2.13 las siguientes:

1. Un profesor **puede emitir** uno o más elementos de información y cada elemento de información **debe ser emitido** por uno y solo un profesor.
2. Un Profesor **puede impartir** en distintos periodos, en un Periodo **debe impartir** un Profesor; en un Periodo **debe ser impartido** un Curso y un Curso **puede ser impartido** en un Periodo.
3. Un Curso **puede sugerir** una o más lecturas, una Lectura **debe ser sugerida** para uno o más cursos, esto a través de la nueva entidad; Curlec.
4. Un profesor **puede tener** asignada una o más actividades (entidad Actiprorec), una actividad **debe ser asignada** a un Profesor; una Actividad **debe generar** un Recurso y un Recurso **puede ser generado** por una o más actividades.
5. Un profesor **puede tener asignada** una o más actividades, una actividad **debe ser asignada** a un Profesor; una Actividad **debe generar** una Herramienta y una Herramienta **puede ser generada** por una o más actividades.
6. Un Profesor **puede tener** uno o más Tipos de participación (entidad Partiprosem), un Tipo de participación **debe estar asignada** a un Profesor; un Tipo de participación **debe pertenecer** a un Seminario y, un Seminario **puede tener** uno o más Tipos de Participación.
7. Un Profesor **puede tener** uno o más Tipos de participación (entidad Partiproproy), un Tipo de participación **debe estar asignado** a un Profesor; un Tipo de participación **debe pertenecer** a un Proyecto y, un Proyecto **puede tener** uno o más Tipos de participación.
8. Un Profesor **puede tener** uno o más niveles de dirección (entidad Niveldireccion), un Nivel de dirección **debe estar ocupado** solo por un profesor; un Nivel de dirección **debe tener** un alumno, y un alumno **puede estar dirigido** en uno o más niveles de dirección (asesor o coasesor).
9. Un Profesor **puede tener** uno o más Niveles de participación (entidad Partiproart), un Nivel de participación **debe estar ocupado** por un Profesor; un Nivel de participación

- debe tener** un Artículo y, un Artículo **puede tener** uno o más Niveles de participación.
10. Un Alumno **puede tener** uno o más Niveles de participación (entidad Partialuart), un Nivel de participación **debe estar ocupado** por un Alumno; un Nivel de participación **debe tener** un Artículo y, un Artículo **puede tener** uno o más Niveles de participación.
11. Un Alumno **puede tener** uno o más Tipos de participación (entidad Partialuprooy), un Tipo de participación **debe estar asignado** a un Alumno; un Tipo de participación **debe pertenecer** a un Proyecto y, un Proyecto **puede tener** uno o más Tipos de participación.

Ahora que ya tenemos nuestro diagrama Entidad-Relación final y descritas las relaciones entre las entidades, también tenemos ya identificadas todas las entidades que necesitamos para de ahí cambiar el diagrama al modelo relacional.

Las entidades que tenemos son las siguientes:

1. PROFESOR.- Información del profesor.
2. ALUMNO.- Información del alumno.
3. NIVELDIRECCION.- El nivel en que dirigirá un profesor a un alumno (asesor o coasesor).
- 4.- INFORMACION.- Información que algún profesor genere, tales como avisos o noticias.
- 5.- CURSO.- Información de cursos que se den dentro del cuerpo académico.
6. PERIODO.- Período en que un curso sea impartido por un profesor.
7. LECTURA.- Lecturas que se sugieran para un curso.
8. CURLEC.- Relaciones entre cursos y lecturas.
9. RECURSO.- Información de los recursos que se podrán utilizar con las herramientas.
10. ACTIPROREC.- La actividad que un profesor tuvo en la generación de un recurso.
11. HERRAMIENTA.- Programas creados en el cuerpo académico y su información.
12. ACTIPROHER.- La actividad que un profesor tuvo en la generación de una herramienta.
13. SEMINARIO.- Seminarios en los que los profesores del cuerpo académico participen.
14. PARTIPROSEM.- Información sobre la participación de un profesor en un seminario.
15. PROYECTO.- Los proyectos activos y terminados dentro del cuerpo académico.

- 16. PARTIPROPROY.- Participación que tienen o tuvieron los profesores en un proyecto.
- 17. ARTICULO.- Artículos que los profesores o alumno hayan firmado y su información.
- 18. PARTIPROART.- Participación de un profesor al hacer un artículo.
- 19. PARTIALUPROY.- Participación de un alumno en un proyecto.
- 20. PARTIALUART.- Participación de un alumno al hacer un artículo.

Ahora veremos qué atributos debe contener cada entidad. Y los representaremos de la siguiente manera:

PROFESOR

- # * id_profesor
- * nombre
- * apellido_pat
- o apellido_mat
- o telefono
- * tel_cel
- * email
- o email_alt
- o website
- o fotografia
- * user
- * psw
- * nivel

ALUMNO

- # * id_alumno
- * nombre
- * nivel
- o direccion
- o telefono
- o tel_cel
- * email
- o email_alt
- o fotografia
- o website
- * status
- o nombre_tesis
- o status_tesis
- o tesis_url
- o observaciones

NIVELDIRECCION

- #* id_profesor
- #* id_alumno
- * tipo
- o comentario

CURSO

- # * id_curso
- * nombre
- * nivel
- o descripcion
- o temario
- o notas

CURLEC

- #* id_curso
- #* id_lectura

RECURSO

- # * id_recurso
- * nombre
- * descripcion
- * enlace_recurso

ACTIPROREC

- #* id_profesor
- #* id_recurso
- * descripcion_activ
- o fecha
- o comentario

INFORMACION

- # * id_informacion
- * descripcion
- * fecha
- * tipo

PERIODO

- #* id_profesor
- #* id_curso
- * periodo
- o fecha_inicio
- o fecha_termino
- * lugar
- o descripción
- o Comentario

LECTURA

- # * id_lectura
- * nombre
- * tipo
- o descripcion
- o referencia

HERRAMIENTA

- # * id_herramienta
- * nombre
- * descripcion
- * enlace_herramienta
- o descripcion_adic

ACTIPROHER

- #* id_profesor
- #* id_herramienta
- * descripcion_activ
- o fecha
- o comentario

SEMINARIO

- # * id_seminario
- * nombre
- * descripcion
- * fecha_inicio
- o fecha_termino

PROYECTO

- # * id_proyecto
- * nombre
- * resumen
- * enlace_proyecto
- * fecha_inicio
- o fecha_termino
- * monto
- o status
- o comentarios

PARTIPROSEM

- #* id_profesor
- #* id_seminario
- * nivel_particip
- o tema
- o fecha
- o comentario

PARTIPROPROY

- #* id_profesor
- #* id_proyecto
- * nivel_particip
- o activ_asig
- o comentario

ARTICULO

- # * id_articulo
- * titulo_paper
- * fecha
- * tipo
- * nombre_pub
- o isbn_issn
- o lugar

PARTIALUPROY

- #* id_alumno
- #* id_proyecto
- * nivel_particip
- o activ_asig
- o comentario

PARTIPROART

- #* id_profesor
- #* id_articulo
- * orden_autoria
- o descripc_particip
- o comentario

PARTIALUART

- #* id_alumno
- #* id_articulo
- * orden_autoria
- o descripc_particip
- o comentario

Entenderemos los signos que se encuentran a la izquierda de los atributos como a continuación se describe:

- # Ese atributo es llave o clave.
- * Ese atributo es mandatorio, es decir, no puede tener valor nulo.
- o Ese atributo es opcional, puede tener valor nulo.

2.3. Diseño del sistema.

El diseño se ha descrito como un proceso multifase en el que se sintetizan representaciones de la estructura de los datos. Es una actividad en el que se toman decisiones importantes frecuentemente de naturaleza estructural. Comparte con la programación un interés por la abstracción de la representación de la información, y de las secuencias de procesamiento [Ingeniería del Software, un enfoque práctico, 5ª Edición, Pressman; Roger, S. McGraw-Hill]

2.3.1. Diseño de Datos.

La estructura de datos ha sido siempre una parte importante del diseño de software. Al nivel de los componentes del programa, el diseño de las estructuras de datos y de los algoritmos asociados requeridos para su manipulación, son la parte esencial en la creación de una aplicación.

Ahora que ya tenemos definidos los objetos de datos (entidades) para nuestro sistema, analizaremos cada uno de ellos en relación de sus atributos, ya que posteriormente debemos crear una base de datos.

PROFESOR				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_profesor	Primary Key	int(5)	not_null	auto_increment
nombre		char(30)	not_null	
apellido_pat		char(25)	not_null	
apellido_mat		char (25)	null	
telefono		char(20)	null	
tel_cel		char(20)	null	

email		char(80)	not_null	
email_alt		char(80)	null	
website		char(80)	null	
fotografia		char(10)	null	
user		char(20)	not_null	
psw		char(20)	not_null	
nivel		int(1)	not_null	

ALUMNO				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_alumno	Primary Key	int(5)	not_null	auto_increment
nombre		char(80)	not_null	
nivel		char(15)	not_null	<i>Licenciatura o Mtría</i>
direccion		char(80)	null	
telefono		char(20)	null	
tel_cel		char(20)	null	
email		char(80)	not_null	
email_alt		char(80)	null	
fotografia		char(10)	null	
website		char(80)	null	
status		char(10)	not_null	<i>Activo, baja, permiso</i>
nombre_tesis		char(130)	null	
status_tesis		char(12)	null	
tesis_url		char(80)	null	
observaciones		text	null	

NIVELDIRECCION				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_profesor	Primary_Key	int(5)	not_null	<i>Prof. que está ligado</i>
id_alumno	Primary_key	int(5)	not_null	<i>Alum. que está ligado</i>
tipo		char(10)	not_null	<i>Asesor o coasesor</i>
comentario		text	null	

INFORMACION				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_informacion	Primary_Key	int(5)	not_null	auto_increment
descripcion		text	not_null	
fecha		date	not_null	
tipo		char(10)	not_null	<i>Aviso, noticia, evento</i>

CURSO				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_curso	Primary_Key	int(5)	not_null	auto_increment
nombre		char(80)	not_null	
nivel		char(15)	not_null	<i>Licenciatura o Mtría</i>
descripcion		text	null	
temario		text	null	
notas		char(80)	null	

PERIODO				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_profesor	Primary_key	int(5)	not_null	<i>Prof. que está ligado</i>

id_curso	Primary_key	int(5)	not_null	<i>Curso que está ligado</i>
periodo		char(10)	not_null	<i>Otoño, verano, prim.</i>
fecha_inicio		date	null	
fecha_termino		date	null	
lugar		char(30)	not_null	
descripción		text	null	
comentario		text	null	

LECTURA				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_lectura	Primary_Key	int(5)	not_null	auto_increment
nombre		char(80)	not_null	
tipo		char(15)	not_null	<i>Artículo, tesis, libro</i>
descripcion		text	null	
referencia		char(80)	null	

CURLEC				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_curso	Primary_key	int(5)	not_null	Curso al que está ligado
id_lectura	Primary_key	int(5)	not_null	Lectura al que está ligado

RECURSO				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_recurso	Primary_Key	int(5)	not_null	auto_increment
nombre		char(50)	not_null	
descripcion		text	not_null	<i>Antología, textos, etc</i>

enlace_recurso		char(80)	not_null	
----------------	--	----------	----------	--

ACTIPROREC				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_profesor	Primary_key	int(5)	not_null	Prof. que está ligado
id_recurso	Primary_key	int(5)	not_null	Recurso ligado
descripcion_activ		char(30)	not_null	
fecha		date	null	
comentario		text	null	

HERRAMIENTA				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_herramienta	Primary_Key	int(5)	not_null	auto_increment
nombre		char(50)	not_null	
descripcion		text	not_null	
enlace_herramienta		char(80)	not_null	<i>Link al programa</i>
descripcion_adic		text	null	

ACTIPROHER				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_profesor	Primary_key	int(5)	not_null	<i>Prof. que está ligado</i>
id_herramienta	Primary_key	int(5)	not_null	<i>Herramienta ligada</i>
descripcion_activ		char(30)	not_null	
fecha		date	null	
comentario		text	null	

SEMINARIO				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_seminario	Primary_Key	int(5)	not_null	auto_increment
nombre		char(80)	not_null	
descripcion		text	not_null	
fecha_inicio		date	not_null	
fecha_termino		date	null	

PARTIPROSEM				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_profesor	Primary_key	int(5)	not_null	<i>Prof. que está ligado</i>
id_seminario	Primary_key	int(5)	not_null	<i>Seminario ligado</i>
nivel_particip		char(15)	not_null	<i>Participante, expositor</i>
tema		char(80)	null	
fecha		date	null	
comentario		text	null	

PROYECTO				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_proyecto	Primary_Key	int(5)	not_null	auto_increment
nombre		char(80)	not_null	
resumen		text	not_null	
enlace_proyecto		char((80)	not_null	
fecha_inicio		date	not_null	

fecha_termino		date	null	
monto		int(7)	not_null	<i>Apoyo monetario</i>
status		char(10)	null	
comentarios		text	null	

PARTIPROPROY				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_profesor	Primary_key	int(5)	not_null	<i>Prof. que está ligado</i>
id_proyecto	Primary_key	int(5)	not_null	<i>Proyecto ligado</i>
nivel_particip		char(15)	not_null	<i>Resp, colaborador</i>
activ_asig		char(30)	null	
Comentario		text	null	

PARTIALUPROY				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_alumno	Primary_key	int(5)	not_null	<i>Alumno ligado</i>
id_proyecto	Primary_key	int(5)	not_null	<i>Proyecto ligado</i>
nivel_particip		char(15)	not_null	<i>Resp, colaborador</i>
activ_asig		char(30)	null	
Comentario		text	null	

ARTICULO				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_articulo	Primary_Key	int(5)	not_null	auto_increment

titulo_paper		char(80)	not_null	
Fecha		date	not_null	
Tipo		char(15)	not_null	<i>Inv, divulg, Journal</i>
nombre_pub		char(80)	not_null	
isbn_issn		char(15)	null	
Lugar		char(30)	null	

PARTIPROART				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_profesor	Primary_key	int(5)	not_null	<i>Prof. que está ligado</i>
id_articulo	Primary_key	int(5)	not_null	<i>Articulo ligado</i>
orden_autoria		char(10)	not_null	<i>Autor, coautor</i>
descrip_particip		char(30)	null	
Comentario		text	null	

PARTIALUART				
<i>Campo</i>	<i>Llave</i>	<i>Tipo</i>	<i>Nulo</i>	<i>Observaciones</i>
id_alumno	Primary_key	int(5)	not_null	<i>Alumno ligado</i>
id_articulo	Primary_key	int(5)	not_null	<i>Articulo ligado</i>
orden_autoria		char(10)	not_null	<i>Autor, coautor</i>
descrip_particip		char(30)	null	
comentario		text	null	

A cada uno de estos objetos de datos o entidades, desde el punto de vista de manejo de bases de datos también les llamaremos **Tablas**.

Como se muestra en la Figura 2.14, en el marco superior irá un encabezado, en el izquierdo un menú donde seleccionar lo que se desee consultar, en el derecho se desplegará la información que vayamos seleccionando en el menú. Los marcos superior, izquierdo e inferior no cambiarán su contenido.

Cuando demos clic en algún elemento del menú nos desplegará la información correspondiente en el marco derecho, por ejemplo cuando demos clic en el elemento *Información* nos desplegará algo como lo que se muestra en la Figura 2.15.

Facultad de Ciencias de la Computación Logo FCC Cuerpo Académico de Bases de Datos y Rec. De Inform. Logo BUAP Recursos Lingüísticos de Recup. De Inf.		
Aquí va el menú: Inicio Información Profesores Alumnos etc.	Información Avisos Elemento de aviso1 Elemento de aviso2 Noticias Noticia 1 Noticia2 Eventos Evento1 Evento2	
Admin	-Laboratorio de Recup. De Inf.	Enviar e-mail.

Figura 2.15.- Pantalla que muestra la sección de consulta Información.

De igual forma, al dar clic en *Profesores* nos desplegará algo como lo mostrado en la

Figura 2.16

Facultad de Ciencias de la Computación Logo FCC Cuerpo Académico de Bases de Datos y Rec. De Inform. Logo BUAP Recursos Lingüísticos de Recup. De Inf.		
Aquí va el menú: Inicio Información Profesores Alumnos etc.	<p>Profesores</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Fotografía Profesor1 </div> Nombre Profesor1 Dirección Telefono e-mail Etc.	
	...	
	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Fotografia ProfesorN </div> Nombre ProfesorN Dirección Telefono e-mail Etc.	
Admin	-Laboratorio de Recup. De Inf.	Enviar e-mail.

Figura 2.16.- Pantalla que muestra la sección de consulta Profesores.

Y así sucesivamente para todas las tablas que contenga nuestra base de datos.

Ahora, nos centraremos en la parte de administración del sitio, ya que aquí es donde daremos de alta toda la información que será consultada a través de la web; para eso habrá que “loguearse” en el link de la esquina inferior izquierda: “Admin”, y nos pedirá un nombre de usuario y contraseña, como se ve en la Figura 2.17

Facultad de Ciencias de la Computación Logo FCC Cuerpo Académico de Bases de Datos y Rec. De Inform. Logo BUAP Recursos Lingüísticos de Recup. De Inf.		
Administrar: Información Profesores Alumnos etc.	Administración del Sitio Usuario: <input type="text"/> Contraseña: <input type="password"/> <input type="button" value="Entrar"/>	
Salir	-Laboratorio de Recup. De Inf.	Enviar e-mail.

Figura 2.17. Pantalla de autenticación para administrar el sitio.

Si el usuario y contraseña son correctos, nos mostrará la pantalla de inicio a la Administración del sitio (Figura 2.18), desde el cual podremos acceder al menú del lado izquierdo y seleccionar ahí qué queremos administrar. Si los datos ingresados son incorrectos no regresará a la misma página de autenticación.

Facultad de Ciencias de la Computación Logo FCC Cuerpo Académico de Bases de Datos y Rec. De Inform. Logo BUAP Recursos Lingüísticos de Recup. De Inf.	
Administrar: Información Profesores Alumnos etc.	Administración del sitio. (Fotografía del Edificio de la FCC) Seleccione en el menú de la izquierda lo que desea administrar.
Salir	-Laboratorio de Recup. De Inf. Enviar e-mail.

Figura 2.18. Pantalla de administración del sitio.

Por simplicidad en la explicación de cada una de las siguientes pantallas. Lo que se muestra a partir de ahora se mostrará en el recuadro (frame) de la derecha. El resto seguirá igual.

Una vez seleccionado una opción del menú, mostrará una lista de los elementos que se encuentran en la tabla correspondiente en la base datos (Figura 2.19).

Administración de (NombreDeTabla)

	<i>Campo1</i>	<i>Campo2</i>	...	<i>CampoN</i>
<input type="checkbox"/>	Dato11	Dato12	...	Dato1n
<input type="checkbox"/>	Dato21	Dato22	...	Dato2n
<input type="checkbox"/>
<input type="checkbox"/>	Daton1	Daton2	...	Datonn

Figura 2.19.- Pantalla de administración de tablas.

Como podremos darnos cuenta, en la primera columna tendrá un elemento de selección (checkbox), esto para facilitar la selección de elementos con el ratón cuando se quiera eliminar o modificar una o más entradas de la lista; es decir, suponiendo que queremos eliminar el elemento 1 y 2 de la lista, solo bastará con seleccionarlos con el mouse dando clic en el checkbox y presionando el botón *Eliminar*. De igual forma sería para modificar algún elemento de la lista.

En caso de que se desee agregar un nuevo elemento a la tabla correspondiente tiene que hacerse dando clic y nos presentará una pantalla en la Figura 2.20.

Agregar nuevo (NombreDeTabla)

Dato1

Dato2

...

DatoN

Figura 2.20.- Pantalla de agregar elementos nuevos.

Al presionar el botón *Guardar* de la figura 2.20, almacenará los datos introducidos en la base de datos y nos regresará a la misma pantalla para agregar más elementos. Si presionamos el botón *Reset*, los datos que hayamos introducido antes de Guardar se borrarán y podremos empezar a capturar nuevamente. Con el botón *Terminar* dejamos de ingresar más elementos y nos regresará a la pantalla de la Figura 2.19.

En caso de presionar el botón *Modificar* de la Figura 2.19, si no hay ningún elemento seleccionado nos regresará a la misma pantalla. Si hay uno o más elementos seleccionados, nos presentará una a una cada pantalla (Figura 2.21) para la modificación de cada elemento.

Modificar (NombreDeTabla)

Dato1

Dato2

...

DatoN

Figura 2.21.- Pantalla de Modificar elementos.

Al presionar el botón *Guardar* debe actualizarse dicha información. Si hay más elementos a modificar entonces se muestra esta misma pantalla hasta que todos los elementos seleccionados hayan sido actualizados. En caso de que solo haya sido seleccionado este elemento o éste sea el último en modificar, nos regresa a la pantalla de la Figura 2.19. Si ya no deseamos modificar la entrada que tenemos en pantalla presionaremos el botón *Cancelar ésta y las restantes modificaciones* y nos regresará a la pantalla de la Figura 2.19.

Si en la pantalla de la Figura 2.19 damos clic en el botón *Eliminar* y no hay elementos seleccionados nos regresa a la misma pantalla si hay uno o más seleccionados, entonces se eliminan en la base de datos las entradas seleccionadas y nos regresa a la misma pantalla de la Figura 2.19.

Al terminar de administrar la información correspondiente se deberá dar clic en “Salir” de la pantalla de la Figura 2.18, de esta manera nos regresará al área de consulta de la Figura 2.14.

2.3.3 Diagrama Funcional del Sistema

De acuerdo al diseño de interfaz que analizamos en la sección anterior, el sistema funcionará en nuestro navegador de acuerdo al Diagrama Funcional que se muestra en la Figura 2.22.

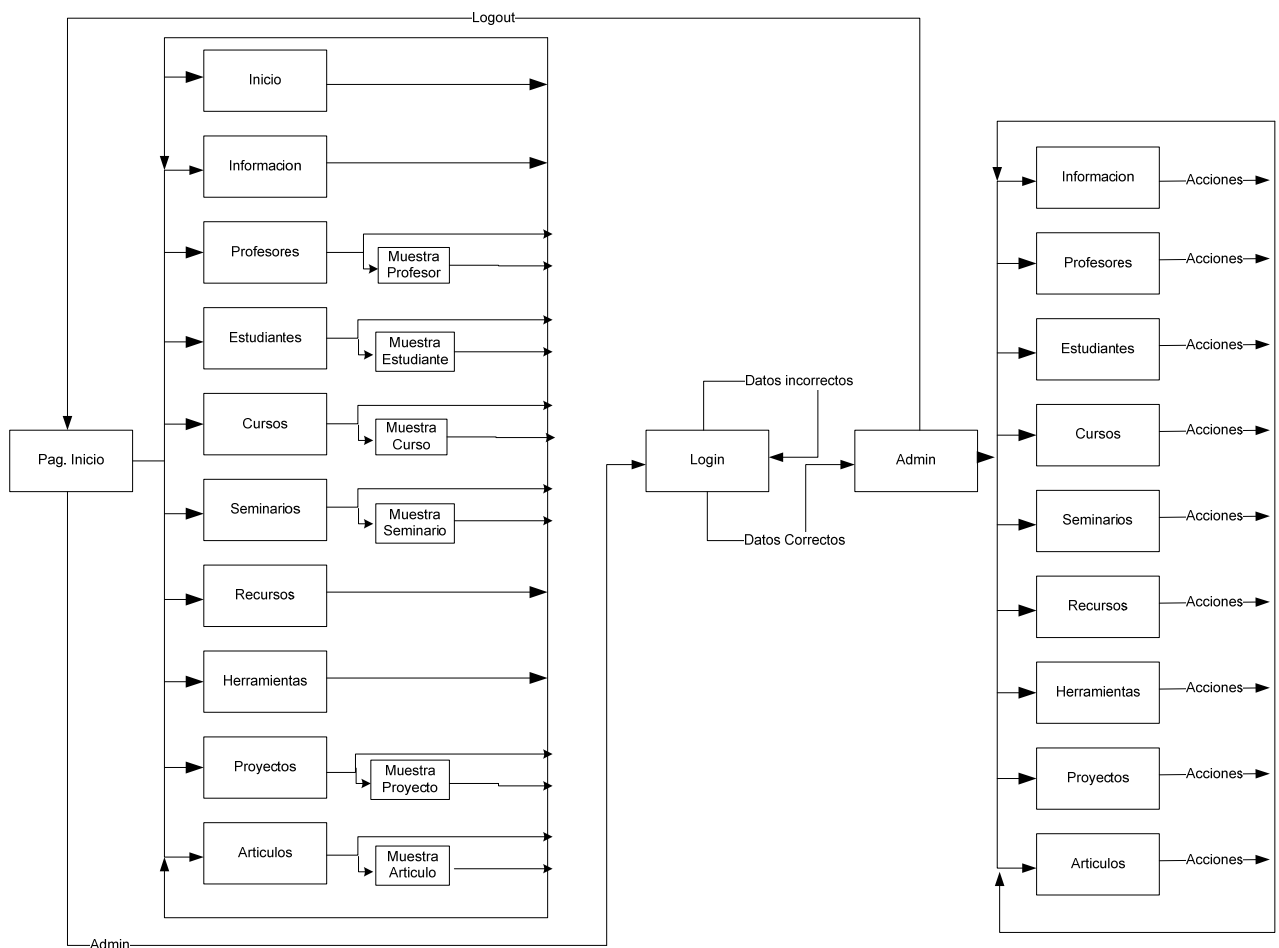


Figura 2.22.- Diagrama Funcional General

La primera parte (lado izquierdo) corresponde al área de consultas y es bastante sencilla, partimos de la página de inicio, y de ahí a cualquier consulta de las diferentes tablas de nuestra BD. La segunda parte (lado derecho) corresponde al área de administración. Para entrar a esta área a través del link “Admin”, deberemos “loguearnos” y una vez entrando, podremos administrar las distintas tablas de nuestra BD. Para cada tabla en el área de administración hay una línea que dice “Acciones”. Estas acciones se definen más adelante.

Cuando seleccionemos una tabla nos mostrará los elementos que se encuentran en la tabla correspondiente de nuestra BD, y tendremos 3 acciones a ejecutar: Nuevo, Modificar y Eliminar. Para esto tomaremos como ejemplo la tabla profesor. Para el resto de tablas deberán ser análogos. A continuación, los Diagramas Funcionales correspondientes.

Nuevo Profesor.

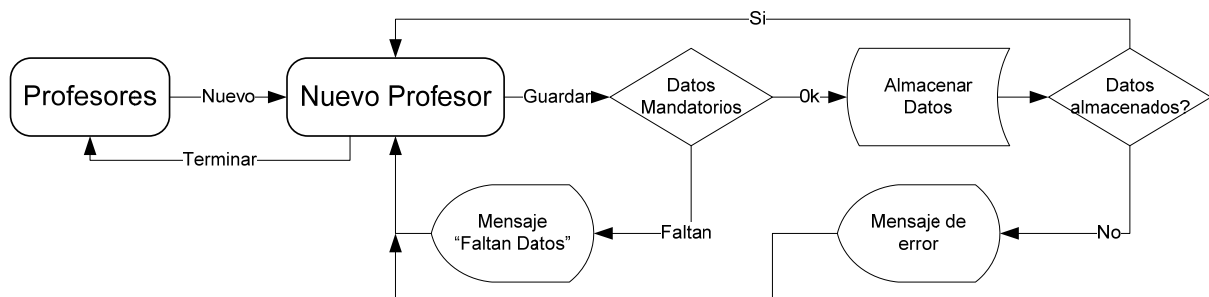


Figura 2.23.- Diagrama Funcional - Nuevo Profesor

Modificar Profesores.

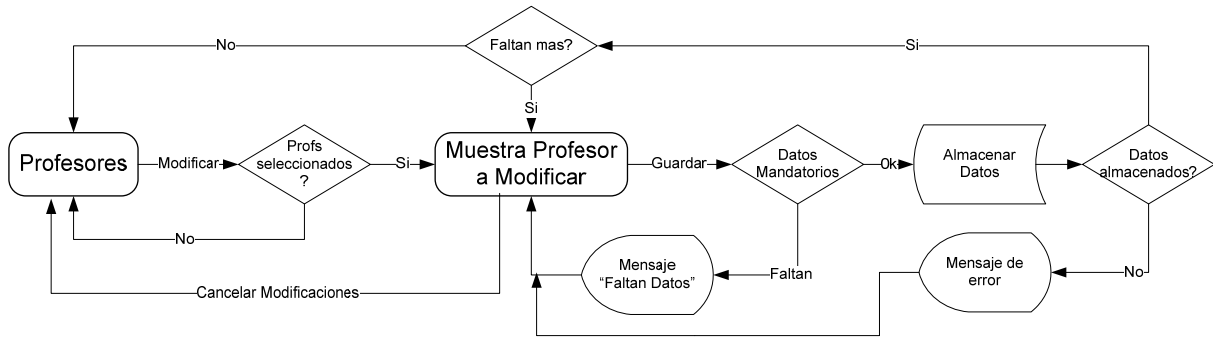


Figura 2.24.- Diagrama Funcional - Modificar Profesores

Eliminar

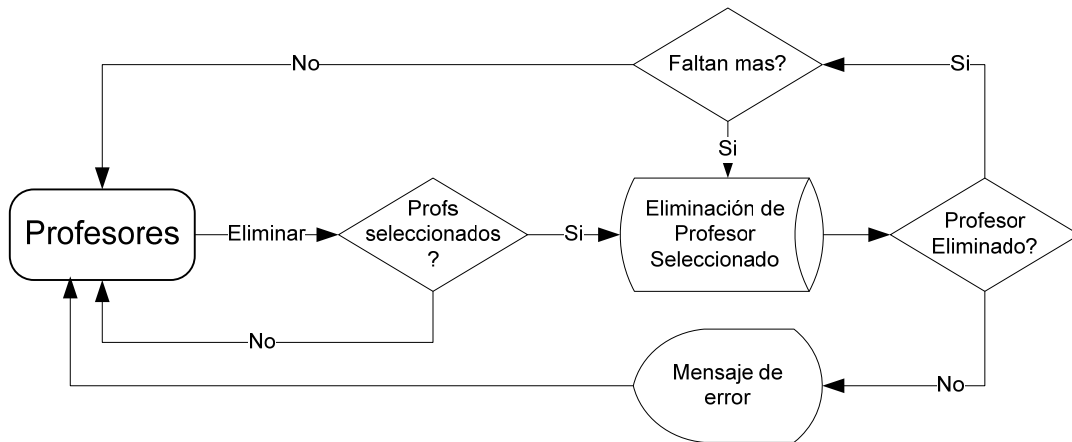


Figura 2.25.- Diagrama Funcional - Eliminar Profesores

CAPITULO III.

DESARROLLO.

El desarrollo de sistemas de software implica una serie de actividades de producción en las que las posibilidades de que aparezca el fallo humano son enormes. Los errores pueden empezar a darse desde el primer momento del proceso, en el que los objetivos pueden estar especificados de forma errónea o imperfecta, así como en posteriores pasos de diseño y desarrollo [*Ingeniería de Software Explicada*, Norris, Mark; Rigby Peter, Megabyte-Noriega Editores. 1994]

3.1. Desarrollo e Implementación del sistema.

La implementación de nuestro sistema será dividida en dos partes. La primera será la creación de nuestra base de datos para posteriormente, usando los lenguajes de programación PHP y HTML, crear los archivos que serán parte de nuestro sistema en web. Cabe hacer mención que también se ocuparán algunas funciones ya definidas de Javascript.

El sistema será implementado en el servidor *yoliztli.cs.buap.mx* con sistema operativo linux y servicios de páginas web Apache, php, y mysql.

A continuación procederemos a crear la base datos haciendo uso de SQL, que es el lenguaje de definición de datos de Mysql.

Particularmente, vamos a suponer el nombre “*datasite*” para la base de datos que usará nuestro sistema. Y procedemos a crearlo de la siguiente forma:

```
CREATE DATABASE datasite;
```

Aquí debemos presuponer que al ejecutar esta instrucción (y las que agregaremos más adelante), ya hemos abierto una terminal de mysql con un nombre de usuario de administrador.

Después procedemos a otorgar permisos a un usuario 'usuario' para poder realizar consultas en la base de datos:

```
GRANT select ON sitedb.* TO 'usuario@localhost' IDENTIFIED BY 'passwordDeUsuario';
```

Además, otorgamos permisos administrativos a un usuario 'administrador' que nos permitirán la administración del site.

```
GRANT all privileges ON sitedb.* TO 'administrador@localhost' IDENTIFIED BY 'passwordDeMyadmin';
```

Y creamos las tablas que contendrá la base da datos como sigue:

profesor

```
create table profesor (id_profesor mediumint(5) primary key not null auto_increment, nombre varchar(30) not null, apellido_pat varchar(25) not null, apellido_mat varchar(25), telefono varchar(20), tel_cel varchar(20), email varchar(80) not null, email_alt varchar(80), website varchar(80), fotografia varchar(10), user varchar(20) not null, psw varchar(20) not null, nivel int(1) not null) engine=innodb;
```

alumno

```
create table alumno (id_alumno mediumint primary key not null auto_increment, nombre varchar(80) not null, nivel varchar(15) not null, direccion varchar(80), telefono varchar(20), tel_cel varchar(20), email varchar(80) not null, email_alt varchar(80), fotografia varchar(10), website varchar(80), status varchar(10) not null, nombre_tesis varchar(130), status_tesis varchar(12), tesis_url varchar(80), observaciones text) engine=innodb;
```

niveldireccion

```
create table niveldireccion (id_profesor mediumint not null, id_alumno mediumint not null, tipo varchar(10) not null, comentario text, primary key (id_profesor,id_alumno), index(id_profesor),index (id_alumno), foreign key (id_profesor) references profesor(id_profesor) on delete cascade, foreign key (id_alumno) references alumno(id_alumno) on delete cascade) engine=innodb;
```

informacion

create table informacion (id_informacion mediumint primary key not null auto_increment, descripcion text not null, fecha date not null, tipo varchar(10) not null) engine=innodb;

curso

create table curso (id_curso mediumint primary key not null auto_increment, nombre varchar(80) not null, nivel varchar(15) not null, descripcion text, temario text, notas varchar(80)) engine=innodb;

periodo

create table periodo (id_profesor mediumint not null, id_curso mediumint not null, periodo varchar(10) not null, fecha_inicio date, fecha_termino date, lugar varchar(30) not null, descripcion text, comentario text, primary key (id_profesor, id_curso), index (id_profesor), index (id_curso), foreign key (id_profesor) references profesor(id_profesor) on delete cascade, foreign key (id_curso) references curso(id_curso) on delete cascade) engine=innodb;

lectura

create table lectura (id_lectura mediumint primary key not null auto_increment, nombre varchar(80) not null, tipo varchar(15) not null, descripcion text, referencia varchar(80)) engine=innodb;

curlec

create table curlec (id_curso mediumint not null, id_lectura mediumint not null, primary key (id_curso, id_lectura), index (id_curso), index (id_lectura), foreign key (id_curso) references curso(id_curso) on delete cascade, foreign key (id_lectura) references lectura(id_lectura) on delete cascade) engine=innodb;

recurso

create table recurso (id_recurso mediumint primary key not null auto_increment, nombre varchar(50) not null, descripcion text not null, enlace_recurso varchar(80) not null) engine=innodb;

actiprorec

create table actiprorec (id_profesor mediumint not null, id_recurso mediumint not null, descripcion_activ varchar(30) not null, fecha date, comentario text, primary key (id_profesor, id_recurso), index (id_profesor), index (id_recurso), foreign key (id_profesor) references profesor(id_profesor) on delete cascade, foreign key (id_recurso) references recurso(id_recurso) on delete cascade) engine=innodb;

herramienta

create table herramienta (id_herramienta mediumint primary key not null auto_increment, nombre varchar(50) not null, descripcion text not null, enlace_herramienta varchar(80) not null, descripcion_adic text) engine=innodb;

actiproher

create table actiproher (id_profesor mediumint not null, id_herramienta mediumint not null, descripcion_activ varchar(30) not null, fecha date, comentario text, primary key (id_profesor, id_herramienta), index (id_profesor), index (id_herramienta), foreign key (id_profesor) references profesor(id_profesor) on delete cascade, foreign key (id_herramienta) references herramienta(id_herramienta) on delete cascade) engine=innodb;

seminario

create table seminario (id_seminario mediumint primary key not null auto_increment, nombre varchar(80) not null, descripcion text not null, fecha_inicio date not null, fecha_termino date) engine=innodb;

partiprosem

create table partiprosem (id_profesor mediumint not null, id_seminario mediumint not null, nivel_particip varchar(15) not null, tema varchar(80), fecha date, comentario text, primary key (id_profesor, id_seminario), index (id_profesor), index (id_seminario), foreign key (id_profesor) references profesor(id_profesor) on delete cascade, foreign key (id_seminario) references seminario(id_seminario) on delete cascade) engine=innodb;

proyecto

create table proyecto (id_proyecto mediumint primary key not null auto_increment, nombre varchar(80) not null, resumen text not null, enlace_proyecto varchar(80) not null, fecha_inicio date not null, fecha_termino date, monto mediumint, status varchar(10), comentarios text) engine=innodb;

partiproproy

create table partiproproy (id_profesor mediumint not null, id_proyecto mediumint not null, nivel_particip varchar(15) not null, activ_asig varchar(30), comentario text, primary key (id_profesor, id_proyecto), index (id_profesor), index (id_proyecto), foreign key (id_profesor) references profesor(id_profesor) on delete cascade, foreign key (id_proyecto) references proyecto(id_proyecto) on delete cascade) engine=innodb;

articulo

```
create table articulo (id_articulo mediumint primary key not null auto_increment, titulo_paper varchar(80) not null, fecha date not null, tipo varchar(15) not null, nombre_pub varchar(80) not null, isbn_issn varchar(15), lugar varchar(30)) engine=innodb;
```

partiproart

```
create table partiproart (id_profesor mediumint not null, id_articulo mediumint not null, orden_autoria varchar(10) not null, descripc_particip varchar(30), comentario text, primary key (id_profesor,id_articulo), index (id_profesor), index (id_articulo), foreign key (id_profesor) references profesor(id_profesor) on delete cascade, foreign key (id_articulo) references articulo(id_articulo) on delete cascade) engine=innodb;
```

partialupro

```
create table partialupro (id_alumno mediumint not null, id_proyecto mediumint not null, nivel_particip varchar(15) not null, activ_asig varchar(30), comentario text, primary key (id_alumno,id_proyecto), index (id_alumno), index (id_proyecto), foreign key (id_alumno) references alumno(id_alumno) on delete cascade, foreign key (id_proyecto) references proyecto(id_proyecto) on delete cascade) engine=innodb;
```

partialuart

```
create table partialuart (id_alumno mediumint not null, id_articulo mediumint not null, orden_autoria varchar(10) not null, descripc_particip varchar(30), comentario text, primary key (id_alumno,id_articulo), index (id_alumno), index (id_articulo), foreign key (id_alumno) references alumno(id_alumno) on delete cascade, foreign key (id_articulo) references articulo(id_articulo) on delete cascade) engine=innodb;
```

Resultando así, una base de datos de nombre *sitedb* con veinte tablas, un usuario *usuario* con permisos de consulta y, un usuario *myadmin* con permisos administrativos.

Ahora procedemos a la creación de los archivos que serán parte de nuestro sistema, como se mencionó antes, usaremos php como lenguaje por defecto en la programación de nuestras páginas.

El archivo inicial de nuestro sistema, que abrirá el navegador al solicitar la página será *index.php* que creamos con el siguiente código:

```

<HTML>
<HEAD>
  <TITLE>Yoliztli :: Cuerpo Académico de Bases de Datos y Recuperación de
  Información</TITLE>
</HEAD>

<FRAMESET rows="120,*,65" FRAMEBORDER="0">
  <FRAME SRC="up.php" NAME="frup" SCROLLING="NO" noresize>
  <FRAMESET cols="115,*" FRAMEBORDER="0">
    <FRAME SRC="left.php" NAME="frleft" SCROLLING="no" noresize>
    <FRAME SRC="start.php" NAME="frright" SCROLLING="AUTO" MARGINWIDTH="15"
    MARGINHEIGHT="15">
  </FRAMESET>
  <FRAME SRC="down.php" NAME="frdown" SCROLLING=NO noresize>
</FRAMESET>
</HTML>

```

Código: *index.php*

Como podemos ver, lo único que contiene son las definiciones de los marcos (frames) en que estará dividida la pantalla. En la definición de cada marco se incluye también el nombre del archivo que se cargará en cada marco al abrir la página.

Así que, creamos también los archivos que se cargarán en cada marco. En el marco superior abrirá el archivo *up.php* que contendrá el encabezado.

```

<html>
<BODY bgcolor="#bdbdbd">
  <TABLE width="100%" cellspacing="0" border="0" cellpadding="0"
  bgcolor="#bdbdbd">
    <TR>
      <TD WIDTH="100" ALIGN="CENTER">
        <A href="http://www.cs.buap.mx" target="_blank">
          <IMG SRC="images/logofcc.png" WIDTH="68" HEIGHT="87" BORDER="0">
        </A>
      </TD>
      <TD align="CENTER" valign="top">
        <FONT COLOR="Navy">
          <h3>Facultad de Ciencias de la Computación <br>
          Cuerpo Académico de Bases de Datos y
          Recuperación de Información</h3>
          <B>Recursos Lingüísticos de la Recuperación de
          Información</b>
        </FONT>
      </TD>
      <TD WIDTH="100" ALIGN="CENTER">
        <A href="http://www.buap.mx/" target="_blank">
          <IMG SRC="images/buap.gif" WIDTH="60" HEIGHT="87" BORDER="0">
        </A>
      </TD>
    </TR>
  </TABLE>

```

```

        </TD>
    </TR>
</TABLE>
</BODY>
</html>

```

Código: *up.php*

El marco inferior contiene el pié de nuestra página y el código se encuentra en *down.php*.

```

<HTML>
<style type="text/css">
a:link
{ color: navy; text-decoration: none }
a:active
{ color: navy; text-decoration: none }
a:visited
{ color: navy; text-decoration: none }
a:hover
{ color: steelblue; text-decoration: none }
</style>
<BODY bgcolor="#bdbdbd" text="navy">
  <TABLE WIDTH="100%" BORDER="0" CELSPACING="0" CELLPADDING="0">
    <TR>
      <TD WIDTH="100" ALIGN="CENTER">
        <A HREF="admin/" target="_top"> <IMG SRC="images/admin.gif" WIDTH="25"
HEIGHT="25" BORDER="0">Admin</A>
      </TD>
      <TD ALIGN="CENTER">
        <B>--- Laboratorio de Recuperaci&oacute;n de Informaci&oacute;n, FCC-BUAP ---
</B>
      </TD>
      <TD WIDTH="100" ALIGN="RIGHT">
        <A HREF="mailto:webmaster@yoliztli.cs.buap.mx"> <b>e-mail</b>
        <IMG SRC="images/email.gif" WIDTH="25" HEIGHT="25" BORDER="0"> </A>
      </TD>
    </TR>
  </TABLE>
  <HR COLOR="NAVY">
</BODY>
</HTML>

```

Código: *down.php*

El marco izquierdo contendrá el menú a través del cual se podrá consultar la información y el código se encuentra en *left.php*

```

<html>

```

```

<HEAD>
<style type="text/css">
#menu {
    border: 1px solid #DBEBF6;
    border-width: 1px 0 0 0;
    width: 140px;
    font: 80% "Trebuchet MS", Arial, Helvetica, sans-serif;
}
#menu ul, li {
    list-style-type:none;
}

#menu ul {
    margin: 0;
    padding: 0;
}

#menu li {
    border-bottom: 1px solid #DBEBF6;
    text-indent:none;
}

#menu a {
    text-decoration: none;
    color: navy;
    display:block;
    padding: 6px 0px;
    width: 140px;
}

#menu a:hover {
    color:#3366CC; background: #DBEBF6;
    font-style:italic;
}

</style>

</HEAD>
<BODY background="images/menuback.jpg">
<BR><br>

<div id="menu">
<ul type="none">
<LI><A HREF="start.php" target="frright"><IMG SRC="images/inicio.gif" WIDTH="16"
HEIGHT="16" BORDER="0"> Inicio</A></LI><LI class="current">
<LI><A HREF="info.php" target="frright"><IMG SRC="images/info1.gif" WIDTH="16"
HEIGHT="16" BORDER="0"> Informaci&oacute;n</A></LI>
<LI><A HREF="profesores.php" target="frright"><IMG SRC="images/profs.gif"
WIDTH="16" HEIGHT="16" BORDER="0"> Profesores</A></LI>
<LI><A HREF="alumnos.php" target="frright"><IMG SRC="images/alums.gif"
WIDTH="16" HEIGHT="16" BORDER="0"> Estudiantes</A></LI>
<LI><A HREF="cursos.php" target="frright"><IMG SRC="images/cursos.gif" WIDTH="16"
HEIGHT="16" BORDER="0"> Cursos</A></LI>
<LI><A HREF="seminarios.php" target="frright"><IMG SRC="images/seminarios.gif"
WIDTH="16" HEIGHT="16" BORDER="0"> Seminarios</A></LI>

```

```

</LI><A HREF="recursos.php" target="frright"><IMG SRC="images/recursos.gif"
WIDTH="16" HEIGHT="16" BORDER="0"> Recursos</A></LI>
<LI><A HREF="herramientas.php" target="frright"><IMG SRC="images/herramientas.gif"
WIDTH="16" HEIGHT="16" BORDER="0"> Herramientas</A></LI>
<LI><A HREF="proyectos.php" target="frright"><IMG SRC="images/proyectos.gif"
WIDTH="16" HEIGHT="16" BORDER="0"> Proyectos</A></LI>
<LI><A HREF="articulos.php" target="frright"><IMG SRC="images/articulos.gif"
WIDTH="16" HEIGHT="16" BORDER="0"> Articulos</A></LI>
</ul>
</div>
</BODY>
</html>

```

Código: *left.php*

Y en el marco derecho, se cargará inicialmente el archivo *start.php*, como ya se había mencionado, este marco es en el que se mostrará toda la información a consultar y, como veremos mas adelante, la misma en la que podremos realizar las actividades de administración de las tablas de la base de datos.

```

<html>
<HEAD>
  <LINK rel="stylesheet" href="estilo.css" type="text/css">
</HEAD>
<BODY>
  <H3>Bienvenido a Yoliztli</H3><br>
  <center><IMG src="images/fcc1.jpg" align="middle" border="0"></center>
  <br><br>
  El site del Cuerpo Academico de Bases de Datos y Recuperacion de Informacion en la Facultad
  de Ciencias de la Computacion, BUAP.
</BODY>
</html>

```

Código: *start.php*

Además, aplicamos hojas de estilo CSS (*Cascading Style Sheets*)

```

body { font-family: Georgia, "Times New Roman", Times, serif;
       color: black;
       background-color:#CEE3F6; line-height:1.5;
       }

h1 { color:green }
h2 { color:orange }
h3 { color:#610B0B; text-align:center;
     border-width:1px; border-style:double; border-color:white;
     background-image:url("./images/bkt2.gif"); background-repeat:repeat-x;

```

```

    font-size-adjust:.6; font-family:monospace;
}
h4 { color:navy; font-size-adjust:.7; font-family:monospace; }
h5 { color:orange }

img { border-color:gray; border-style:inset; border-width:medium }
img.nob { border-style:none }

a { text-decoration:none; color:black}
a.sub { text-decoration:underline; }

ul { list-style-type:square; line-height:1.5; }

table { background-color:#E0ECF8 }
table.tablborder { width:100%; border-width:medium; border-style:double; border-color:white; }

hr { color:white; }

```

Código: *estilo.css*

De esta forma, obtenemos nuestra página inicial como se muestra en la Figura 3.1



Fig. 3.1.- *Página de inicio de nuestro sistema.*

Análogamente, creamos el área de administración con la misma estructura de frames (marcos) definidos previamente para al área de consultas. Así, obtenemos la pantalla de administración como se muestra en la Figura 3.2.



Fig. 3.2.- Página de Administración del sitio.

Para poder acceder al área de administración, tendremos que autenticarnos dando clic en la liga “Admin” de la figura 3.1. Para eso, generamos una pantalla donde ingresar un nombre de usuario y contraseña como se ve en la pantalla de la Figura 3.3; otro módulo se encargará de verificar que los datos sean correctos, de ser así, verificaremos qué tipo de usuario es: “Administrador” o “Colaborador”; en el primer caso nos dará acceso a la pantalla de la Figura 3.2. Si el usuario y contraseña corresponden a un Colaborador, el usuario solo podrá cambiar su propia información. En caso de que el usuario y contraseña no correspondan, entonces regresará a la misma pantalla de autenticación.

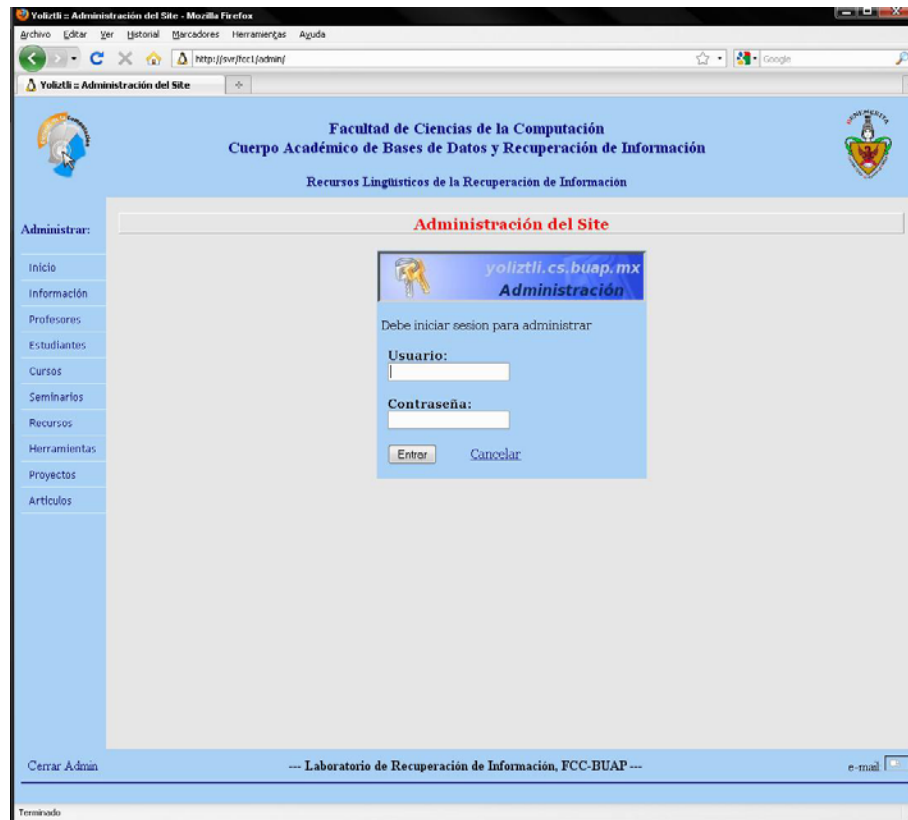


Figura 3.3.- Pantalla de autenticación.

Al seleccionar un elemento a administrar en la pantalla de la figura 3.2, nos mostrará una lista con los elementos de la tabla relacionada. Cada elemento tendrá un checkbox (cuadro de selección). Así mismo, en la parte inferior, al final de la lista nos mostrará los botones de “Nuevo”, “Modificar” y “Eliminar” que son las acciones que se podrán ejecutar sobre los elementos de la lista.

Ahora veamos la programación del área de administración, por el momento nos centraremos en la administración de profesores. Para mostrar la lista de los elementos guardados en la tabla Profesores escribimos el siguiente código:

```

<?
  session_start();
  if(!session_is_registered('administrador'))
  {
    header ("Location: ../login.html");
    exit;
  }

```

```

}
$var="";
?>
<HTML>
<HEAD>
  <LINK href="../estiloadmin.css" rel="stylesheet" type="text/css">
</HEAD>

<BODY>
  <h3>Administraci&oacute;n de Profesores</h3>
  <br>
  <FORM ACTION="accionprofesor.php" METHOD="POST">
    <TABLE BORDER="1">
      <TR>
        <TD> </TD>
        <TD><CENTER><B>Nombre</B></CENTER></TD>
        <TD><CENTER><B>Ap. Paterno</B></CENTER></TD>
        <TD><CENTER><B>Ap. Materno</B></CENTER></TD>
        <TD><CENTER><B>Tel&eacute;fono</B></CENTER></TD>
        <TD><CENTER><B>Tel. Celular</B></CENTER></TD>
        <TD><CENTER><B>e-mail</B></CENTER></TD>
        <TD><CENTER><B>e-mail alt.</B></CENTER></TD>
        <TD><CENTER><B>Website</B></CENTER></TD>
        <TD><CENTER><B>Fotografia</B></CENTER></TD>
        <TD><CENTER><B>Username</B></CENTER></TD>
        <TD><CENTER><B>Password</B></CENTER></TD>
        <TD><CENTER><B>Nivel*</B></CENTER></TD>
      </TR>

<?
  $myhost="localhost";
  $myuser="administrador";
  $mypsw="datasite";

  if($dblink=mysql_connect($myhost,$myuser,$mypsw))
  {
    if(!mysql_select_db("sitedb",$dblink))
    {
      echo "Error al conectarse a la base de datos";
      exit();
    }
    $query="select * from profesor order by id_profesor";
    if($result=mysql_query($query,$dblink))
    {

      $filas=mysql_num_rows($result);
      echo "$filas Registros encontrados. <br/>";

      while($r=mysql_fetch_row($result))
      {
?> <TR>

        <TD><INPUT TYPE="checkbox" NAME="chb[]" VALUE="<? echo "$r[0]";?>">

```

```

</TD>
    <TD><?echo "$r[1]";?></TD>
    <TD><?echo "$r[2]";?></TD>
    <TD><?echo "$r[3]";?></TD>
    <TD><?echo "$r[4]";?></TD>
    <TD><?echo "$r[5]";?></TD>
    <TD><?echo "$r[6]";?></TD>
    <TD><?echo "$r[7]";?></TD>
    <TD><A
        HREF="<?echo "$r[8]";?>"
        target="_blank"><?echo
"$r[8]";?></A></TD>
    <TD><?echo "$r[9]";?></TD>
    <TD><?echo "$r[10]";?></TD>
    <TD><?echo "$r[11]";?></TD>
    <TD><?echo "$r[12]";?></TD>
</TR>
<? } ?>
</TABLE>

    *Nivel 1=Participante 2=Administrador<br><br>
    <INPUT TYPE="submit" NAME="new" VALUE="Nuevo...">
    <INPUT TYPE="submit" NAME="upd" VALUE="Modificar">
    <INPUT TYPE="submit" NAME="del" VALUE="Eliminar">
<br><br>
<A href=" ../alumnos/niveldirecciones.php">Direccion de Profesores a Alumnos</A>

<? }
    else echo "Error al consultar la bd";
}
else
{
    print("Error al conectarse al servidor");
}

?>

</BODY>
</HTML>

```

Código: profesores.php

El código a ejecutar después de presionar un botón de acción (tales como Nuevo, Modificar o Eliminar) para los profesores seleccionados de la lista se muestra a continuación.

```

<?
    session_start();
    if(!session_is_registered('administrador'))
    {
        header ("Location: ../login.html");
        exit;
    }

```



```

<INPUT TYPE="hidden" name="faltan" value="<?echo $faltan;?>">
<INPUT TYPE="hidden" name="i" value="<?echo $i?>">
<? $temp = serialize($chb);?>
<INPUT TYPE="hidden" name="tempchb" value='<?echo $temp;?>'>

<INPUT TYPE="submit" NAME="guardar" VALUE="Guardar">
<INPUT TYPE="reset" NAME="reset" VALUE="Reset">
</FORM>

<FORM ACTION="profesores.php">
  <INPUT TYPE="submit" NAME="cancelar" VALUE="Cancelar esta y las restantes
modificaciones">
</FORM>

</BODY>
</HTML>

<?
    }
    else echo "Error al consultar el registro";
  }
}
else
if (isset($del))
{
  //$chb = $HTTP_POST_VARS["chb"];
  if (($regsel=count($chb)) < 1)
  {
    header ("Location: profesores.php");
    //echo "no hay elementos seleccionados"
    exit;
  }

  $success=false;
  for ($i=0; $i<count($chb); $i++)
  {
    $query="select fotografia from profesor where id_profesor='$chb[$i]'";
    if($result=mysql_query($query,$dblink))
    {
      $r=mysql_fetch_row($result);
      $foto=$r[0];
    }

    $query="delete from profesor where id_profesor='$chb[$i]'";
    if ($result=mysql_query($query,$dblink) )
    {
      $success=true;
      if ($foto!="")
        if (!$borrado=unlink("../files/prof/$foto"))
          echo "Registro borrado. Archivo de fotografÃa no pudo eliminarse.<br/>";
    }
  }
}
if ($success)

```

```

{
  header("Location: profesores.php");
  //echo "elementos eliminados<br/>";
  exit;
}
else
  echo "error el eliminar los elementos seleccionados";

}

?>

```

Código: accionprofesor.php

En el código de accionprofesor.php se estarán realizando las siguientes acciones de acuerdo al botón que se haya presionado:

Nuevo.- Nos redirecciona a la página de captura de profesores nuevos: nuevoprofesor.php.

Modificar.- Primero se revisa que haya uno o más profesores seleccionados, si es así, nos muestra el primer elemento a modificar después de hacer los cambios correspondientes al elemento mostrado, debe presionarse el boton “Guardar”, el cual nos llevará a otro código (“actualizaprofesor.php” que mostraremos mas adelante). En caso de que no se haya seleccionado ningun elemento de la lista, nos regresará a la misma pantalla.

Eliminar.- Primero se revisa que haya uno o más elementos seleccionados, si es así, elimina los elementos seleccionados y nos regresa a la misma pantalla con la lista de los elementos. Si no hay ningun elemento seleccionado, nos regresa a la misma pantalla.

A continuación mostramos el código para la página de captura de datos de un nuevo profesor.

```

<?
  session_start();
  if(!session_is_registered('administrador'))
  {
    header ("Location: ../login.html");
    exit;
  }
?>

<HTML>
<HEAD>
  <LINK href="../estiloadmin.css" rel="stylesheet" type="text/css">
</HEAD>

```



```

$myuser="administrador";
$mypsw="datasite";

if
(($nombre!="")&&($apellido_pat!="")&&($email!="")&&($user!="")&&($psw!="")&&($nivel!
=""))
{
if($dblink=mysql_connect($myhost,$myuser,$mypsw))
{

if(!mysql_select_db("sitedb",$dblink))
{
echo "Error al conectarse a la base de datos";
exit();
}

$query="select id_profesor from profesor where user='$user'";
if ($result=mysql_query($query,$dblink))
{
if (( $n=mysql_num_rows($result) ) > 0)
{
echo "El username ya se encuentra en uso. Intente con uno diferente.<br/>";
?> <A href="javascript:history.back()">Regresar</A> <?
exit;
}
} else { echo "Error al realizar una consulta."; exit; }

$errorfoto=0;
if ($fotografia_name!="")
{
if (strpos($fotografia_type,"jpeg")) $foto_ext="jpg";
if (strpos($fotografia_type,"gif")) $foto_ext="gif";
if (strpos($fotografia_type,"png")) $foto_ext="png";

if ((( $foto_ext != "jpg" ) && ( $foto_ext != "gif" ) && ( $foto_ext != "png" ) ) || ( $fotografia_size
> $MAX_FILE_SIZE)
{
echo "El archivo no es jpg o gif o png, o es mayor de $MAX_FILE_SIZE bytes";
$errorfoto="1";
}
else
{
if (copy($fotografia,"../files/prof/".$fotografia_name))
{
$foto=$fotografia_name;
}
else
{
echo "Error al subir la imagen.";
$errorfoto="1";
}
}
}
}
}
}

```

```

else
  $foto="";

  $passwd=crypt($psw,'P455WORD'); // P455WORD es la clave que usamos aqui para que
  la semilla no se genere automaticamente, nos servira para poder hacer las comparaciones al
  loguearse
  $psw=$passwd;

  $id_profesor="";
  $query="INSERT INTO
profesor(id_profesor,nombre,apellido_pat,apellido_mat,telefono,tel_cel,email,email_alt,website,fot
ografia,user,psw,nivel)
VALUES('$id_profesor','$nombre','$apellido_pat','$apellido_mat','$telefono','$tel_cel','$email','$e
mail_alt','$website','$foto','$user','$psw','$nivel)";
  if($result=mysql_query($query,$dblink))
  {
  $query="select max(id_profesor) as id_profesor from profesor";
  if($result=mysql_query($query,$dblink))
  {
    if ($foto!="")
  {
    $r=mysql_fetch_row($result);
    $query="update profesor set fotografia='p$r[0].$foto_ext' where id_profesor='$r[0]'";
    $result=mysql_query($query,$dblink);
    rename("../files/prof/$foto","../files/prof/p$r[0].$foto_ext");
  }
    if ($errorfoto=="1")
    {
      echo "Registro Guardado<br/>Hubo un error al subir la foto, reintente con una
modificacion."; ?> <a href="profesores.php">Ir a Profesores</a> <? exit;
    }
    $var="";
    header ("Location: nuevoprofesor.php");
    exit;

  }

  }
  else echo "Error al guardar el registro";
}
else
{
  print("Error al conectarse al servidor");
}
}
else { echo "Faltan datos obligatorios"; ?> <a href="javascript:history.back()">Regresar</a> <?
exit; }
?>

```

Código: guardaprofesor.php

Después de haberse guardado el nuevo profesor en la base de datos, nos regresará a la

misma pantalla para agregar un elemento más, y así sucesivamente hasta que se presione el botón “Terminar”.

Como mencionamos anteriormente, una de las acciones sobre cualquier elemento de la lista de profesores, es la de “Modificar”, cuando se selecciona ésta acción, se revisa que hayan sido seleccionados uno o más elementos de la lista, si es así, nos muestra la información del primer profesor para ser modificado, después para guardar los cambios hechos a este profesor, hay que presionar el botón “Guardar” y entrará en acción el código de `actualizaprofesor.php`:

```
<?
session_start();
if(!session_is_registered('administrador'))
{
    header ("Location: ../login.html");
    exit;
}

$myhost="localhost";
$myuser="administrador";
$mypsw="datasite";

if (($nombre!="")&&($apellido_pat!="")&&($email!="")&&($user!="")&&($nivel!=""))
{

    if($dblink=mysql_connect($myhost,$myuser,$mypsw))
    {

        if(!mysql_select_db("sitedb",$dblink))
        {
            echo "Error al conectarse a la base de datos";
            exit;
        }
        else
        {

            $errorfoto=0;
            if ($fotografia_name!="")
            {
                //echo "foto..";
                if (strpos($fotografia_type,"jpeg")) $foto_ext="jpg";
                if (strpos($fotografia_type,"gif")) $foto_ext="gif";
                if (strpos($fotografia_type,"png")) $foto_ext="png";
                //echo "nombre: $fotografia_name<br/>";
                //echo "tipo: $fotografia_type<br/>";
                //echo "tam: $fotografia_size<br/>";
                if (((($foto_ext != "jpg") && ($foto_ext != "gif") && ($foto_ext != "png")) || ($fotografia_size
                > $MAX_FILE_SIZE)
                {
```

```

echo "El archivo no es jpg o gif o png, o es mayor de $MAX_FILE_SIZE bytes";
$errorfoto=1;
}
else
{

if (copy($fotografia,"../files/prof/".$fotografia_name))
{
    $foto="$p$nprofesor.$foto_ext";
    rename("../files/prof/$fotografia_name","../files/prof/$foto");

}
else
{
    echo "Error al subir la imagen.";
    $errorfoto=1;
}

}
}
else
if ($fotoact!="") $foto=$fotoact;
else $foto="";

if ($psw!="")
{
    $passwd=crypt($psw,'P455WORD'); // P455WORD es la clave que usamos aqui para
que la semilla no se genere automaticamente, nos servira para poder hacer las comparaciones al
loguearse
    $psw=$passwd;
}
else
{
    $psw=$pswact;
}

$query="update
profesor
set
nombre='$nombre',apellido_pat='$apellido_pat',apellido_mat='$apellido_mat',telefono='$telefono',tel_cel='$tel_cel',email='$email',email_alt='$email_alt',website='$website',fotografia='$foto',user='$user',psw='$psw',nivel='$nivel' where id_profesor='$nprofesor'";

if($result=mysql_query($query,$dblink))
{
    $faltan=$faltan-1;
    $i=$i+1;
    if ($errorfoto==1)
    { echo "Registro Guardado<br/>Hubo un error al subir la foto, reintente con
una modificacion."; ?> <a href="profesores.php">Ir a Profesores</a> <? exit;
}

}
else echo "Error al guardar el registro";
}
}

```



```
    }  
    else echo "Error al consultar el registro";  
  
    }  
    else  
    {  
        header ("Location: profesores.php");  
    }  
  
?>
```

Código: actualizaprofesor.php

El código de `actualizaprofesor.php`, guardará la información actualizada en la BD, después revisará si aun faltan elementos a modificar, si es así, muestra la información del siguiente profesor para ser modificada su información, y así sucesivamente hasta que no haya mas elementos que modificar, en este caso, nos regresará a la lista de profesores.

El código de administración de profesores, mostrado previamente, debe servirnos como base para la administración del resto de las tablas de nuestra BD. Por lo tanto, en esta sección del presente trabajo, omitimos entrar en detalles en el código para las paginas de administración de las tablas restantes.

3.2. Pruebas y depuración.

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Es, por tanto, muy importante presentar en esta sección, los avances en las pantallas obtenidas con el código que hemos programado.

Inicialmente, y como lo mostramos en la sección anterior, al abrir el sitio nos abre la pantalla mostrada en la figura 3.1. Esta pantalla pertenece al area de consulta; pero primero entraremos al area de administración dando clic en la liga “Admin” que se encuentra en la parte inferior izquierda. Así, nos mostrará la pantalla mostrada en la figura 3.3 para poder

ingresar un usuario y contraseña. Después, si los datos son correctos, nos muestra la pantalla mostrada en la figura 3.2.

Ahora, decidimos administrar Profesores, dando clic en el menú de la izquierda, y nos muestra la pantalla de la Figura 3.4.

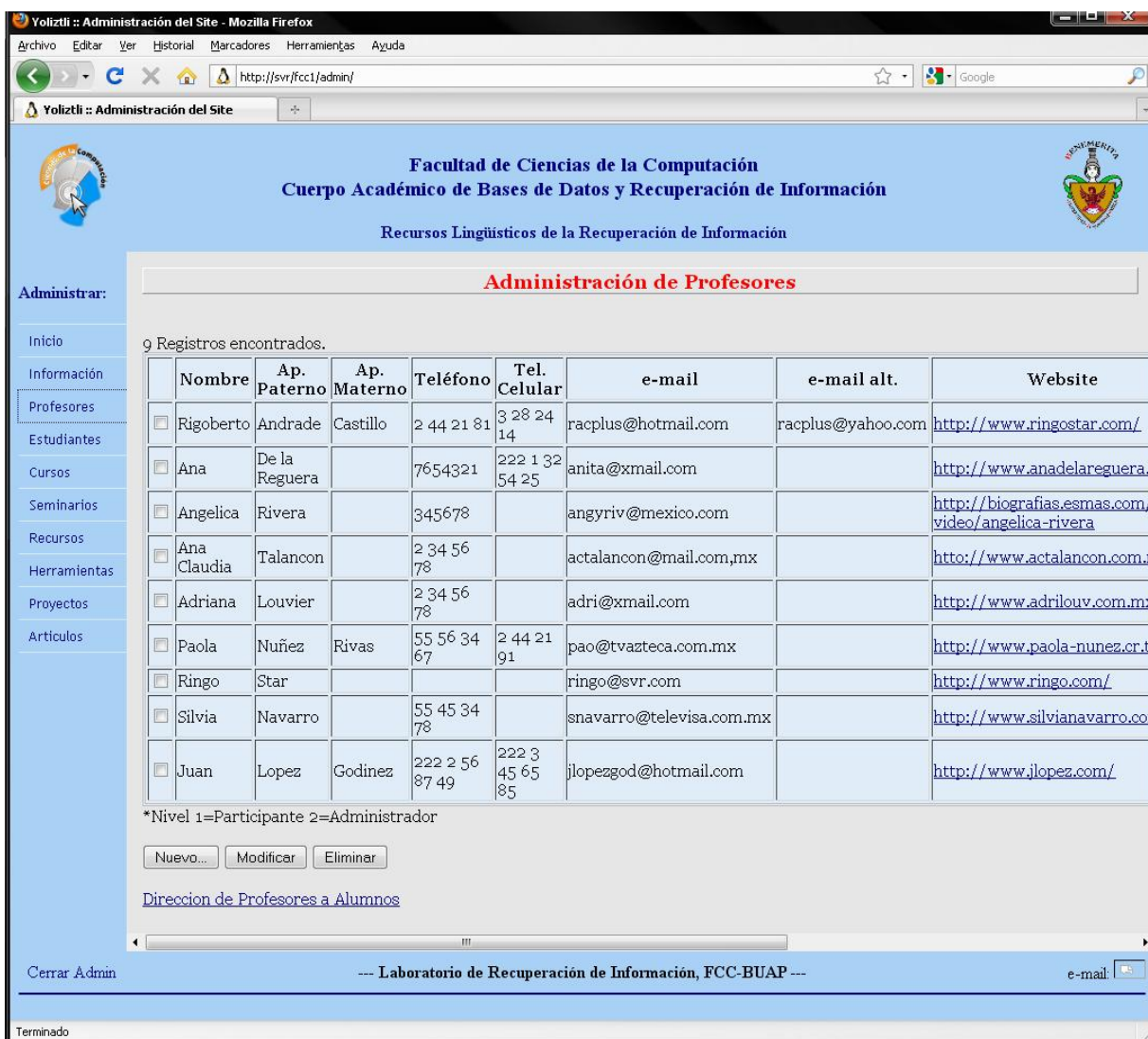


Figura 3.4.- Administración de profesores.

Dando clic en el botón “Nuevo...”, nos regresa la pantalla de la Figura 3.5.

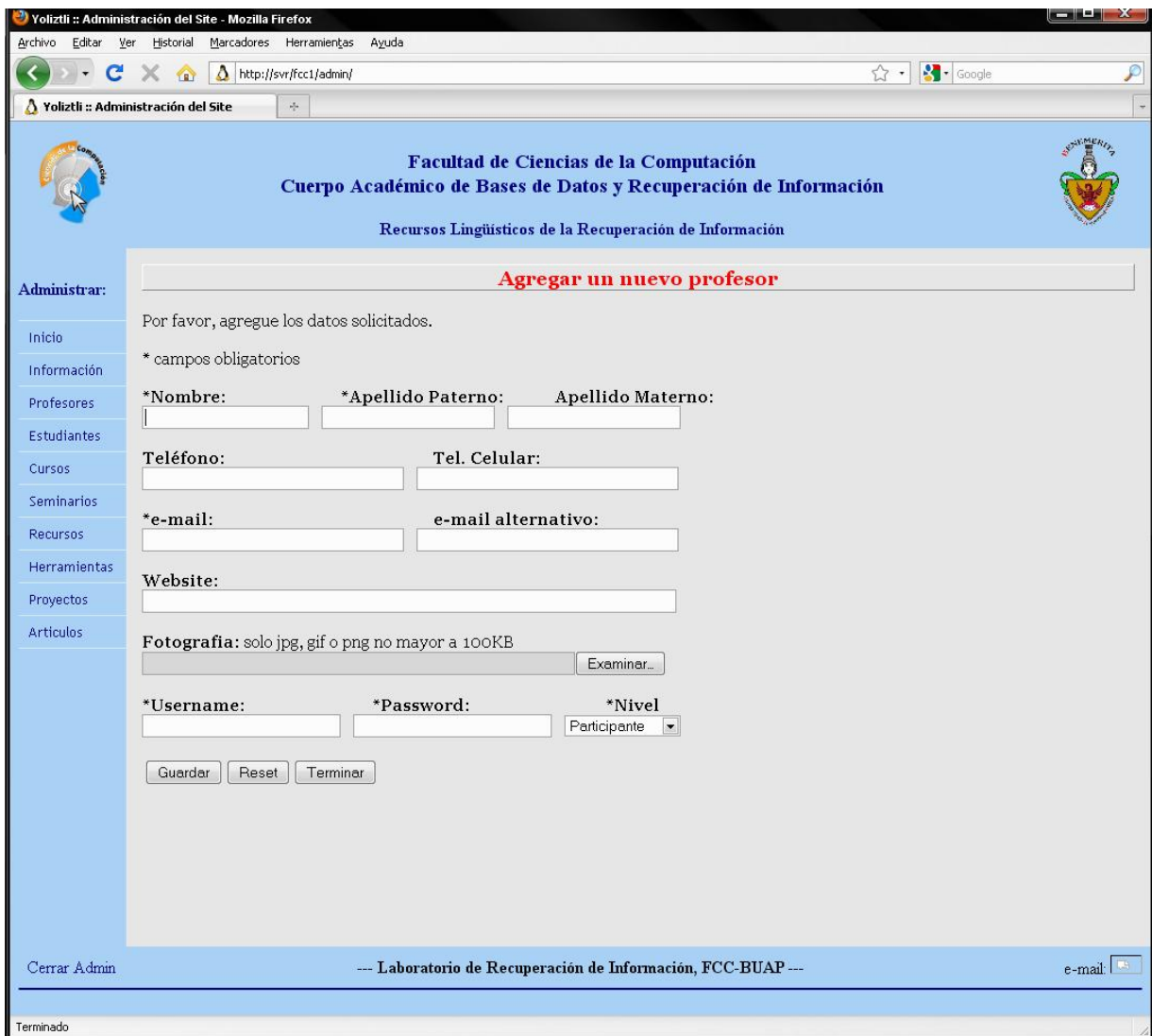


Figura 3.5.- Administración – Nuevo Profesor

Después de haber ingresado la información requerida, como se muestra en la Figura 3.6, Damos clic en el botón Guardar. Nos muestra nuevamente la pantalla de la Figura 3.5 para agregar más profesores. El botón Reset borrará la información que hayamos introducido hasta ese momento y así podemos nuevamente iniciar la captura. Con el botón Terminar abandonamos la pantalla de captura de nuevos profesores y nos regresa a la pantalla de la figura 3.4 mostrando ya en la lista los elementos que se agregaron.

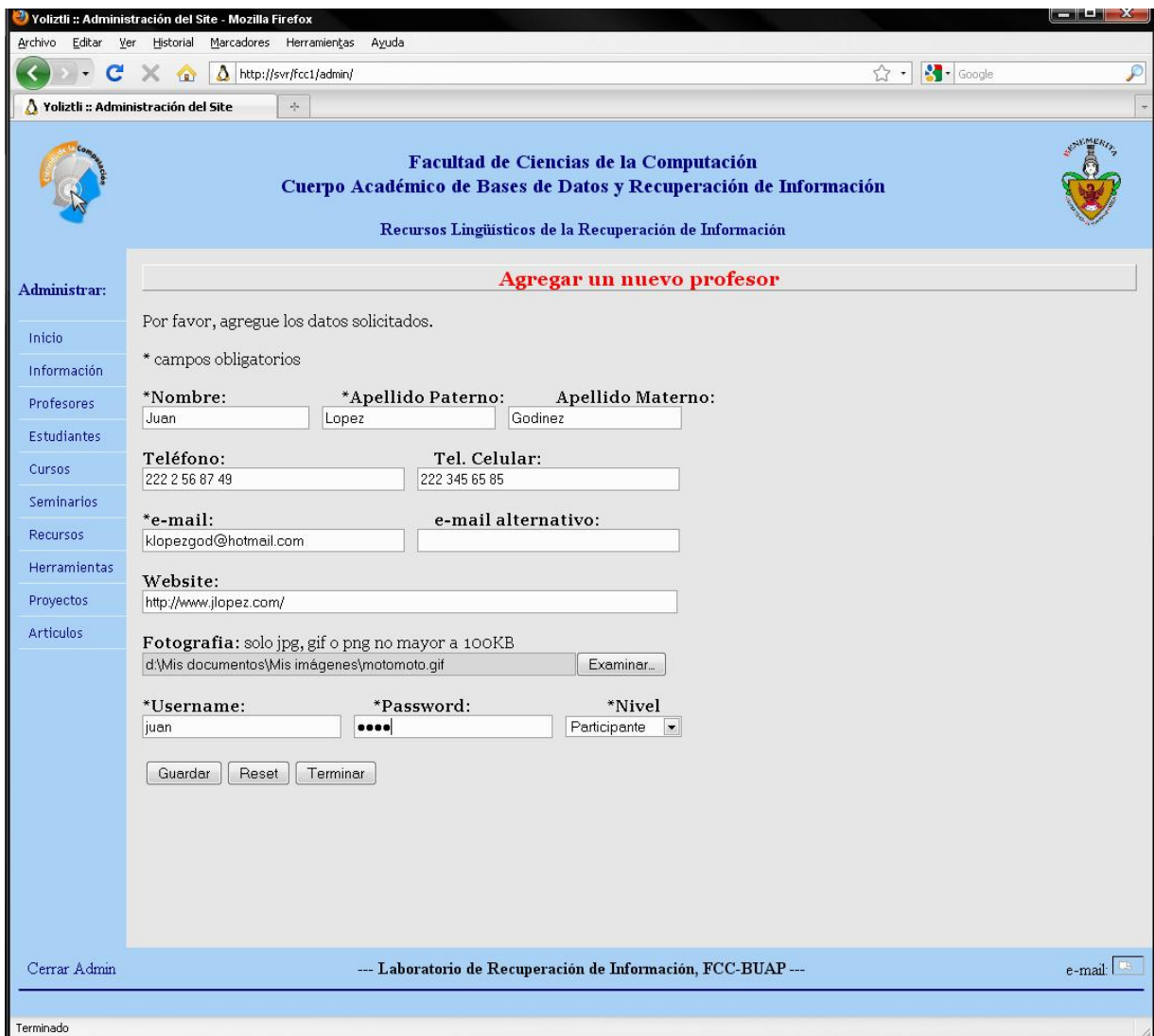


Figura 3.6.- Agregar nuevo profesor.

Si queremos modificar la información de algún(os) profesor(es), seleccionamos los elementos de la lista a modificar como se muestra en la Figura 3.7, y damos clic en el botón “Modificar”, y así nos muestra las pantallas de modificación como se muestra en la Figura 3.8. para cada uno de los profesores que se hayan seleccionado. Después de modificar y guardar todos los elementos seleccionados, nos regresa a la lista de profesores como se muestra en la Figura 3.4.

Yoliztli :: Administración del Site - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://svr/fcc1/admin/

Yoliztli :: Administración del Site

Facultad de Ciencias de la Computación
Cuerpo Académico de Bases de Datos y Recuperación de Información
 Recursos Lingüísticos de la Recuperación de Información

Administración de Profesores

Administrar:

Inicio

Información

Profesores

Estudiantes

Cursos

Seminarios

Recursos

Herramientas

Proyectos

Artículos

9 Registros encontrados.

	Nombre	Ap. Paterno	Ap. Materno	Teléfono	Tel. Celular	e-mail	e-mail alt.	Website
<input type="checkbox"/>	Rigoberto	Andrade	Castillo	2 44 21 81	3 28 24 14	racplus@hotmail.com	racplus@yahoo.com	http://www.ringostar.com/
<input checked="" type="checkbox"/>	Ana	De la Reguera		7654321	222 1 32 54 25	anita@xmail.com		http://www.anadelareguera.r
<input type="checkbox"/>	Angelica	Rivera		345678		angyriv@mexico.com		http://biografias.esmas.com/video/angelica-rivera
<input type="checkbox"/>	Ana Claudia	Talancon		2 34 56 78		actalancon@mail.com,mx		http://www.actalancon.com.r
<input type="checkbox"/>	Adriana	Louvier		2 34 56 78		adri@xmail.com		http://www.adrilouv.com.mx
<input checked="" type="checkbox"/>	Paola	Núñez	Rivas	55 56 34 67	2 44 21 91	pao@tvazteca.com.mx		http://www.paola-nunez.cr.t
<input type="checkbox"/>	Ringo	Star				ringo@svr.com		http://www.ringo.com/
<input type="checkbox"/>	Silvia	Navarro		55 45 34 78		snavarro@televisa.com.mx		http://www.silvianavarro.com
<input checked="" type="checkbox"/>	Juan	Lopez	Godinez	222 2 56 87 49	222 3 45 65 85	jlopezgod@hotmail.com		http://www.jlopez.com/

*Nivel 1=Participante 2=Administrador

[Direccion de Profesores a Alumnos](#)

Cerrar Admin --- Laboratorio de Recuperación de Información, FCC-BUAP --- e-mail:

Terminado

Figura 3.7.- Selección de elementos a modificar.

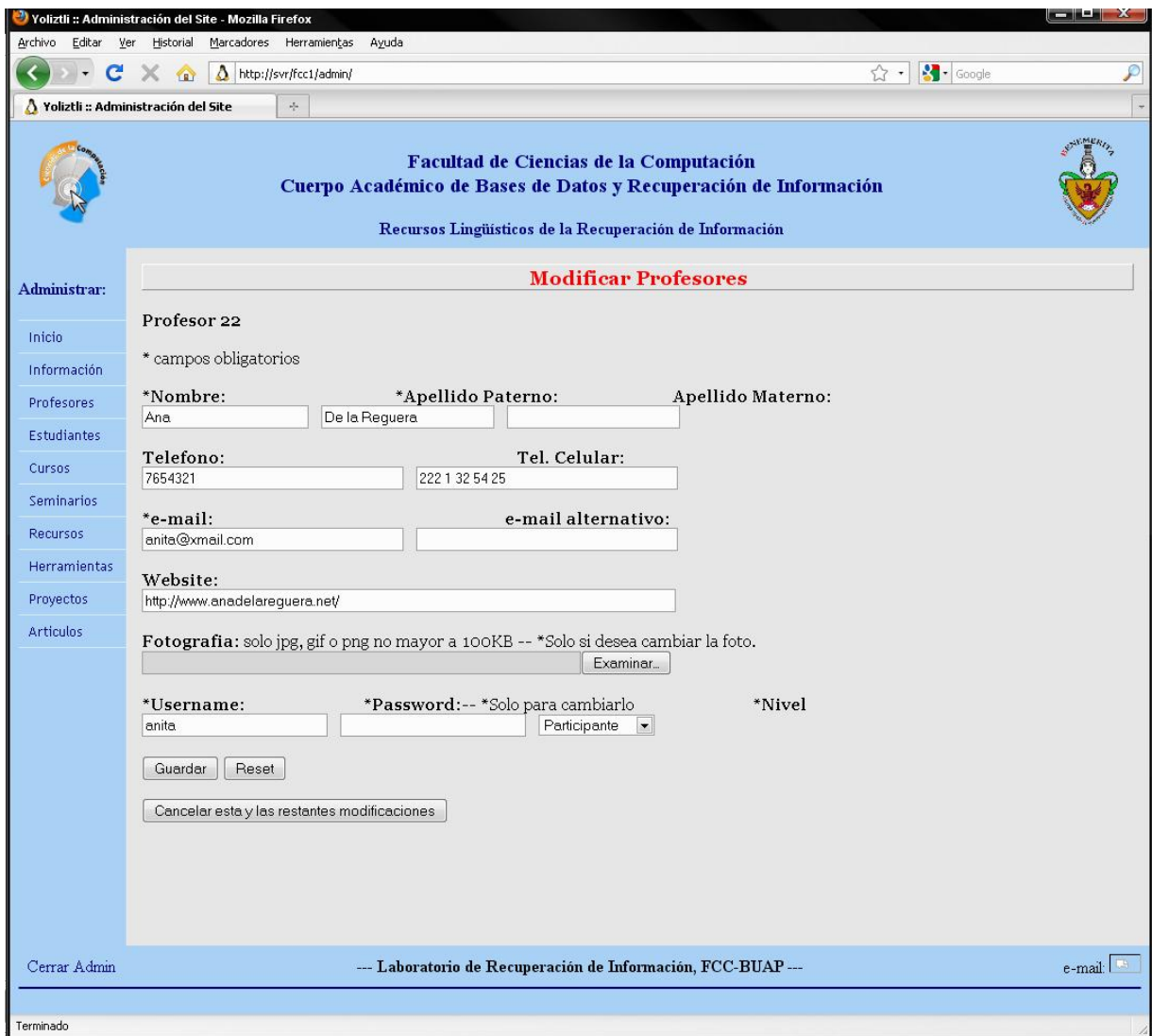


Figura 3.8.- Modificar Profesor.

Para eliminar elementos de la lista de profesores, solo se seleccionan los elementos a eliminar y damos clic en el botón “Eliminar”. Así, los elementos seleccionados son eliminados y nos regresa nuevamente a la pantalla de la Figura 3.4.

Cabe señalar que, de la misma forma que se trabaja administrando la tabla Profesores, se trabaja también con el resto de las tablas. No incluimos aquí los ejemplos de las otras tablas ya que sería algo repetitivo. Pero se hicieron también las pruebas pertinentes y han sido satisfactorias.

CAPÍTULO IV.

RESULTADOS Y CONCLUSIONES.

En la actualidad, la sociedad es totalmente dependiente de una infraestructura controlada por sistemas de software. Es imposible regresar a los sistemas manuales. Lo que implica que la disponibilidad, confiabilidad y seguridad en estos sistemas, tiene que ser muy alta.

4.1. Resultados

Después de todo lo realizado en el presente trabajo, hemos conseguido la construcción de un sitio de internet que será útil para el Cuerpo Académico de Base de Datos y Recuperación de Información de la Facultad de Ciencias de la Computación, integrado por Profesores y Alumnos principalmente de Maestría.

El sitio logrado es totalmente administrable vía web ya que todo el contenido es administrado desde interfaces de usuario creadas para dicho fin y no es necesario ningún lenguaje de programación para ello. Además, es dinámico ya que todo lo que se visualiza en el sitio es generado al momento de la consulta, gracias a las herramientas ocupadas en la realización de este trabajo tales como Php, MySQL, y Apache montados en un sistema Linux.

4.2. Conclusiones.

Gracias a los avances en la tecnología y, específicamente en los sistemas de información, ahora es posible acceder a un sinnúmero de información solo a través de una computadora con una conexión a internet. Gracias a ello, hemos implementado para nuestro

uso, un sistema a través del cual se podrán hacer consultas y algunos ejercicios sobre recuperación de información y bases de datos.

Aunque el sistema es totalmente funcional, podrían lograrse un poco más modificando un poco el aspecto para que sea más vistoso, con animaciones de flash por ejemplo, y aunque en este caso estamos ocupando hojas de estilo, podrían hacerse personalizaciones para cada usuario.

En lo personal, gracias a la investigación y las prácticas que he realizado para la finalización de este trabajo, he adquirido conocimientos en la programación dinámica con el lenguaje Php así como de algunas características en su configuración, también en bases de datos de MySQL, de Apache, y un poco más sobre Linux.

BIBLIOGRAFÍA

- 1** *Ingeniería de software explicada.*
Norris, Mark; Rigby Peter.
Megabyte-Noriega Editores. 1994.
- 2** *Ingeniería del Software, un enfoque práctico, 5ª Edición*
Pressman, Roger S.
McGraw-Hill.
- 3** *Fundamentos de Bases de Datos. 4ª Edición.*
Silberschatz, Abraham; Korth, Henry F; Sudarshan, S.
McGraw-Hill
- 4** *Computer Networks, 4th Edt.*
Tanenbaum; Andrew S.
Prentice Hall, 2003
- 5** *Domine PHP 5.*
López Quijado, José
Ra-Ma, 2008
- 6** *PHP, MySQL y Apache*
Julie C. Meloni
Anaya Multimedia
- 7** **PHP FAQ – Documentación PHP**
<http://www.php.net/FAQ.php>
- 8** *MySQL Reference Manual. MySQL, Technical Reference for version 5.0.0-alpha.*
MySQL AB, 1997-2004
- 9** *Guía Completa de Microsoft Windows 2000 Profesional*
Peter Norton
Prentice Hall, 2001
- 10** *Manual de PHP*
Stig Saether Bakken, Alexander Aulbach
Grupo de documentación de PHP, GNU