



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“DISEÑO Y DESARROLLO DE UN SISTEMA DE
INSCRIPCIONES A LABORATORIOS DE
INGENIERÍA A TRAVÉS DE INTERNET”**

T E S I S

**QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACIÓN**

P R E S E N T A N :

**PEDRO EDUARDO RODRÍGUEZ ALVAREZ
ALEJANDRO GUADARRAMA SERRANO**

ASESOR:

ING. RODOLFO VAZQUEZ MORALES

MEXICO, 2006





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimiento

A mi mamá María Alicia por todo su amor y ejemplo de lucha incansable ante las adversidades.

A mi papá Pedro por todo su amor y apoyo incondicional en todo momento.

A mi esposa Nohemi por todo su amor y paciencia para la realización de este proyecto.

A mis hijos Pedro Eduardo y Joel por ser mi constante fuente de inspiración de crecimiento.

A Mamá Lolita por todo su amor y ternura.

A mi compañero y amigo Alejandro Guadarrama Serrano por su amistad y colaboración para el desarrollo de este trabajo.

Al Ing, Rodolfo Vázquez Morales por la asesoría brindada para elaborar este proyecto.

A la FES Aragón y sus profesores por la formación académica que me ofrecieron.

Pedro Eduardo Rodríguez Álvarez

Dedicatoria

A mis padres María Alicia y Pedro, como muestra del resultado a su esfuerzo y apoyo incondicional para mi formación personal y académica, con todo mi amor, respeto y admiración.

A mis hijos Pedro Eduardo y Joel como muestra de superación constante en la vida, con todo mi amor.

A mi esposa Nohemi con todo mi amor por siempre.

A Mamá Lolita con todo mi amor y respeto.

Pedro Eduardo Rodríguez Álvarez

Dedicatoria

A Teresa Serrano Barragán mi madre

...porque, si tengo suerte, lograré tener la mitad de su fortaleza...

A Dios

...por la necesidad que tengo de comprenderlo...

A Gabriela, Manuel, Victoria, Juana, Isaura, Valente, Raymundo

...por todo su apoyo, consejo, paciencia y porque aun creen en mí.

A Pedro Eduardo Rodríguez Álvarez

...por enseñarme que la dedicación y disciplina te hacen realizar tus metas.

A mis profesores...

...por su decisión de compartir todo su conocimiento conmigo.

A la Universidad Nacional Autónoma de México

...por el tiempo que fue mi hogar.

Y a aquellas personas que, sin saberlo, han sido parte importante de mi vida.

Alejandro Guadarrama Serrano

Contenido

Introducción.....	1
Capítulo 1.Conceptos Básicos.....	7
1.1.Aplicaciones Web.	8
1.1.1.El Navegador Web o Browser.	10
1.1.2.El Servidor Web.....	11
1.2.Internet.....	13
1.2.1.Correo electrónico.	15
1.2.2.Servicio de noticias (Newsgroup).....	16
1.2.3.Transferencia de archivos.	16
1.2.4.Hipertexto (Web).....	16
1.2.5.Acceso remoto.	17
1.3.Programación Orientada a Objetos.....	17
1.3.1.Sobrecarga de Métodos.....	18
1.3.2.Clases, Objetos y Métodos.....	19
1.3.3.Herencia.	20
1.3.4.Polimorfismo.	21
1.4.Bases de Datos.	24
1.4.1.Base de Datos Relacional.....	26
1.4.2.Restricciones de asignación (Mapping).....	27
1.4.3.Claves (Llaves)	29
1.4.4.Normalización.....	29
1.5.Sistema Operativo Linux.....	35
1.5.1.¿Qué es LINUX?	36
1.5.2.Entorno de trabajo: el shell y X Window.	37
1.5.3.Usuarios y grupos.....	38
1.5.4.Estructura del sistema de archivos de Linux.	38
1.5.5.El shell: Comandos básicos de Linux.	44
1.6.Como funciona la integración de una base de datos en Internet... 62	
1.6.1.Servidor APACHE.....	63
1.6.2.Lenguaje PHP.	82

1.6.3.MySQL.....	85
Capítulo 2.Análisis y Diseño.	89
2.1.Problemática Actual.	90
2.2.Diseño de la Base de Datos.	98
2.2.1.Diccionario de Datos.....	98
2.2.2.Diagrama E/R (Modelo Conceptual).....	105
2.2.3.Diagrama E/R (Modelo Lógico).	106
2.2.4.Diagrama E/R (Modelo Físico).....	107
2.3.Identificación de Módulos.....	108
2.4.Descripción de Módulos dentro del sistema.	110
2.4.1.Sistema.	110
2.4.2.Utilerías.	114
2.4.3.Inscripciones y Calificaciones.	116
2.4.4.Administración.	118
2.4.5.Reportes.....	121
2.4.6.Opciones Generales.....	122
2.4.7.Reglas de negocio.....	124
Capítulo 3.Configuración del Sistema.....	129
3.1.Requerimientos Mínimos.	130
3.2.Estructura del CD de instalación.....	131
3.3.Arquitectura de la aplicación.....	131
3.4.Instalación en plataforma Windows.....	132
3.4.1.Instalación de MySQL.....	133
3.4.2.Instalación de Apache Web Server.....	142
3.4.3.Instalación de PHP + PEAR.	147
3.4.4.Instalación de la base de datos.....	150
3.4.5.Instalación del Sistema de Laboratorios de Ingeniería....	150
3.5.Instalación en plataforma Linux.	150
3.5.1.Instalación de MySQL.....	151

3.5.2.Instalación de APACHE + PHP + PEAR.....	151
3.5.3.Instalación de la base de datos.....	154
3.5.4.Instalación del Sistema de Laboratorios de Ingeniería....	155
3.6.Configuración del Sistema.....	155
3.7.Configuración de Seguridad.....	157
3.8.Seguridad en Windows.	157
3.9.Seguridad en Linux.....	157
Capítulo 4.Manual de Usuario y Administrador.....	161
4.1.Navegación básica Web.....	162
4.1.1.Control de Entrada de Texto.	164
4.1.2.Control de Lista de Selección.	164
4.1.3.Control de Casilla de Verificación.	165
4.1.4.Control de Botón de Radio.....	165
4.1.5.Control de Enlace.....	165
4.1.6.Control de Botón.....	166
4.1.7.Navegación Especial.....	166
4.2.Acceso al Sistema.	167
4.3.Menús del Sistema.....	168
4.3.1.Sistema.	168
4.3.2.Inicio.....	168
4.3.3.Alta en el Sistema.....	168
4.3.4.Acceso al Sistema.....	172
4.3.5.Salir.	174
4.4.Utilerías.....	174
4.4.1.Prácticas.	175
4.4.2.Apuntes.....	175
4.4.3.Consultar Sorteo.	176
4.5.Inscripciones y Calificaciones.....	177
4.5.1.Inscripciones.	178
4.5.2.Comprobante de Inscripción y Calificaciones.....	179

Contenido

4.5.3.Calificar.....	181
4.6.Reportes.....	182
4.6.1.Usuarios.....	183
4.6.2.Materias.....	184
4.6.3.Materias de Teoría Inscritas.....	185
4.6.4.Maestros.....	186
4.6.5.Grupos.....	186
4.6.6.Inscripciones y Calificaciones.....	187
4.6.7.Sorteo.....	188
4.7.Administración.....	189
4.7.1.Catálogo Ciclo Escolar.....	190
4.7.2.Catálogo Usuarios.....	192
4.7.3.Catálogo Maestros.....	195
4.7.4.Catálogo Materias.....	197
4.7.5.Catálogo Grupos.....	199
4.7.6.Prácticas.....	203
4.7.7.Apuntes.....	204
4.7.8. Carga de Sorteo de Inscripción.....	205
4.7.9.Carga de Materias de Teoría Inscritas.....	205
4.7.10.Validación de listado Materias de Teoría Inscritas.....	206
4.7.11.Cambiar Fecha/Hora de Inscripción de Alumnos.....	206
4.8.Pie de página.....	207
Recomendaciones.....	209
Conclusiones.....	213
Anexos.....	217
Glosario.....	237
Bibliografía.....	245

Introducción

Introducción

Hoy en día la mayor parte de las actividades que se llevan a cabo en cualquier empresa es con la ayuda de la computadora, más aún cuando se requiere el desarrollo de actividades en grupos de trabajo donde la sincronización de las tareas, la disponibilidad y la velocidad con que se maneja la información es crucial o simplemente si los datos requeridos se encuentran ubicados fuera de un área física, es decir, en otro departamento, piso, ciudad o país. Para ello y con el paso del tiempo, se fueron creando un conjunto de configuraciones lógicas y físicas de los centros de cómputo de las empresas que fueron ayudando de manera óptima a dicho intercambio de datos.

A dichas configuraciones se les dio el nombre de Redes de Computadoras y se clasificaron de varias formas: arquitectura, características lógicas y físicas, modelos matemáticos del comportamiento de la red y cobertura; esta clasificación es importante, porque es aquí donde se define la extensión geográfica que abarca una determinada red, que puede ir desde comunicar varios dispositivos en un área restringida (recinto, edificio, campus, etc.), varios edificios o hasta ciudades, estados o países. Como un ejemplo, tenemos a la creciente comunidad de la red Internet, que está formada a su vez por múltiples redes interconectadas entre sí a través de elementos que encaminan la información por diferentes puntos a nivel mundial.

Esta red, por la naturaleza misma de sus características, ha tenido una aceptación enorme en los últimos años ya que en ella se pueden realizar múltiples actividades desde un sólo punto geográfico como pueden ser actividades administrativas, comercio, educación, arte, etc. y es casi seguro que se encuentra todo tipo de información a través de esta red. Por eso mismo, la mayor parte de las empresas a nivel mundial, están experimentando un cambio revolucionario en sus sistemas para que prácticamente todas las tareas propias de dicha organización trabajen en Internet.

Introducción

Es en este punto, donde el presente trabajo de investigación toma su rumbo, hacemos una propuesta mediante el Desarrollo de un Sistema de Inscripciones a Laboratorios de Ingeniería para explotar los servicios de Internet aplicados a un caso real. Cabe señalar que dicho sistema podría ser implementado por la institución educativa a la que va dirigido el presente escrito.

La propuesta consiste en un sistema de información desarrollado con un conjunto de herramientas de código abierto como lo son: PHP lenguaje multiplataforma interpretado, un gestor de bases de datos SQL como MySQL, APACHE un servidor que nos permite en este caso utilizarlo como servidor Web para el sistema. Para que el sistema proporcione cierto grado de portabilidad, estará desarrollado de tal forma que pueda ser instalado en sistemas operativos Linux y Windows sin perder funcionalidad alguna.

Justificación.

La red de Internet es un medio para localizar, enviar y recibir información de diversos tipos. En el ámbito competitivo es esencial aprovechar las ventajas que esta vía electrónica proporciona, la presentación de información en diversos formatos, la rapidez de difusión de la misma, la reducción de costos, acceso desde cualquier sitio alrededor del mundo, con lo cual se incrementa el número de personas a las cuales llega la información, etc.

Internet provee de un formato de presentación dinámico para ofrecer campañas de publicidad para mejorar negocios; sistemas de educación por conferencia; mercadeo de infinidad de productos; turismo; etc. Millones de personas alrededor del mundo hacen uso de Internet; prácticamente, es el medio de comunicación principal utilizado para distintos fines. Pero no sólo es una vía para hacer negocios, sino una gran fuente de información, siendo éste uno de los principales propósitos con que fue creada. Una gran porción de dicha información requiere de un manejo especial que puede ser provista por las bases de datos.

Introducción

Antes, las bases de datos sólo podían utilizarse al interior de las instituciones o en redes locales, pero actualmente la red de Internet permite acceder a bases de datos desde cualquier parte del mundo. Éstas ofrecen, a través de la red, un manejo dinámico y una gran flexibilidad de los datos, ventajas que no podrían obtenerse a través de otro medio informativo. Con estas ventajas, los usuarios de Internet pueden obtener un medio que puede adecuarse a sus necesidades de información con un costo, inversión de tiempo y recursos mínimos. Asimismo. Las bases de datos son usadas para permitir el acceso y manejo de la variada información que se encuentra a lo largo de la red.

La evaluación de la seguridad es un tema de los más importantes en la interconexión de la red de Internet con las bases de datos. A nivel de una red local, se puede permitir o impedir el acceso a cierta información a diferentes usuarios, pero en la red mundial de Internet se necesita de controles más efectivos en este sentido ante espionaje, copia de datos, manipulación de éstos, etc.

La identificación del usuario es una de las formas de guardar la seguridad, permitiendo el acceso a distintos campos de una base de datos solamente a usuarios autorizados para ello, se puede conservar la seguridad e integridad de los datos. En este sentido, los datos pueden ser presentados a través de la red de una forma segura y proporcionar un mayor impacto en todos los usuarios de la red mundial.

Para la integración de bases de datos con la Web es necesario contar con una interfaz que realice las conexiones, extraiga la información de la base de datos, le dé un formato adecuado, de tal forma que pueda ser visualizada desde un navegador y permita lograr sesiones interactivas entre ambos dejando que el usuario haga elecciones de la información que requiere. Muchas instituciones se dan cuenta de la importancia que la red de Internet tiene en el desarrollo de sus potencialidades, ya que con ello pueden lograr una mejor comunicación con

Introducción

personas e instituciones situadas en cualquier lugar del mundo. Gracias a la conexión con la red de Internet, cada individuo o institución va teniendo acceso a mayor cantidad de información de las diversas ramas de la ciencia con distintos formatos de almacenamiento.

La mayor parte de la información es presentada de forma estática a través de documentos HTML, lo cual limita el acceso a los distintos tipos de almacenamiento en que ésta pueda encontrarse. Pero en la actualidad existe la posibilidad de utilizar aplicaciones que permitan acceder a información de forma dinámica, tal como las bases de datos, con contenidos y formatos muy diversos. Una de las ventajas de utilizar el Web para este fin, es que no hay restricciones para el sistema operativo que se debe de usar, permitiendo la conexión entre sí, de las páginas desplegadas en un navegador que funciona en una plataforma, con servidores de bases de datos alojados en plataformas distintas. Además no hay necesidad de cambiar el formato o estructura de la información dentro de la base de datos.

Capítulo 1

Conceptos Básicos

Para conocer el entorno de desarrollo, en este primer capítulo se describen los conceptos teóricos relacionados con el sistema propuesto, se da a conocer de una forma global la construcción y estructura de una aplicación Web. El alcance y el tipo de usuarios a los que va dirigida, así como la elección de las herramientas que fueron consideradas para el desarrollo de la misma.

1.1. Aplicaciones Web.

Inicialmente el Web era simple, los navegadores (*browsers*) presentaban documentos escritos en HTML que se obtenían de un servidor Web. Estos documentos HTML habitualmente presentan información de forma estática, sin más posibilidad de interacción con ellos.

El modo de crear documentos HTML ha variado a lo largo de la evolución de las tecnologías Web, pasando desde las primeras páginas escritas en HTML almacenadas en un servidor, hasta aquellas que se generan al momento como respuesta a una acción del cliente y cuyo contenido varía según las circunstancias.

Además, el modo de generar *sitios dinámicos* también ha cambiado, desde la utilización del CGI (*Common Gateway Interface*), un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos.

Los CGI siguen siendo utilizados aunque en menor proporción, puesto que la mayoría de los servidores Web los soportan debido a su sencillez. Además, nos proporcionan total libertad a la hora de escoger el lenguaje de programación para desarrollarlos.

Los *servlets*, son aplicaciones sin interfaz gráfica que se ejecuta en un servidor de Internet, son programas Java que proveen la funcionalidad de generar dinámicamente contenidos Web y también pueden ser ejecutados a través de una

línea de comando. A diferencia de los *applets* que son aplicaciones pequeñas que se ejecutan dentro de un navegador o browser, no poseen restricciones en cuanto a seguridad. Tienen las propiedades de cualquier aplicación Java y pueden acceder a los archivos del servidor para escribir y leer, cargar clases, cambiar propiedades del sistema, etc. Del mismo modo que las aplicaciones de programas Java, los servlets están restringidos por los permisos del sistema.

Otra de las tecnologías que más éxito ha obtenido y una de las que más se utiliza en Internet es el lenguaje de programación interpretado por el servidor PHP. Se trata de un lenguaje que permite incrustar HTML en los programas, con una sintáxis que proviene de C y Perl. Además de su facilidad de aprendizaje, su sencillez y potencia se está convirtiendo en una herramienta muy utilizada para muchos desarrollos.

Todas estas tecnologías se encuadran dentro de aquellas conocidas como *Server Side*, ya que se ejecutan en el servidor Web.

La arquitectura de un sitio tiene tres componentes principales: un servidor Web, una conexión de red y uno o más clientes (browsers). El servidor Web distribuye páginas de información con formato, a los clientes que la solicitan. Los requerimientos se realizan a través de una conexión de red y para ello se usa el protocolo http (HiperText Transfer Protocol).

Para abrir una página Web en un navegador, normalmente se teclea la correspondiente URL o se da un click en el vínculo adecuado.

Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó, como se muestra en la figura 1.

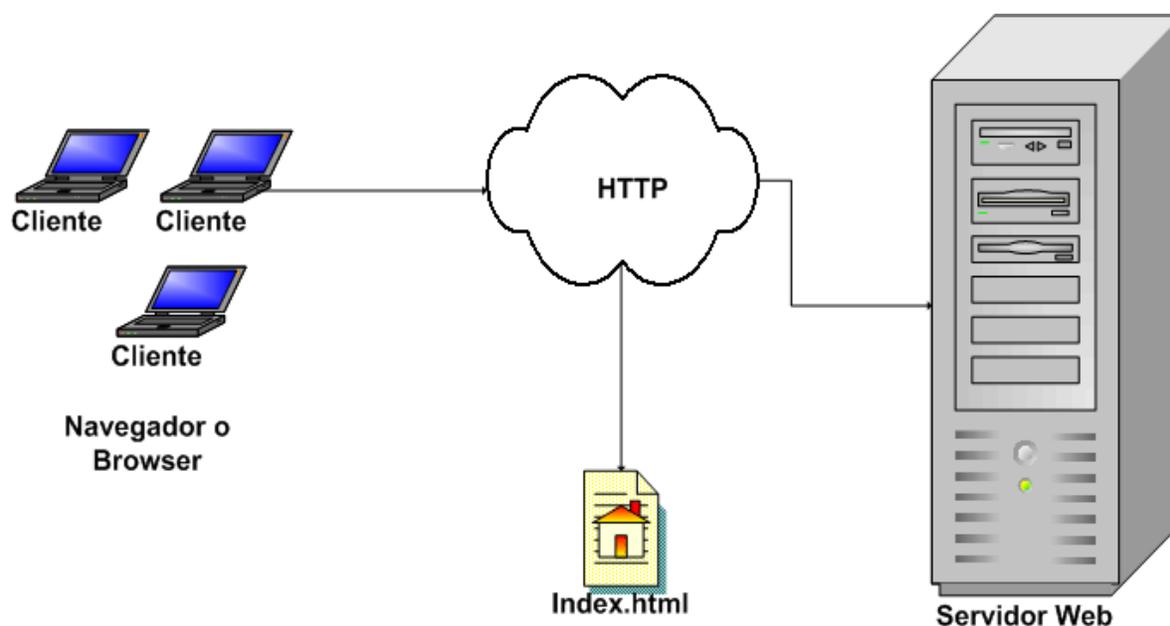


Figura 1. Arquitectura Web básica.

1.1.1. El Navegador Web o Browser.

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

Para todo esto, los fabricantes de navegadores, como Microsoft, Netscape Communications les han dotado de posibilidades de ejecución de programas de tipo script, con modelos de objetos que permiten manipular los contenidos de los documentos.

Estos lenguajes de programación son VBScript, JScript (ambas de Microsoft) y JavaScript (de Netscape), y proporcionan las soluciones que son ejecutadas en la máquina del cliente, *client side* y permiten realizar validaciones de datos recogidos en las páginas antes de enviarlos al servidor y proporcionan un alto grado de interacción con el usuario dentro del documento.

Otras de las posibilidades de los navegadores es la gestión del llamado HTML dinámico (Dynamic HTML, DHTML). Éste está compuesto de HTML, hojas de estilo en cascada, (Cascade Style Sheets, CSS), modelo de objetos y scripts de programación que permiten formatear y posicionar correctamente los distintos elementos HTML de las páginas Web, permitiendo un mayor control sobre la visualización de las páginas.

En esta línea, los navegadores han ido un poco más allá y permiten las visualizaciones de documentos XML (eXtensible Markup Language, Lenguaje de Etiquetas Extendido) después de haber sido transformado adecuadamente a HTML por las hojas de estilo extensibles (eXtensible Style Sheets, XSL). De esta manera se puede elegir visualizar ciertos elementos y otros no, dependiendo de las circunstancias.

Además, los navegadores permiten la ejecución de aplicaciones dentro de los documentos mostrados. Las dos posibilidades más populares son la tecnología ActiveX y los applets Java.

Los applets Java son pequeños programas que se descargan del servidor Web y se ejecutan en la JVM (*Java Virtual Machine*, Máquina Virtual de Java) del navegador.

1.1.2. El Servidor Web.

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. El servidor Web va a ser fundamental en el desarrollo de las aplicaciones del lado del servidor que vayamos a construir, ya que se ejecutarán en él.

La información mostrada en las páginas esta típicamente almacenada en archivos. Sin embargo, muchas veces esta información esta almacenada en una base de datos, y las páginas son creadas dinámicamente. Los sitios Web que usan este esquema, son llamados *sitios dinámicos*.

Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor.

En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, a pesar de que los clientes suelen ser máquinas menos potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: interfaz de presentación, lógica de la aplicación y los datos.

La capa intermedia (lógica de aplicación) es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación (interfaz de presentación) recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML. La arquitectura de las aplicaciones Web suele representar un esquema de tres niveles como se muestra en la figura 2.

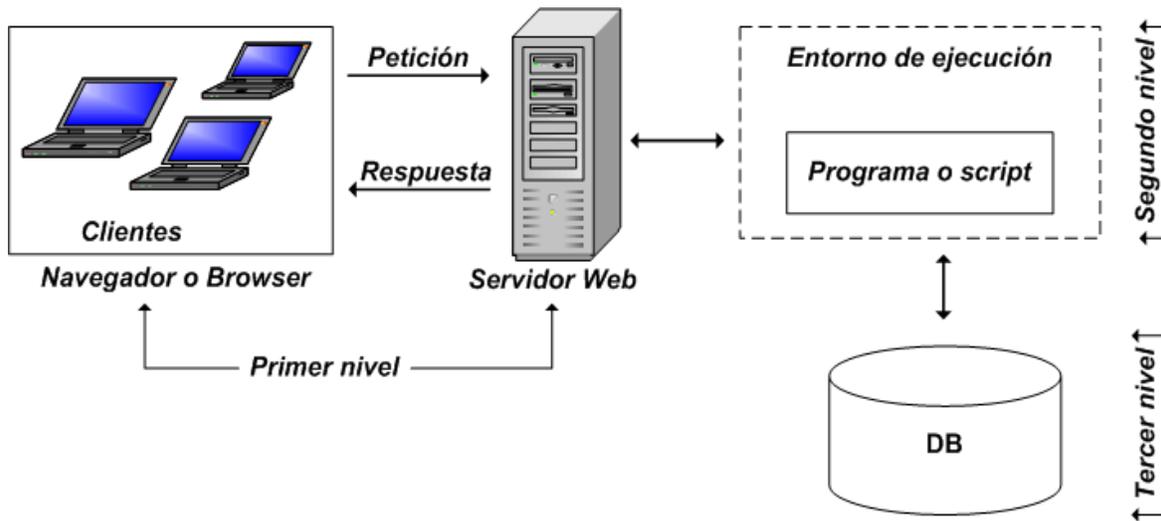


Figura 2. Arquitectura de una aplicación Web multinivel.

El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez)

1.2. Internet.

Internet se podría definir como una red global de redes de computadoras cuya finalidad es permitir el intercambio libre de información entre todos sus usuarios.

Para hacernos una idea, la persona que esta sentada delante de una computadora conectada a Internet tiene acceso a miles de redes de computadoras en todo el mundo y cuenta con millones de usuarios de la red potencialmente dispuestos a

compartir con ella conocimientos e información. A Internet están conectados medios de comunicación, oficinas de gobierno, instituciones académicas, investigadores, gente de negocios, estudiantes, etc.

Internet no es un sistema centralizado. No existe ninguna empresa con el nombre de Internet a la que podríamos llamar para que nos facilite un catálogo con todos sus servicios, o una lista con todas sus bases de datos, o un índice con todos los temas sobre los que podemos pedir información. Internet es simplemente una red de redes, donde cada red o incluso cada computadora, es una “entidad” gobernada de forma local.

Lo que cada entidad ponga a disposición del resto de los usuarios o lo que cada usuario haga al utilizar estos medios es responsabilidad de cada uno de ellos. Internet en sí, sólo se limita a establecer los procedimientos de interconexión de las distintas redes, así como las normas de interacción, pero cada red, cada computadora, o cada medio de transmisión tiene su propio dueño. Esta característica, que en principio podría parecer inconveniente, es realmente la ventaja de Internet y es lo que ha hecho que Internet crezca más allá de lo imaginable.

Internet basa su utilidad en cuatro servicios: correo electrónico, servicio de noticias, acceso remoto y transferencia de archivos. Con el correo electrónico, cada usuario puede ponerse en contacto con cualquier otro, a lo largo de todo el mundo e intercambiar mensajes, archivos, imágenes, etc. Con el servicio de noticias, cada usuario puede suscribirse a los temas que más le interesen y diariamente tendrá acceso a multitud de información sobre estos temas. Mediante el acceso remoto podemos conectarnos potencialmente como terminal de cualquier computadora o red situada en cualquier parte del mundo. El servicio de transferencia de archivos nos ofrece la facilidad de recuperar archivos desde cualquier computadora conectada a la red.

Con lo anterior, se ha podido suponer que cada usuario no solamente puede ser receptor de servicios de Internet, sino que cualquiera puede ser también generador de información.

Dado que no existe ninguna empresa llamada Internet que cobre ninguna tarifa a las distintas redes o usuarios que están conectados a ella, cada red paga los medios que la forman y su conexión al resto de las redes que componen Internet. Por lo tanto, cada red, de acuerdo con su coste particular de mantenimiento, fija sus propias tarifas a los usuarios que se conecten a ella.

Internet es sobre todo una herramienta para la comunicación. El uso que se le pueda dar a esta herramienta depende de las necesidades y de la inventiva de sus usuarios. Si hay algo que caracteriza a Internet es su capacidad de evolución y de adaptación a las nuevas necesidades que puedan aparecer.

1.2.1. Correo electrónico.

Éste es el uso más popular en Internet. El correo electrónico consiste en el intercambio de mensajes entre los usuarios de la red. Mediante el correo electrónico, cualquier usuario puede comunicar sus ideas tan claramente como lo haría con una carta y tan rápido como lo haría con una llamada telefónica. No hay sellos, no hay sobre, no hay la necesidad de contratar a un mensajero, no se tiene que imprimir, ni copiar en papel, etc.

El tiempo que se tarda solamente en escribir y enviar una carta por el procedimiento tradicional, es muy superior al que se necesita para escribir, enviar y obtener la respuesta mediante el correo electrónico. Con el correo electrónico no solamente se puede enviar texto, sino que también se puede enviar imágenes, audio y video. Los usuarios de Internet pueden utilizar el servicio de correo electrónico no solamente para intercambiar mensajes entre ellos, sino que también

pueden acceder a otras redes, aunque éstas no utilicen el mismo protocolo o no dispongan de las mismas normas de direccionamiento.

1.2.2. Servicio de noticias (Newsgroup).

El servicio de noticias es similar al del uso de listas de correo electrónico, donde la información es enviada a un grupo de direcciones de correo electrónico pero su diferencia estriba en que la información generada por los usuarios de Internet, que en este caso se les da el nombre de artículos, no son enviadas automáticamente a todas las direcciones contenidas en una lista, sino que se almacenan en un servidor de noticias al cual acceden los interesados cuando lo deseen. Los artículos de servicio de noticias están agrupados por temas; en este caso son llamados grupos de noticias.

1.2.3. Transferencia de archivos.

Mediante la transferencia de archivos, cualquier usuario puede traer a su computadora un archivo situado en cualquier otra computadora de cualquier parte del mundo. Mediante este sistema se pueden obtener software de libre distribución, informes sobre temas de interés, datos e información sobre temas particulares.

1.2.4. Hipertexto (Web).

Este servicio sin duda es uno de los más usados en todo el mundo, consiste en la presentación de información en páginas Web donde la información contenida puede tener en una sola palabra, enunciado, algún icono o símbolo de imagen, la aplicación de los llamados vínculos hacia otros documentos (páginas, llamados también sitios) en la red, esto nos facilita el ir “visitando” de sitio en sitio hasta encontrar la información deseada. A este sencillo procedimiento se le conoce como “navegar” sobre Internet.

1.2.5. Acceso remoto.

Mediante Internet se puede acceder en forma remota a cualquier ordenador o red situada en cualquier parte del mundo. Con los permisos de acceso adecuados se puede usar una supercomputadora o acceder a nuestra computadora o red de la oficina desde la casa o desde otra ciudad y trabajar en ellas como si se estuviese conectado directamente. Todos estos servicios convierten a Internet en un excelente medio de comunicación sin que para ello se tenga que coincidir en tiempo y espacio.

1.3. Programación Orientada a Objetos.

La *Programación Orientada a Objetos* (POO) permite realizar grandes programas mediante la unión de elementos más simples, que pueden ser diseñados y comprobados de manera independiente del programa que va a usarlos. Muchos de estos elementos podrán ser reutilizados en otros programas.

A estas "piezas", "módulos" o "componentes", que interactúan entre sí cuando se ejecuta un programa, se les denomina *objetos*. Estos objetos contienen tanto *datos* (variables y campos) como las *funciones* (métodos) que actúan sobre esos datos. Normalmente, cada uno de estos objetos corresponde a algún elemento que debe utilizar el programa. Algunos de estos elementos representan entidades del mundo real (matrices, personas, cuentas de banco, elementos mecánicos o eléctricos,...) y otros pueden ser componentes de la computadora (tanto de software como de hardware: otro programa, un archivo de disco, una impresora conectada en un puerto serie, una ventana abierta en la pantalla,...) También pueden ser estructuras de datos: colas, pilas, listas...

Durante la ejecución del programa, los objetos interactúan pasándose *mensajes* y *respuestas*. Es fundamental darse cuenta de que un objeto no necesita conocer el funcionamiento interno de los demás objetos para poder interactuar con ellos

(igual que el hombre no necesita conocer cómo funciona por dentro un televisor o una computadora para poder utilizarlos), sino que le es suficiente con saber la forma en que debe enviarle sus mensajes y cómo va a recibir la respuesta (al hombre le puede ser suficiente con saber cómo funcionan el interruptor, el dial del volumen y los botones de cambio de canal para utilizar un televisor).

Sucede a menudo que hay que utilizar varios ejemplares análogos de un determinado elemento u objeto (por ejemplo varias ventanas en la pantalla de la PC, varios usuarios, varios clientes, varias cuentas corrientes de un banco, etc.) A la definición genérica de estos objetos análogos se le llama *clase*. Así, una clase contiene una completa y detallada descripción de la información y las funciones que contendrá cada objeto de esa clase.

1.3.1. Sobrecarga de Métodos.

La sobrecarga (*overload*) de funciones (métodos) consiste en declarar y definir varias funciones distintas que tienen *un mismo nombre*. Dichas funciones se definen de forma diferente.

En el momento de la ejecución se llama a una u otra función dependiendo del número y/o tipo de los argumentos actuales de la llamada a la función. Por ejemplo, se pueden definir varias funciones para calcular el valor absoluto de una variable, todas con el mismo nombre *abs()*, pero cada una aceptando un tipo de argumento diferente y con un valor de retorno diferente.

La sobrecarga de funciones no admite funciones que difieran sólo en el tipo del valor de retorno, pero con el mismo número y tipo de argumentos. De hecho, el valor de retorno no influye en la determinación de la función que es llamada; sólo influyen el número y tipo de los argumentos. Tampoco se admite que la diferencia sea el que en una función un argumento se pasa por valor y en otra función ese argumento se pasa por referencia.

1.3.2. Clases, Objetos y Métodos.

Cualquier *función* (método) se puede comunicar con las demás a través de *variables globales*, del *valor de retorno* y de los *argumentos*, pasados *por valor* o *por referencia*.

Esta facilidad para comunicarse con otras funciones hace que se puedan producir efectos laterales no deseados. En un *Lenguaje Orientado a Objetos*, el centro del lenguaje no son las funciones sino los *datos*, es decir los objetos que contienen *datos* y *funciones* concretas que permiten manipularlos y trabajar sobre ellos. Esto hace que la mentalidad con la que se aborda la realización de un programa tenga que ser muy diferente.

Para proteger a las variables de modificaciones no deseadas se introduce el concepto de *encapsulado*, *ocultamiento* o *abstracción de datos*. Los miembros de una clase se pueden dividir en *públicos* y *privados*. Los miembros públicos son aquellos a los que se puede acceder libremente desde fuera de la clase. Los miembros privados, por el contrario, solamente pueden ser accedidos por los métodos de la propia clase.

Comunmente una clase ofrece un conjunto de *funciones públicas* a través de las cuales se puede actuar sobre los datos, que serán *privados*. Estas funciones o *métodos públicos* constituyen la *interfaz* de la clase. De esta forma, se garantiza que se hace buen uso de los objetos, manteniendo la coherencia de la información. Esto sería imposible si se accediera libre e independientemente a cada variable miembro. Al usuario le es suficiente con saber cómo comunicarse con un objeto, pero no tiene por qué conocer el funcionamiento interno del mismo.

Ya se ha hablado de las *funciones sobrecargadas*, que son funciones con el mismo nombre pero con distintos argumentos y definición. Otra posibilidad interesante es la de que objetos de distintas clases respondan de manera análoga

al aplicarles funciones con idéntico nombre y argumentos. Esta posibilidad da origen a las *funciones virtuales* y al *polimorfismo* que más adelante definiremos.

1.3.3. Herencia.

Necesidad de la herencia

La mente humana clasifica los *conceptos* de acuerdo a dos dimensiones: *pertenencia* y *variedad*. Se puede decir que el Ford Fiesta es un tipo de coche (*variedad* o, en inglés, una relación del tipo *is a*) y que una rueda es parte de un coche (*pertenencia* o una relación del tipo *has a*) Con la *herencia*, se consigue clasificar los tipos de datos (*abstracciones*) por *variedad*, acercando así un paso más la programación al modo de razonar humano.

Definición de herencia

La herencia, entendida como una característica de la programación orientada a objetos, permite *definir una clase modificando una o más clases ya existentes*. Estas modificaciones consisten habitualmente en añadir nuevos miembros (variables o funciones), a la clase que se está definiendo, aunque también se puede redefinir variables o funciones miembro ya existentes.

La clase de la que se parte en este proceso recibe el nombre de *clase base*, y la nueva clase que se obtiene se denomina *clase derivada*. Ésta a su vez puede ser *clase base* en un nuevo proceso de derivación, iniciando de esta manera una *jerarquía de clases*. Normalmente las clases base suelen ser más generales que las clases derivadas. Esto es así porque a las clases derivadas se les suelen ir añadiendo características, es decir variables y funciones, que diferencian y particularizan. En algunos casos una clase no tiene otra utilidad que la de ser *clase base* para otras clases que se deriven de ella. A este tipo de *clases base*, de las que no se declara ningún objeto, se les denomina *clases base abstractas* (*ABC, Abstract Base Class*) y su función es la de agrupar miembros comunes de

otras clases que se derivarán de ellas. Por ejemplo, se puede definir la clase *vehículo* para después derivar de ella *coche*, *bicicleta*, *patineta*, etc., pero todos los objetos que se declaren pertenecerán a alguna de estas últimas clases; no habrá vehículos que sean sólo *vehículos*. Las características comunes de estas clases (como una variable que indique si está parado o en marcha, otra que indique su velocidad, la función de arrancar y la de frenar, etc.), pertenecerán a la *clase base* y las que sean particulares de alguna de ellas pertenecerán sólo a la *clase derivada* (por ejemplo timbrar o pedalear, que sólo tiene sentido para una *bicicleta*, o la función *hunde_clutch* que sólo se aplicará a los vehículos de motor con varias velocidades). Este mecanismo de *herencia* presenta múltiples ventajas evidentes a primera vista, como la *posibilidad de reutilizar código* sin tener que escribirlo de nuevo. Esto es posible porque todas las *clases derivadas* pueden utilizar el código de la *clase base* sin tener que volver a definirlo en cada una de ellas.

1.3.4. Polimorfismo.

Polimorfismo, por definición, es la *capacidad de adoptar formas distintas*. En el ámbito de la Programación Orientada a Objetos se entiende por polimorfismo la capacidad de *llamar a funciones distintas con un mismo nombre*.

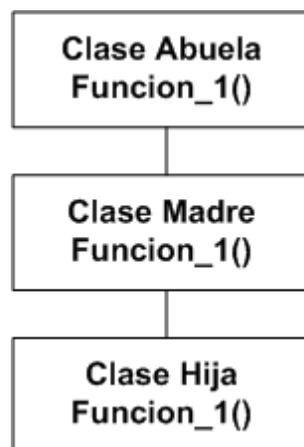


Figura 3. Jerarquía de clases

Estas funciones pueden actuar sobre objetos distintos dentro de una jerarquía de clases, sin tener que especificar el tipo exacto de los objetos como muestra la figura 3. En el ejemplo de la figura se observa una jerarquía de clases. En todos los niveles de esta jerarquía está contenida una función llamada *Función_1()*. Esta función no tiene por qué ser igual en todas las clases, aunque es habitual que sea una función que efectúe una operación muy parecida sobre distintos tipos de objetos.

Es importante comprender que el compilador no decide, en tiempo de compilación, cuál será la función que se debe utilizar en un momento dado del programa. Esa decisión se toma *en tiempo de ejecución*. A este proceso de decisión en tiempo de ejecución se le denomina *vinculación dinámica o tardía*, en oposición a la habitual *vinculación estática o temprana*, consistente en decidir en tiempo de compilación qué función se aplica en cada caso.

A este tipo de funciones, incluidas en varios niveles de una jerarquía de clases con *el mismo nombre pero con distinta definición*, se les denominan *funciones virtuales*. Hay que insistir en que la definición de la función en cada nivel es distinta.

El *polimorfismo* hace posible que un usuario pueda *añadir nuevas clases* a una jerarquía sin *modificar o recompilar* el código original. Esto quiere decir que si desea añadir una nueva clase derivada es suficiente con establecer la clase de la que deriva, definir sus nuevas variables y funciones miembro, y compilar esta parte del código, ensamblándolo después con lo que ya estaba compilado previamente. Es necesario comentar que las funciones virtuales son algo menos eficientes que las funciones normales. A continuación se explica, sin entrar en gran detalle, el funcionamiento de las funciones virtuales. Cada clase que utiliza funciones virtuales tiene un *vector de punteros*, uno por cada función virtual, llamado *v-table*.

Cada uno de los punteros contenidos en ese vector apunta a la función virtual apropiada para esa clase, que será, habitualmente, la función virtual definida en la propia clase.

En el caso de que en esa clase no esté definida la función virtual en cuestión, el puntero de *v-table* apuntará a la función virtual de su clase base más próxima en la jerarquía, que tenga una definición propia de la función virtual. Esto quiere decir que buscará primero en la propia clase, luego en la clase anterior en el orden jerárquico y se irá subiendo en ese orden hasta dar con una clase que tenga definida la función buscada.

Cada objeto creado de una clase que tenga una función virtual contiene un puntero oculto a la *v-table* de su clase. Mediante ese puntero accede a su *v-table* correspondiente y a través de esta tabla accede a la definición adecuada de la función virtual. Es este trabajo extra el que hace que las funciones virtuales sean menos eficientes que las funciones normales.

La idea central del polimorfismo es la de *poder llamar a funciones distintas aunque tengan el mismo nombre, según la clase a la que pertenece el objeto al que se aplican*. Esto es imposible utilizando nombres de objetos: siempre se aplica la función miembro de la clase correspondiente al nombre del objeto, y esto se decide en tiempo de compilación.

Sin embargo, utilizando punteros puede conseguirse el objetivo buscado. Recuérdese que un puntero a la clase base puede contener direcciones de objetos de cualquiera de las clases derivadas.

En principio, el tipo de puntero determina también la función que es llamada, pero si se utilizan funciones virtuales es el tipo de objeto el que apunta el puntero lo que determina la función que se llama. Esta es la esencia del *polimorfismo*.

1.4. Bases de Datos.

El término de bases de datos permite distintas interpretaciones. Ha sido usado para referirse tanto a un conjunto de tarjetas índice como a millones de datos, por ejemplo, que un gobierno recopila acerca de sus ciudadanos. Se puede decir entonces que una base de datos es un conjunto de registros integrados o bien una colección de datos interrelacionados.

Para comprender el significado de “bases de datos” debemos recordar que existe una jerarquía de datos que, en términos informáticos, como se muestra en la figura 4.

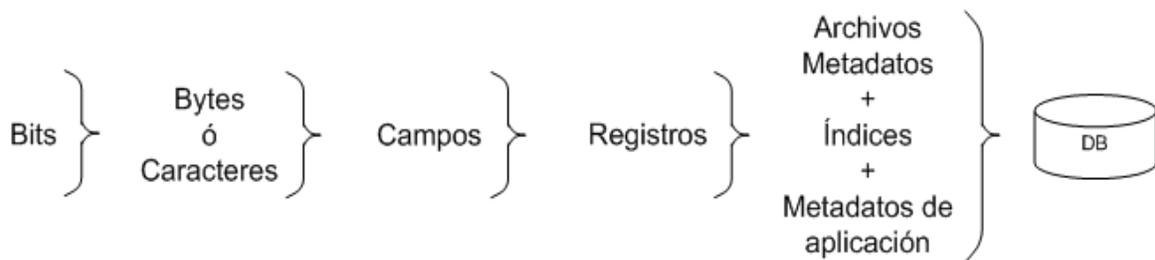


Figura 4. Jerarquía de datos

La implementación de bases de datos en las organizaciones trae consigo ventajas sobre los procesos manuales o sistemas de procesamientos de archivos que anteriormente se utilizaban pero que todavía no ofrecían por completo un óptimo manejo de la información, tales ventajas son:

- Reducción de la redundancia de datos
- Omisión de la inconsistencia de datos
- Posibilidad de compartir los datos
- Integridad de los datos

Un objetivo importante de un sistema de bases de datos es proporcionar a los usuarios una visión *abstracta* de los datos, es decir, el sistema esconde ciertos

detalles de cómo se almacenan y se mantienen los datos. Sin embargo, para que el sistema sea manejable los datos se deben extraer eficientemente. Este requerimiento ha llevado al diseño de estructuras de datos complejas para la representación de estos en la base de datos. Por lo que en muchas ocasiones esta complejidad se oculta a través de diversos niveles de abstracción para simplificar la interacción con el sistema.

Nivel físico: Es el nivel más bajo de abstracción y describe como se almacenan realmente los datos.

Nivel conceptual: Es el siguiente de abstracción describe qué datos son realmente almacenados en la base de datos. Aquí se describe la base de datos completa en términos de un número pequeño de estructuras relativamente sencillas. Este nivel es usado por los administradores de la base de datos quienes deben decidir qué información se va a guardar en la base de datos.

Nivel visual: Es el nivel más alto de abstracción describe sólo parte de la base de datos.

Para describir la estructura de una base de datos es necesario definir el concepto de *modelo de datos*: *una colección de herramientas conceptuales para describir datos, relaciones entre ellos, semántica asociada a los datos y restricciones de consistencia.*

Los diversos modelos de datos que se han propuesto se dividen en tres grupos: modelos lógicos basados en objetos, modelos lógicos basados en registros y modelos físicos de datos.

Este proyecto se basa en un modelo lógico basado en objetos: el modelo entidad-relación.

Los modelos lógicos basados en objetos se usan para describir datos en los niveles conceptual y de visión. Se caracterizan por el hecho de que proporcionan capacidad de estructuración flexible y permiten especificar restricciones de datos explícitamente.

1.4.1. Base de Datos Relacional.

El modelo de datos entidad-relación (ER) fue introducido por Peter Chen en 1976 se basa en una percepción de un mundo real que consiste en un conjunto de objetos básico llamados *entidades* y *relaciones* entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema que representa la estructura lógica global de la base de datos.

Una base de datos relacional consiste en una colección de *tablas (afinidades)*, a cada una de las cuales se les asigna un nombre único. Una fila en una tabla representa una relación entre un conjunto de valores. Puesto que una tabla es una colección de dichas relaciones, hay una estrecha correspondencia entre el concepto de *tabla* y el concepto matemático de *relación* del cual toma su nombre el modelo de datos relacional.

Cuando hablamos de una base de datos se debe diferenciar entre el *esquema de la base de datos* o el diseño lógico de la base de datos y una *instancia de la base de datos* que son los datos en la base de datos en un instante de tiempo dado. En general el esquema de una relación es una lista de atributos y sus correspondientes dominios. Una *entidad* es un objeto que existe y es distinguible de otros objetos. Un *conjunto de entidades* es un conjunto de entidades del mismo tipo. Los conjuntos de entidades no necesitan ser disjuntos. Una entidad está representada por un conjunto de *atributos* que son las particularidades o características distintivas de las entidades. Y se dice que un *dominio* es el conjunto de valores del mismo tipo. Una *relación* es una asociación entre varias entidades. Un *conjunto de relaciones* es un conjunto de relaciones del mismo tipo.

1.4.2. Restricciones de asignación (Mapping).

Una planificación de una empresa puede definir ciertas restricciones a las cuales deben ajustarse los contenidos de una base de datos. Una restricción importante es la de cardinalidad de asignación, que expresa el número de entidades con las que puede asociarse otra entidad mediante un conjunto de relaciones. Los tipos de cardinalidad (relación) que se pueden presentar son los siguientes:

- **Una a una:** Una entidad del conjunto A está relacionada únicamente con una entidad del conjunto B y viceversa, como se observa en la figura 5.

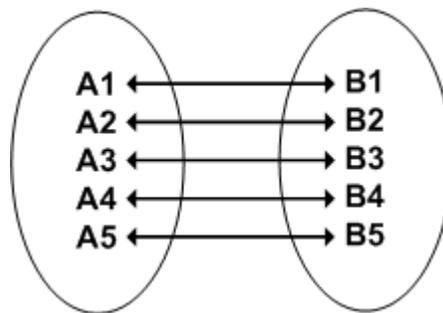


Figura 5. Relación uno a uno (mapeo:1:1)

- **Una a muchas:** Cada elemento del conjunto A está asociado con un número cualquiera de entidades en el conjunto B, sin embargo una entidad del conjunto B puede estar asociada a lo sumo con una entidad del conjunto A, como se muestra en la figura 6.

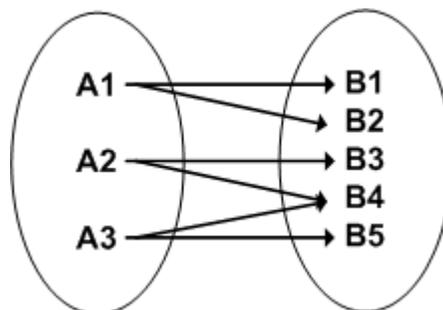


Figura 6. Relación uno a muchos (mapeo:1:N)

- **Muchas a una:** Cada elemento del conjunto A esta asociado a lo sumo con una entidad en el conjunto B, y una entidad del conjunto B puede estar asociada con un numero cualquiera de entidades en A, esto se muestra en la figura 7.

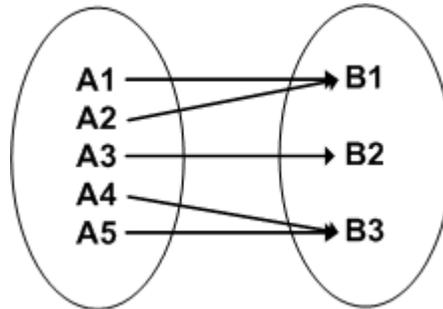


Figura 7. Relación muchos a una (mapeo:N:1)

- **Muchas a muchas:** Cada elemento del conjunto A esta asociado con un número cualquiera de entidades de en el conjunto B, y una entidad del conjunto B esta asociada con un número cualquiera de entidades en el conjunto A, en la figura 8 se muestra esta relación.

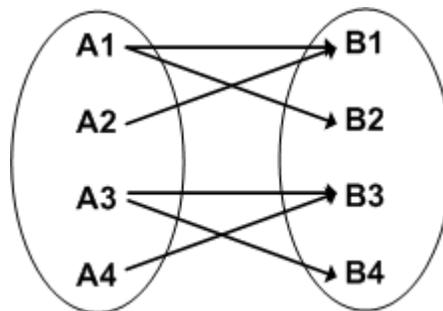


Figura 8. Relación muchos a muchos (mapeo:N:M)

El modelo relacional se ha establecido como el principal modelo de datos para aplicaciones comerciales de procesamiento de datos.

1.4.3. Claves (Llaves)

El concepto de *llave o clave* es un atributo o conjunto de atributos, cuyo valor unívoco identifica a la entidad o registro de un archivo, por ejemplo, el número de un empleado, la clave de un registro, la matrícula de un alumno, etc.

Existen dos tipos de claves, la primaria y la secundaria. La clave primaria es aquella que se utiliza unívocamente en un registro, representa el identificador de la entidad, formado por uno o más atributos.

Para acceder a un registro en particular, la computadora utiliza la clave primaria a través de un índice o algoritmo de direccionamiento.

Las claves secundarias son aquellas que no identifican a un registro único; por el contrario, identifican a todos los que cumplan con cierta propiedad, por ejemplo, los alumnos de la FES Aragón su clave primaria puede ser su número de cuenta y una llave secundaria la matrícula de la carrera a la que pertenecen.

1.4.4. Normalización.

La normalización es un proceso para convertir una afinidad que tiene ciertos problemas a dos o más afinidades que no tienen tales problemas. Y lo más importante la normalización puede usarse como lineamiento para verificar lo deseable y lo correcto de las afinidades.

Una afinidad es una tabla de dos dimensiones que tiene entradas de un solo valor. Todas las entradas en una columna dada son del mismo tipo; las columnas tienen un nombre único y el orden de las columnas no es importante. Las columnas también se denominan *atributos*. Dos hileras de una solo tabla nunca son idénticas y el orden de las hileras en la tabla no es importante. Las hileras también son llamadas *tuplas*. Los términos *tabla*, *archivo* y *afinidad* son sinónimos, los

vocablos *columna*, *campo* y *atributo* son sinónimos y los conceptos *hilera*, *registro* y *tupla* son sinónimos.

Para entender la normalización se deben comprender dos términos: dependencia funcional y clave. Estos conceptos aluden a las relaciones entre atributos en una afinidad. De manera que, como ya se describió una *afinidad* es una tabla y no una *relación* que es una asociación de entidades.

Una dependencia funcional es una relación entre uno o más atributos. Supóngase que si se da el valor de un atributo se puede obtener o buscar el valor de otro, por ejemplo si se conoce el "NCUENTA" de un "USUARIO" se puede hallar el valor de "NOMBRE" se dice que es *funcionalmente dependiente* de "NCUENTA".

Las dependencias funcionales se escriben usando la siguiente notación:

NCUENTA → USUARIO

La expresión se lee como *NCUENTA* determina funcionalmente *USUARIO*. Los atributos del lado izquierdo de la flecha se llaman *determinantes*. Una clave como ya habíamos definido es un grupo de uno o más atributos que identifican de modo único a un registro.

No todas las afinidades son deseables. Una tabla que cumple la definición mínima de una afinidad tal vez no tenga una estructura efectiva o apropiada. Para algunas afinidades, el cambiar los datos pueden tener consecuencias no deseables, llamadas *anomalías de modificación*. Las anomalías pueden eliminarse volviendo a definir la afinidad o más afinidades. En la mayor parte de las circunstancias, se prefieren las afinidades redefinidas o *normalizadas*. Vamos a considerar la siguiente afinidad, que, aunque no es una afinidad bien definida, nos sirve de ejemplo para nuestro caso:

INSCRIPCIÓN					
USUARIOID	NOMBRE	APATERNO	AMATERNO	MATERIA INSCRITA	CLAVE MATERIA
1D25	Pedro	Rodríguez	Álvarez	Dispositivos electrónicos	03065
1D48	Alejandro	Guadarrama	Serrano	Electricidad y Magnetismo	04230

Si se elimina la hilera para USUARIOID 1D25 no solo se perderá el nombre del estudiante sino también el hecho de que tiene inscrita una materia y que la materia tiene la clave 03065. Esto se denomina una *anomalía de eliminación*.

Puede usarse la misma afinidad para ilustrar una *anomalía de inserción*. Suponga que se quiere almacenar el nombre de otra materia Filtrado y Modulación y su clave correspondiente 04527, pero no se pueden acceder estos datos en la afinidad INSCRIPCIÓN hasta que un estudiante se inscriba a esa materia. Esta restricción se llama anomalía de inserción.

Se pueden corregir las anomalías de inserción y eliminación dividiendo la afinidad INSCRIPCIÓN en dos afinidades, cada una concerniente de un tema distinto. Se pueden agregar atributos de USUARIOID, NOMBRE, APATERNO, AMATERNO, MATERIA INSCRITA en una nueva afinidad llamada INSCRIPCIÓN y crear otra afinidad llamada MATERIAS que tenga los atributos NMATERIA, CVEMATERIA. Como se muestra:

INSCRIPCION				
USUARIOID	NOMBRE	APATERNO	AMATERNO	MATERIA INSCRITA
1D25	Pedro	Rodríguez	Álvarez	Dispositivos electrónicos
1D48	Alejandro	Guadarrama	Serrano	Electricidad y Magnetismo

MATERIAS	
NMATERIA	CVEMATERIA
Dispositivos electrónicos	03065
Electricidad y Magnetismo	04230

Si se elimina al estudiante 1D48 no se pierde el hecho de que Electricidad y Magnetismo tiene la clave 04230, de esta forma se pueden agregar las diferentes materias que hay disponibles incluso antes de que algún alumno se haya inscrito. Por lo tanto se han eliminado las anomalías de inserción y eliminación.

Esencia de la normalización.

Cada afinidad normalizada tiene un solo tema. Si tiene dos o más deberá fragmentarse en afinidades y cada una de estas afinidades tendrá un solo tema. Cada vez que se divida una afinidad es probable que surja la necesidad de crear una restricción de interrelación. Este proceso es la esencia de la normalización.

Cuando se encuentra una afinidad con anomalías de modificación se les elimina dividiendo la afinidad en dos o más, cada una de las cuales contendrá un solo tema.

Las afinidades pueden clasificarse por los tipos de anomalías de modificación a las cuales son vulnerables. En los años setenta, los teóricos relacionales investigaron estos tipos. Alguien encontraba una anomalía, la clasificaba y pensaba en una forma de prevenirla. Cada vez que esto ocurría, mejoraban los criterios para diseñar las afinidades.

Estas clases de afinidades y las técnicas para prevenir las anomalías son llamadas *formas normales*. Dependiendo su estructura, una afinidad puede estar en una primera forma normal, segunda forma normal, o alguna otra.

En 1970 E.F. Codd, estableció la primera, segunda y tercera formas normales (1NF, 2NF, 3NF) Más adelante se definió la forma normal Boyce-Codd (BCNF) y después se definieron la cuarta y quinta formas normales. Dichas formas normales están anidadas. Esto es, una afinidad en la segunda forma normal también está en la primera forma normal y una afinidad en 5NF (quinta forma normal), también esta en 4NF, BCNF, 3NF, 2NF y 1NF.

Primera forma normal.

Cualquier tabla de datos que cumpla con la definición de una afinidad se dice que esta en *primera forma normal*.

Segunda forma normal.

Una afinidad esta en segunda forma normal si todos sus atributos que no son claves dependen por completo de la clave. De acuerdo con esta definición, cada afinidad que tiene un atributo único como clave, esta en segunda forma normal.

Tercera forma normal.

Las afinidades en la segunda forma normal también tienen anomalías. Para ejemplificarlo tenemos la siguiente afinidad llamada Alumno:

Alumno		
IDUSUARIO	NCUENTA	NOMBRE
1D25	094355707	ALEJANDRO
1D32	094266537	PEDRO

Debido a que IDUSUARIO determina NCUENTA y NCUENTA determina NOMBRE indirectamente IDUSUARIO→NOMBRE. Un arreglo de este tipo se conoce como *dependencia transitiva*, ya que IDUSUARIO determina NOMBRE por medio del atributo NCUENTA. Para eliminar las anomalías de una afinidad en

segunda forma normal, debe quitarse la dependencia transitiva, lo que nos lleva a la definición de tercera forma normal:

Una afinidad esta en tercera forma si está en segunda forma normal y no tiene dependencias transitivas.

Forma Normal de Boyce – Codd.

Incluso las afinidades en tercera forma normal pueden tener anomalías. Para verificarlo considere la siguiente afinidad llamada InscrLab, suponga que los requerimientos para esta afinidad son que un estudiante puede tener una o más asignaturas, cada asignatura puede ser impartida por varios miembros de la Facultad y que solo un miembro de la Facultad imparte una asignatura.

InscrLab		
NCUENTA	MATERIA	PROFESOR
094355707	Electricidad y Magnetismo	Arista Patiño
094344722	Filtrado y Modulación	Ledesma
094344722	Dispositivos Electrónicos	Soto
094659874	Análisis de Circuitos	Luna
094521256	Control Digital	García

Ahora suponga que el estudiante 094344722 deja la escuela. Si se quita ese registro se perderá el hecho de que Ledesma imparte Filtrado y Modulación, esta es una anomalía de eliminación. En forma similar ¿cómo se almacena el hecho de que Ocampo imparte Microprocesadores? si esto no es posible hasta que un estudiante se inscribe a esta materia. Esta pues es una anomalía de inserción.

Situaciones como la descrita anteriormente conducen a la definición de la forma normal Boyce-Codd (BCNF): una afinidad esta en BCNF si cada determinante es una clave candidata.

Cuarta Forma Normal.

Existe pues el concepto de *dependencia de valores múltiples* se da cuando en una afinidad que tiene al menos tres atributos, dos de los cuales poseen valores múltiples y sus valores dependen sólo del tercer atributo, estas dependencias conducen a anomalías de modificación.

Se dice por tanto que *una afinidad está en cuarta forma normal si esta en BCNF y no tiene dependencias de valores múltiples.*

Quinta Forma Normal.

Finalmente, una afinidad está en quinta forma normal si se encuentra en cuarta forma normal y cada una de las dependencias de combinación está implicada por una clave candidata.

1.5. Sistema Operativo Linux.

La mayor parte de las computadoras que existen en la actualidad están diseñadas de forma que puedan ejecutar diversas tareas o programas. Estos programas pueden ir desde un procesador de textos, a un programa para la animación de gráficos tridimensionales o distintos tipos de juegos.

Para su correcto funcionamiento deben ser además capaces de acceder a los recursos de que dispone la computadora, como escribir o leer datos en un disco duro, mostrar un gráfico por pantalla, etc. Es evidente que si cada programa actuase de una forma independiente, existirían graves problemas y conflictos; por ejemplo, tendrían libertad para escribir sus datos sobre los de otro, etc. Para solucionar este tipo de problemas se desarrollaron los *Sistemas Operativos*, los cuales aportan unos mecanismos y reglas básicas de funcionamiento, de forma que los programas puedan acceder a los recursos de la computadora de una forma adecuada.

Aunque esta fue la funcionalidad inicial de los sistemas operativos, con el tiempo se han añadido otras muchas, como la ejecución de programas, el control de la memoria, la creación y control de interfaces gráficas de usuario, etc.

En la actualidad existe una gran cantidad de sistemas operativos dependiendo del tipo de plataforma en el que se va a ejecutar. Por ejemplo, para las PC uno de los sistemas operativos más difundidos es *Microsoft Windows*, tanto en las versiones 95, 98, 2000 y XP. Otros posibles sistemas operativos para este tipo de computadoras son *Solaris*, *OS/2*, *BeOS*, *Microsoft DOS*, o uno de los sistemas operativos más poderosos y en rápida expansión para PC: LINUX.

1.5.1. ¿Qué es LINUX?

Linux es un sistema operativo gratuito y de libre distribución inspirado en el sistema Unix, escrito por *Linus Torvalds* con la ayuda de miles de programadores en Internet. Unix es un sistema operativo desarrollado en 1970, una de cuyas mayores ventajas es que es fácilmente portable a diferentes tipos de plataformas, por lo que existen versiones de Unix para casi todos los tipos de ordenadores, desde *PC* y *Mac* hasta estaciones de trabajo y supercomputadoras. Al contrario que otros sistemas operativos, como *MacOS* (Sistema operativo de los *Apple Macintosh*), Unix no está pensado para ser fácil de emplear, sino para ser sumamente *flexible*. Por lo tanto, Linux no es en general tan sencillo de emplear como otros sistemas operativos, aunque se están realizando grandes esfuerzos para facilitar su uso. Pese a todo, la enorme flexibilidad de Linux y su gran estabilidad (y el bajo coste) han hecho de este sistema operativo una opción muy a tener en cuenta por aquellos usuarios que se dediquen a trabajar a través de redes, naveguen por Internet, o se dediquen a la programación.

Además el uso de Linux es brillante y cada vez más gente y más empresas están apoyando este proyecto, con lo que el sistema es cada vez más sencillo de emplear y los programas son cada vez mejores.

Una de los primeros conceptos que aparecen al iniciarse en Linux es el concepto de *distribución*. Una *distribución* es un agrupamiento del núcleo del sistema operativo Linux (la parte desarrollada por Linus Torvalds) y otra serie de aplicaciones de uso general o no tan general. En principio las empresas que desarrollan las distribuciones de Linux están en su derecho al cobrar una cierta cantidad por el software que ofrecen, aunque en la mayor parte de las ocasiones se pueden conseguir estas distribuciones desde Internet, de revistas o de amigos, siendo todas estas formas *legales y gratuitas*.

Las distribuciones más conocidas son *RedHat*, *Debian*, *Slackware*, *SuSE* y *Corel Linux*, todas ellas incluyen el software más reciente y empleado lo cual incluye compiladores de C/C++, editores de texto, juegos, programas para el acceso a Internet, así como el entorno gráfico de Linux: *X Window*.

1.5.2. Entorno de trabajo: el shell y X Window.

Al contrario que otros sistemas operativos, por defecto el trabajo con Linux no se realiza de una forma gráfica, sino introduciendo *comandos* de forma manual. Linux dispone de varios programas que se encargan de interpretar los comandos que introduce el usuario y realiza las acciones oportunas en respuesta. Estos programas denominados *shell* son el modo típico de comunicación en todos los sistemas Unix incluido Linux. Para muchas personas el hecho de tener que introducir los comandos de forma manual les puede parecer intimidante y difícil, aunque como se verá, los comandos de Linux son relativamente simples y muy poderosos.

No obstante, casi todas las distribuciones más recientes incluyen el sistema *X Window* (no *X Windows*), el cual es el encargado de controlar y manejar la interfaz de usuario. *X Window* es mucho más poderoso que otros entornos similares como *Microsoft Windows*, puesto que permite que el usuario tenga un control absoluto de la representación de los elementos gráficos.

1.5.3. Usuarios y grupos.

Linux es un sistema operativo *multitarea* y *multiusuario*. Esto quiere decir que es capaz de ejecutar varios programas (o tareas) de forma simultánea y albergar a varios usuarios de forma simultánea. Por lo tanto, todos los usuarios de Linux deben tener una cuenta de usuario en el sistema que establezca los privilegios del mismo.

A su vez Linux organiza a los usuarios en *grupos* de forma que se puedan establecer privilegios a un determinado grupo de trabajo, para el acceso a determinados archivos o servicios del sistema.

1.5.4. Estructura del sistema de archivos de Linux.

La base del sistema de archivos de Linux, es el archivo, que es la estructura empleada por el sistema operativo para almacenar información en un dispositivo físico como un disco duro, un disquete, un CD-ROM o un DVD. Como es natural, un archivo puede contener cualquier tipo de información, desde una imagen en formato PNG o JPEG a un texto o una página Web en formato HTML, ... El sistema de archivos es la estructura que permite que Linux maneje los archivos que contiene.

Todos los archivos de Linux tienen un nombre, el cual debe cumplir unas ciertas reglas:

- Un nombre de archivo puede tener entre 1 y 255 caracteres.
- Se puede utilizar cualquier carácter excepto la barra inclinada / y no es recomendable emplear los caracteres con significado especial en Linux, que son los siguientes: = \ ^ ~ ' " ` * ; - & [] () | & ~ < >. Para emplear archivos con estos caracteres o espacios hay que introducir el nombre del archivo entre comillas.

- Se pueden utilizar números exclusivamente si así se desea. Las letras mayúsculas y minúsculas se consideran diferentes, y por lo tanto no es lo mismo *carta.txt* que *Carta.txt* ó *carta.Txt*

Como en Windows, se puede emplear un cierto criterio de “tipo” para marcar las distintas clases de archivos empleando una serie de caracteres al final del nombre que indique el tipo de archivo del que se trata. Así, los archivos de texto, HTML, las imágenes PNG o JPEG tienen extensiones *.txt*, *.htm* (o *.html*), *.png* y *.jpg* (o *.jpeg*) respectivamente.

Pese a esto Linux sólo distingue tres tipos de archivos:

- **Archivos ordinarios**, son los mencionados anteriormente.
- **Directorios** (o carpetas), es un archivo especial que agrupa otros archivos de una forma estructurada.
- **Archivos especiales**, son la base sobre la que se asienta Linux, puesto que representan los dispositivos conectados a la computadora, como puede ser una impresora, escáner, etc. De esta forma, introducir información en ese archivo equivale a enviar información a la impresora. Para el usuario estos dispositivos tienen el mismo aspecto y uso que los archivos ordinarios.

Enlaces

Los enlaces son un tipo de archivo ordinario cuyo objetivo es crear un nuevo nombre para un archivo determinado. Una vez creado el enlace simbólico éste permite acceder al archivo que enlaza de igual modo que si se hubiera copiado el contenido del mismo a otro archivo, con la ventaja de que éste realmente no se ha copiado. Los enlaces simbólicos son especialmente útiles cuando se quiere que un grupo de personas trabajen sobre un mismo archivo, puesto que permiten compartirlo pero centralizan las modificaciones. Como ejemplo, se puede suponer la existencia de un archivo llamado *balance2005.txt*, al que se crea un enlace

simbólico *balance.txt*. Cualquier acceso a *balance.txt* es traducido por el sistema de forma que se accede al contenido de *balance2005.txt*.

En cualquier sistema operativo moderno, la estructura de archivos es jerárquica y depende de los directorios. En general la estructura del sistema de archivos se asemeja a una estructura de árbol, estando compuesto cada nudo por un directorio o carpeta, que contiene otros directorios o archivos. En Windows, cada unidad de disco se identifica como una carpeta básica que sirve de raíz a otras, y cuyo nombre es especial *a:*, *c:*, *d:* etc. En los sistemas Unix, y por lo tanto en Linux, existe una única raíz llamada */* de la que cuelgan todos los archivos y directorios, y que es independiente de qué dispositivos estén conectados a la computadora. El camino o path de un archivo o directorio es la secuencia de directorios que se ha de recorrer para acceder a un determinado archivo separados por */*. Supongamos la estructura de archivos de la figura 9.

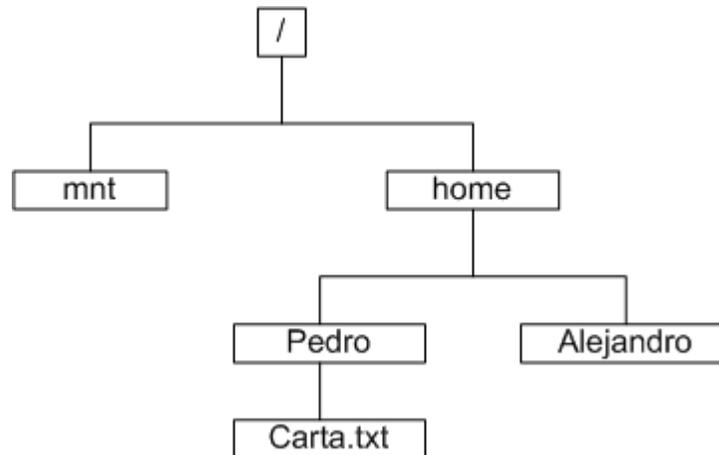


Figura 9. Ejemplo de una estructura de archivos en Linux.

Existen dos formas del path o camino:

- El camino absoluto que muestra toda la ruta a un archivo, ***/home/Pedro/Carta.txt***.

- El path relativo a un determinado directorio, por ejemplo si nos encontramos en el directorio **/home**, el path relativo al archivo **Carta.txt** es **Pedro/Carta.txt**

Para complicar aún más las cosas, todos los directorios contienen dos directorios especiales:

- El directorio actual, representado por el punto **.**
- El directorio padre representado por dos puntos **..**

Estando en el directorio **/home/Pedro** se puede acceder a **Carta.txt** con **/home/Pedro/Carta.txt** (path absoluto) o bien **../Pedro/Carta.txt** (path relativo).

En **/Pedro** como **./Carta.txt** o simplemente **Carta.txt**. El sistema de archivo de Linux sigue todas las convenciones de Unix, lo cual significa que tiene una estructura determinada, compatible y homogénea con el resto de los sistemas Unix. Al contrario que en Windows o MS-DOS el sistema de archivos en cualquier sistema Unix no está ligado de una forma directa con la estructura del hardware, esto es, no depende de si una determinada computadora tiene 1, 2 o más discos duros para crear las unidades **c:**, **d:** o **e:**. Todo el sistema de archivos de Unix tiene un origen único la raíz o root representada por **/**. Bajo este directorio se encuentran todos los archivos a los que puede acceder el sistema operativo. Estos archivos se organizan en distintos directorios cuya misión y nombre son estándar para todos los sistemas Unix.

/ Raíz del sistema de archivos.

/dev Contiene archivos del sistema representando los dispositivos que estén físicamente instalados en la computadora.

/etc Este directorio está reservado para los archivos de configuración del sistema. En este directorio no debe aparecer ningún archivo binario (programas).

Bajo éste deben aparecer otros dos subdirectorios:

/etc/X11 Archivos de configuración de **X Window**

/etc/skel Archivos de configuración básica que son copiados al directorio del usuario cuando se crea uno nuevo.

/lib Contiene las librerías necesarias para que se ejecuten los programas que residen en **/bin** (no las librerías de los programas de los usuarios)

/proc Contiene archivos especiales que o bien reciben o envían información al **kernel** del sistema (Se recomienda no modificar el contenido de este directorio y sus archivos)

/Sin Contiene programas que son únicamente accesibles al súper usuario o **root**.

/usr Este es uno de los directorios más importantes del sistema puesto que contiene los programas de uso común para todos los usuarios.

Su estructura suele ser similar a la siguiente:

/usr/X11R6 Contiene los programas para ejecutar **X Window**.

/usr/bin Programas de uso general, lo que incluye el compilador de C/C++.

/usr/doc Documentación general del sistema.

/usr/etc Archivos de configuración generales.

/usr/include Archivos de cabecera de C/C++ (.h)

/usr/info Archivos de información de GNU.

/usr/lib Librerías generales de los programas.

/usr/man Manuales accesibles con el comando *man*

/usr/sbin Programas de administración del sistema.

/usr/src Código fuente de programas.

Existen además de los anteriores otros directorios que se suelen localizar en el directorio **/usr**, como las carpetas de los programas que se instalen en el sistema.

/var Este directorio contiene información temporal de los programas (lo cual no implica que se pueda borrar su contenido, de hecho, no se debe hacer)

Acceso a los diferentes sistemas de archivos

Como se ha visto anteriormente, el sistema de archivos de Linux sólo tiene una raíz y su estructura es independiente de los dispositivos de almacenamiento existentes. Esto implica que el procedimiento a emplear para acceder a la información almacenada en los distintos sistemas de almacenamiento de una computadora no es tan sencillo como en Windows, y requiere un proceso llamado “montado”. Cuando se ha terminado de trabajar con un determinado dispositivo hay que “desmontarlo”.

Por ejemplo el proceso para leer un disquete sería el siguiente:

1. Introducir el disquete en la unidad de discos flexibles.
2. Montar el sistema de archivos del mismo.
3. Leer, grabar, y manipular el contenido del disquete.
4. Desmontar el sistema de archivos del disquete.
5. Extraer el disquete de la unidad.

El proceso puede parecer complejo pero es el precio a pagar por la seguridad, puesto que de esta forma se garantiza que no exista ninguna aplicación que esté usando el disquete cuando se extraiga. (En el caso de los CD-ROM Linux impide su extracción hasta que se desmonta). Para complicar más las cosas sólo el administrador o **root** tiene permiso para montar y desmontar un sistema de archivos (por motivos de seguridad), aunque esto puede ser arreglado.

Permisos

Linux, al igual que todos los sistemas Unix, mantiene un sistema de permisos de acceso a los archivos muy estricto, a fin de controlar qué es lo que se puede hacer con ellos, y quien lo puede hacer.

Estos permisos se identifican con letras y son:

r	permiso de lectura del archivo
w	permiso de escritura en el archivo
x	permiso de ejecución del archivo
s	permiso para cambiar el propietario del archivo

Al contrario que en Windows o MS-DOS los programas ejecutables de Linux no están marcados por una determinada extensión (**.exe**) sino por un atributo, el permiso de ejecución **x**. Si se elimina este atributo a un programa, Linux no será capaz de ejecutarlo.

A su vez, cada uno de estos permisos se aplica: al dueño del archivo (**u**), al grupo de usuarios al que pertenece el dueño (**g**), o al resto de usuarios (**a**). Así un archivo determinado puede tener permiso para ser leído, escrito y ejecutado por su dueño, leído y ejecutado por el grupo al que pertenece y no tener ningún tipo de acceso para los demás usuarios. Como se puede entender este tipo de mecanismo es especialmente útil cuando se trabaja en grupo en un determinado proyecto.

1.5.5. El shell: Comandos básicos de Linux.

Sabemos que Linux cuenta con ambientes gráficos (por ejemplo el **GNOME** y **KDE**) como cualquier sistema operativo moderno, no obstante cualquier usuario de Linux acabará antes o después relacionándose con el sistema empleando el modo texto. Este modo se basa en la ejecución de una serie de comandos, que son interpretados por un programa o *shell*.

Linux dispone de varios de estos programas pero el más habitual es conocido como *bash* o *Bourne Shell*. Si Linux se ha arrancado en modo texto, el sistema arranca de forma directa el *shell* y queda a la espera de introducción de nuevos comandos. Si se ha arrancado en modo gráfico, se puede acceder al *shell* de dos formas:

- Se puede acceder al *shell* del sistema presionando alguna de las siguientes combinaciones de teclas:

<ctrl>+<alt>+<F1>

<ctrl>+<alt>+<F2>

<ctrl>+<alt>+<F3>

<ctrl>+<alt>+<F4>

<ctrl>+<alt>+<F5>

<ctrl>+<alt>+<F6>

Esto hace que el sistema salga del modo gráfico y acceda a alguna de las seis consolas virtuales de Linux, a las cuales también se puede acceder cuando se arranca en modo de texto. Para volver al modo gráfico hay que presionar:

<ctrl>+<alt>+<F7> o <ctrl>+<alt>+<F8>.

- La segunda forma es más cómoda y menos radical permitiendo acceder al shell desde el mismo entorno gráfico. Para esto hay que abrir un programa llamado terminal o consola, por ejemplo: *kconsole* (en el entorno *KDE*), *xterm*, *gnome-terminal* (en *GNOME*)

Existen una serie de nociones básicas que hay que tener en cuenta a la hora de introducir los comandos. En primer lugar citaremos las siguientes:

- Los comandos hay que teclearlos exactamente.
- Las letras mayúsculas y minúsculas se consideran como diferentes.
- En su forma más habitual (los **shells** de Bourne o de Korn), el sistema operativo utiliza un signo de **\$** como **prompt** para indicar que está preparado para aceptar comandos, aunque este carácter puede ser fácilmente sustituido por otro u otros elegidos por el usuario. En el caso de que el usuario acceda como administrador este signo se sustituye por **#**.

- Cuando sea necesario introducir el nombre de un archivo o directorio como argumento a un comando, Linux permite escribir las primeras letras del mismo y realiza un autorellenado al presionar la tecla del **tabulador**. Si no puede distinguir entre diversos casos rellenará hasta el punto en el que se diferencien. Por ejemplo, supongamos una carpeta con los siguientes directorios:

```
Programas
Documentos _ proyecto
Documentos _ privados
```

Al escribir `cd Pr<tab>` Linux rellenará el resto del contenido hasta escribir `cd Programas`. Por el contrario al escribir `cd D<tab>` escribirá `cd Documentos_`

Algunos comandos sencillos de Linux

Para efectuar el cambio o la introducción de un *password* o *contraseña* se utiliza el comando **passwd**. El proceso a seguir es el siguiente:

passwd

(*current*) *UNIX password*: (Se teclea la contraseña actual; no aparece en pantalla)

New UNIX password: (se teclea la nueva contraseña; no aparece en pantalla)

Retype new UNIX password: (se teclea de nuevo la nueva contraseña comprobando que se ha tecleado bien, si coincide se produce el cambio)

A continuación se describen algunos comandos sencillos de que pueden ser útiles para familiarizarse con los comandos del sistema.

date Muestra por pantalla el día y la hora.

cal 2006 Muestra el calendario del año 2006.

cal 02 2006	Muestra el calendario de febrero de 2006.
who	Indica qué usuarios tiene la computadora en ese momento, en qué terminal están y desde qué hora.
whoami	Indica cuál es la terminal y la sesión en la que se está trabajando.
man comando	Todos los manuales de Linux están dentro del propio sistema operativo, y este comando permite acceder a la información correspondiente al comando solicitado. Por ejemplo con man who aparecerá por pantalla y de forma formateada por páginas, la explicación del comando who . Se puede navegar a través de estas páginas con los cursores del teclado, y presionando q para salir.
clear	Este comando limpia la consola

Directorio personal

El directorio personal es un directorio con un determinado nombre asignado a un usuario. Los directorios personales habitualmente son subdirectorios de **/home** (en algunos casos se utiliza **mnt**, u otro subdirectorio de orden inferior) Generalmente el nombre coincide con el del *nombre de usuario*, aunque puede no ser así, y varios usuarios pueden estar trabajando en el mismo directorio. Cada usuario de Linux puede crear una estructura en árbol de subdirectorios y archivos tan compleja como desee bajo su directorio personal pero normalmente nunca fuera de él.

Listado del contenido de directorios: comando ls

Una de las acciones más habituales a la hora de trabajar es mostrar el contenido de un directorio. Cómo se ha visto, existen herramientas gráficas con este fin, no obstante el *shell* incluye un programa con este mismo fin: **ls**

- ls** Muestra los nombres de los archivos y subdirectorios contenidos en el directorio en el que se está. Sólo se obtienen los nombres de los archivos, sin ninguna otra información.
- ls -a** Muestra todos los archivos incluyendo algunos que ordinariamente están ocultos para el usuario (aquellos que comienzan por un punto) Se recuerda que el archivo punto `.` indica el directorio actual y el doble punto `..` el directorio padre, que contiene, al actual.
- ls -l** Esta es la opción de lista larga: Muestra toda la información de cada archivo incluyendo: protecciones, tamaño y fecha de creación o del último cambio introducido,...
- ls -c** Muestra ordenando por día y hora de creación.
- ls -t** Muestra ordenando por día y hora de modificación.
- ls -r** Muestra el directorio y lo ordena en orden inverso.
- ls *subdir*** Muestra el contenido del subdirectorio ***subdir***.
- ls -l *filename*** Muestra toda la información sobre el archivo.
- ls -color** Muestra el contenido del directorio coloreado.

Las opciones anteriores pueden combinarse. Por ejemplo:

ls -cr Muestra el directorio ordenando inversamente por fechas.

El comando **ls** admite los caracteres de sustitución o meta caracteres (*) y (?). El carácter * representa cualquier conjunto o secuencia de caracteres. El carácter ? representa cualquier carácter, pero sólo uno. Por ejemplo:

- ls *.gif** Muestra todos los nombres de archivos que acaben en **.gif**, por ejemplo **dib1.gif, a.gif**, etc.
- ls file?** Muestra todos los archivos cuyos nombres empiecen por **file** y tengan un nombre de cinco caracteres, por ejemplo: **file1, file2, file3**, etc.

Creación de subdirectorios: Comando mkdir

El comando *mkdir* (*make directory*) permite a cada usuario crear un nuevo subdirectorio:

```
mkdir subdir1
```

donde *subdir* es el nombre del directorio que se va a crear.

Borrado de subdirectorios: Comando rmdir

Este comando borra uno o más directorios del sistema (*remove directory*), siempre que estos subdirectorios estén **vacíos**. Por ejemplo:

```
rmdir subdir1
```

donde **subdir** es el nombre del directorio que se va a eliminar.

Cambio de directorio: Comando cd

Este comando permite cambiar de directorio a partir del directorio actual de trabajo. Por ejemplo:

cd /home/Pedro En este ejemplo pasamos del directorio actual de trabajo al nuevo directorio **/home/Pedro**, que será desde ahora nuestro nuevo directorio.

cd dire Nos traslada al subdirectorio **dire** (que deberá existir como subdirectorio en el directorio actual)

cd .. Retrocede un nivel en la jerarquía de directorios. Por ejemplo, si estamos en **/home/Pedro** y usamos este comando, pasaremos al escalafón inmediatamente superior de la jerarquía de directorios, en este caso a **/home**. **Nota:** al contrario que en **MS-DOS** en Linux no

existe la forma **cd..** sin espacio entre **cd** y los dos puntos.

Cd

Nos sitúa nuevamente en el directorio personal del usuario.

Situación actual: Comando pwd

El comando **pwd** (**print working directory**) visualiza o imprime la ruta del directorio en el que nos encontramos en este momento. Este comando es uno de los pocos que no tiene opciones y se utiliza escribiendo simplemente **pwd**.

Acceso a unidades de disco

Linux a diferencia de Windows no utiliza letras ("a:", "c:", "d:", ...) para acceder a las distintas unidades de disco de una computadora. En Linux para acceder al contenido de una unidad de disco o de un CD-ROM, éste tiene que haber sido previamente "*montado*". El *montado* se realiza mediante el comando *mount*, con lo que el contenido de la unidad se pone a disposición del usuario en el directorio de Linux que se elija. Por ejemplo para acceder al CD-ROM se teclearía el siguiente comando:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

donde *-t iso9660* indica el tipo de sistema que usa la unidad de disco para guardar los archivos (las más usuales son: *iso9660* en el caso de un CD-ROM, *vfat* en el caso de *Windows*, y *ext2* en el caso de Linux), */dev/cdrom* indica el dispositivo que se va a montar. Todos los dispositivos están representados por un archivo del directorio */dev*; por ejemplo, en el caso de un disquete será seguramente */dev/fd0*, por último */mnt/cdrom* es el directorio en el que se pondrá a disposición del usuario el contenido del CD-ROM. Para montar disquetes se suele utilizar el directorio */mnt/floppy*.

De todas formas el usuario siempre puede crear un directorio vacío con el nombre que él elija para montar las unidades de disco que desee y donde desee. Cuando el usuario haya dejado de usar ese disco deberá "*desmontarlo*" mediante el comando *umount* antes de sacar el disquete o el CD-ROM. En este último caso debería escribir:

```
umount /mnt/cdrom
```

Para utilizar el comando *mount* de la forma anterior hace falta ser administrador o *root*. Para que un usuario común pueda utilizar disquetes, CD-ROM, etc. hay que editar el archivo */etc/fstab* Por ejemplo para que cualquier usuario pueda acceder a un disquete habrá que indicar la siguiente línea:

```
/dev/fd0 /mnt/floppy vfat user, noauto 0 0
```

También habrá que asegurarse de que el directorio */mnt/floppy* sea accesible por todos los usuarios. Una vez seguidos los pasos anteriores cualquier usuario podrá "montar" un disquete escribiendo el siguiente comando:

```
mount /mnt/floppy
```

Al igual que antes el usuario deberá ejecutar el comando ***umount /mnt/floppy*** antes de sacar el disquete.

Nota: Existen en la actualidad distribuciones (p. Ej. Linux Mandrake) que realizan este proceso de forma automática por lo que las unidades de disquete y CD-ROM quedan accesibles a todos los usuarios de una forma sencilla, empleando los comandos:

```
mount /mnt/floppy
umount /mnt/floppy
```

siempre que */mnt/floppy* sea la ruta adecuada.

Copia de archivos: Comando cp

Este comando tiene la siguiente forma:

```
cp file1 file2
```

y hace una copia de *file1* y le llama *file2*. Si *file2* no existía, lo crea con los mismos atributos de *file1*. Si *file2* existía antes, su contenido queda destruido y es sustituido por el de *file1*.

El archivo *file2* estará en el mismo directorio que *file1*. Tanto *file1* como *file2* indican el nombre de un archivo, que puede incluir en la ruta al mismo si alguno de ellos no se encuentra en el directorio actual.

Otra posibilidad es:

```
cp file1 file2 namedir
```

que hace copias de *file1* y *file2* en el directorio *namedir*.

Traslado y cambio de nombre de archivos: Comando mv

Este comando tiene una forma similar al anterior:

```
mv file1 file2
```

El comando *mv* realiza la misma función que el anterior (*cp*) pero además destruye el archivo original. En definitiva traslada el contenido de *file1* a *file2*; a efectos del usuario lo que ha hecho es cambiar el nombre a *file1*, llamándole *file2*. De igual forma:

```
mv file1 file2 namedir
```

traslada uno o más archivos (*file1*, *file2*,...) al directorio *namedir* conservándoles el nombre. El comando:

```
mv namedir1 namedir2
```

cambia el nombre del subdirectorio *namedir1* por *namedir2*.

Hay que recalcar que el comando *mv* sirve así mismo para cambiar el nombre de los archivos.

Enlaces a archivos: Comando ln

En Linux un mismo archivo puede estar repetido con más de un nombre, ya que con el comando *cp* se pueden realizar cuantas copias se desee del archivo. Pero, a veces, es más práctico tener un mismo archivo con varios nombres distintos, y lo que es más importante, poder acceder a él desde más de un directorio. En Linux esto recibe el nombre de enlaces múltiples a un archivo. El ahorro de espacio de disco es importante al poder compartir un archivo más de un usuario. Estos enlaces son muy prácticos a la hora de utilizar archivos que pertenecen a directorios distintos. Gracias a los enlaces se puede acceder a muchos archivos desde un mismo directorio, sin necesidad de copiar en ese directorio todos esos archivos. La forma de este comando es:

```
ln file1 file2
```

A partir de este momento el archivo *file1* tiene dos nombres: *file1* y *file2*. A diferencia de los comandos *cp* y *mv*, este comando toma más precauciones, ya que advierte previamente si el nombre *file2* está ocupado, en este caso no se ejecuta.

```
In report subdir/report
```

Después de este comando el archivo *report* tendrá el mismo nombre, pero a efectos del usuario estará colocado en dos sitios distintos: en el directorio actual y en el subdirectorio *subdir*. Los archivos enlazados a otro se borran como los archivos normales. Si se borra el archivo original permanece su contenido en los archivos enganchados.

Borrado de archivos: Comando rm

Este comando tiene las formas siguientes:

```
rm file1 file2
```

Este comando elimina uno o más archivos de un directorio en el cual tengamos permiso de escritura. Con este comando resulta facilísimo borrar archivos inútiles, y desgraciadamente, también los útiles. Por eso es conveniente y casi imprescindible emplear la opción *-i*, de la forma siguiente:

```
rm -i file1 file2
```

Con esta opción, Linux pedirá confirmación para borrar cada archivo de la lista, si realmente se desea su destrucción o no. Se recomienda usar siempre este comando con esta opción para evitar el borrado de archivos útiles. Por ejemplo, si se teclea:

```
rm -i superfluo
```

aparecerá en pantalla el aviso siguiente:

```
remove superfluo?
```

y habrá que contestar **y** (yes) o **n** (not). En este comando se pueden utilizar los caracteres de sustitución (* y ?), como por ejemplo:

```
rm fich*
```

que borraría todos los archivos del directorio actual que comiencen por **fich**. El comando **rm *** borraría todos los archivos del directorio actual, mientras que

```
rm -i *
```

realiza una labor análoga, pero con previa confirmación.

Características de un archivo: Comando file

Este comando realiza una serie de comprobaciones en un archivo para tratar de clasificarlo. Su formato es:

```
file fich
```

Tras su ejecución este comando muestra el tipo del archivo e información al respecto del mismo.

Cambio de modo de los archivos comandos chmod, chown y chgrp

Los permisos de cada archivo se pueden ver con el comando `ls -l`. Para cambiar los permisos de un archivo se emplea el comando `chmod`, que tiene el formato siguiente:

chmod [quien] oper permiso files

quien Indica a quien afecta el permiso que se desea cambiar. Es una combinación cualquiera de las letras **u** para el usuario, **g**

	para el grupo del usuario, o para los otros usuarios, y a para todos los anteriores. Si no se da el quien , el sistema supone a .
oper	Indica la operación que se desea hacer con el permiso. Para dar un permiso se pondrá un + , y para quitarlo se pondrá un - .
Permiso	Indica el permiso que se quiere dar o quitar. Será una combinación cualquiera de las letras anteriores: r, w, x, s .
files	Nombres de los archivos cuyos modos de acceso se quieren cambiar.

Por ejemplo, para quitar el permiso de lectura a los usuarios de un archivo el comando es:

```
chmod a -r archivo.txt
```

Los permisos de lectura, escritura y ejecución tienen un significado diferente cuando se aplican a directorios y no a archivos normales. En el caso de los directorios el permiso **r** significa la posibilidad de ver el contenido del directorio con el comando **ls**; el permiso **w** da la posibilidad de crear y borrar archivos en ese directorio, y el permiso **x** autoriza a buscar y utilizar un archivo concreto. Por otra parte, el comando *chown* se emplea para cambiar de propietario (“**change owner**”) a un determinado conjunto de archivos. Este comando sólo le puede emplear el actual propietario de los mismos. Los nombres de propietario que admite Linux son los nombres de *usuario*, que están almacenados en el archivo */etc/passwd*. La forma general del comando *chown* es la siguiente:

```
chown newowner file1 file2 ...
```

Análogamente, el grupo al que pertenece un archivo puede ser cambiado con el comando *chgrp*, que tiene una forma general similar a la de *chown*,

```
chgrp newgroup file1 file2...
```

Los grupos de usuarios están almacenados en el archivo **/etc/group**

Espacio ocupado en el disco comandos **du** y **df**

El comando **du** permite conocer el espacio ocupado en el disco por un determinado directorio y todos los subdirectorios que cuelgan de él. Para usarlo basta simplemente colocarse en el directorio adecuado y teclear, **du**, éste comando da el espacio de disco utilizado en bloques. Para obtener la información en bytes se debe emplear el comando con la opción **-h**: **du -h**

El comando **df** por el contrario informa del espacio usado por las particiones del sistema que se encuentren montadas.

Visualización sin formato de un archivo: Comando **cat**

Este comando permite visualizar el contenido de uno o más archivos de forma no formateada. También permite copiar uno o más archivos como apéndice de otro ya existente. Algunas formas de utilizar este comando son las siguientes:

- cat filename** Saca por pantalla el contenido del archivo *filename*.
- cat file1 file2...** Saca por pantalla, secuencialmente y según el orden especificado, el contenido de los archivos indicados.
- cat file1 file2 >file3** El contenido de los archivos *file1* y *file2* es almacenado en *file3*.
- cat file1 file2 >>file3** El contenido de *file1* y *file2* es añadido al final de *file3*.
- cat >file1** Acepta lo que se introduce por el teclado y lo almacena en *file1* (se crea *file1*) Para terminar se emplea **<ctrl>d**

Comando head

```
head -7 filename
```

escribe las 7 primeras líneas del archivo filename.

Visualización de archivos con formato: Comando pr

Este comando, a diferencia de *cat*, imprime por consola el contenido de los archivos de una manera formateada, por columnas, controlando el tamaño de página y poniendo cabeceras al comienzo de las mismas. Está muy en relación con el comando *lp* de salida por impresora. Las formas más importantes que admite son las siguientes:

- pr file*** Produce una salida estándar de 66 líneas por página, con un encabezamiento de 5 líneas (2 en blanco, una de identificación y otras 2 líneas en blanco)
- pr -ln file*** Produce una salida de n líneas por página (cuando el tamaño de papel de impresora, por ejemplo, tiene un número de líneas distinto de 66)
- pr -p file*** Hace una pausa para presentar la página, hasta que se pulsa **<return>** para continuar
- pr -t file*** Suprime las 5 líneas del encabezamiento y las del final de página.
- pr -wn file*** Ajusta la anchura de la línea a n posiciones.
- pr -d file*** Lista el archivo con espaciado doble.
- pr -h `caracteres` file*** el argumento o cadena de caracteres *`caracteres`* se convertirán en la cabecera del listado.
- pr +n file*** Imprime el archivo a partir de la página n.

Además de los ejemplos anteriores, se pueden combinar varias opciones en un mismo comando, como por ejemplo en: `pr -dt file` la salida de este comando es por la consola, pero puede redireccionarse a otro archivo, por ejemplo, si ejecutamos el comando: `pr file1 > file2` se crea un archivo nuevo llamado `file2` que es idéntico a `file1`, pero con formato por páginas y columnas.

Visualización de archivos pantalla a pantalla: Comandos `more` y `less`

Estos comandos permiten visualizar un archivo pantalla a pantalla. El número de líneas por pantalla es de 23 líneas de texto y una última línea de mensajes, donde aparecerá la palabra `more`. Cuando se pulsa la barra espaciadora (el espacio en blanco), se visualizará la siguiente pantalla. Para salir de este comando (terminar la visualización) se pulsa `<ctrl>d` o `q`. Por ejemplo: `more file`. El comando `less` es muy similar al anterior pero permite el desplazamiento a lo largo del texto empleando las teclas de cursores pudiendo desplazarse hacia arriba o abajo de un archivo.

Búsqueda en archivos: Comandos `grep`, `fgrep` y `egrep`

El comando `grep` localiza una palabra, clave o frase en un conjunto de directorios, indicando en cuáles de ellos la ha encontrado. Este comando rastrea archivo por archivo, por turno, imprimiendo aquellas líneas que contienen el conjunto de caracteres buscado. Si el conjunto de caracteres a buscar está compuesto por dos o más palabras separadas por un espacio, se colocará el conjunto de caracteres entre apóstrofes (') Su formato es el siguiente:

```
grep 'conjunto caracteres' file1 file2 file3
```

siendo 'conjunto caracteres' la secuencia de caracteres a buscar, y `file1`, `file2`, y `file3` los archivos donde se debe buscar. Veamos un nuevo ejemplo:

```
grep 'TRIANGULARIZACION MATRIZ' matrix.f scaling.f
```

Este comando buscará *TRIANGULARIZACION MATRIZ* entre las líneas de los archivos *matrix.f* y *scaling.f*. Este comando permite seleccionar, entre todas las líneas de uno o más archivos, aquellas que contienen un motivo que satisface una expresión regular determinada.

```
grep [-opción] expresión_regul ar [referencia...]
```

Las opciones principales son:

- c** lo único que se hace es escribir el número de las líneas que satisfacen la condición.
- i** no se distinguen mayúsculas y minúsculas.
- l** se escriben los nombres de los archivos que contienen líneas buscadas.
- n** cada línea es precedida por su número en el archivo.
- s** no se vuelcan los mensajes que indican que un archivo no se puede abrir.
- v** se muestran sólo las líneas que no satisfacen el criterio de selección.

A continuación se muestra una serie de ejemplos:

- **grep '^d' text** líneas que comienzan por d.
- **grep '[^d]' text** líneas que no comienzan por d.
- **grep -v '^C' file1 > file2** quita las líneas de file1 que comienzan por C y lo copia en file2.

Comandos tar y gzip

Tanto el comando **tar** como **gzip** son ampliamente empleados para la difusión de programas y archivos en Linux. El primero de ellos agrupa varios archivos en uno solo o “*archivo*”, mientras que el segundo los comprime. En conjunto estos dos

programas actúan de forma muy similar a programas como *Winzip*. Para crear un nuevo archivo se emplea:

```
tar -cvf nombre_archivo.tar archivo1 archivo2 ...
```

donde *archivo1*, *archivo2* etc. son los archivos que se van a añadir al archivo ***tar***. Si se desea extraer los archivos se emplea:

```
tar -xpvf nombre_archivo.tar archivo1 ...
```

Al contrario que *tar* que agrupa varios archivos en uno, *gzip* comprime un único archivo con lo que la información se mantiene pero se reduce el tamaño del mismo. El uso de *gzip* es muy sencillo:

```
gzip archivo
```

con lo que se comprime *archivo* (que es borrado) y se crea un archivo con nombre *archivo.gz*. Si lo que se desea es descomprimir un archivo se emplea entonces:

```
gzip -d archivo.gz
```

recuperando el archivo inicial. Como se ha comentado al principio es típico emplear *tar* y *gzip* de forma consecutiva, para obtener archivos con extensión *tar.gz* o *tgz* que contienen varios archivos de forma comprimida (similar a un archivo *zip*) El comando *tar* incluye la opción *z* para estos archivos de forma que para extraer los archivos que contiene:

```
tar -zxf archivo.tar.gz
```

Comando lpr

El comando *lpr* se emplea para imprimir una serie de archivos. Si se emplea sin argumentos imprime el texto que se introduzca a continuación en la impresora por defecto. Por el contrario:

```
lpr nombre_archivo
```

imprime en la impresora por defecto el archivo indicado.

1.6. Como funciona la integración de una base de datos en Internet.

Para realizar una requisición de acceso desde el Web hasta una base de datos no sólo se necesita de un navegador de Internet y de un servidor Web, sino también de una serie de scripts de procesamiento, que para nuestro caso están desarrollados en PHP, los cuales son llamados directamente desde un documento HTML en el cliente. Dichos scripts leen la entrada de datos desde que provienen del cliente y toma cierta información de variables de ambiente.

Una vez que se reciben los datos de entrada (sentencias SQL o piezas de ellas), los scripts de procesamiento los prepara para enviarlos a la interfaz en forma de SQL, y luego ésta procesa los resultados que se extraen de la base de datos.

La interfaz contiene las especificaciones de la base de datos necesarias para traducir las solicitudes enviadas desde el cliente, a un formato que sea reconocido por dicha base. Además, contiene toda la información, estructuras, variables y llamadas a funciones, necesarias para comunicarse con la base de datos.

El software de acceso usualmente es el software distribuido con la base de datos, el cual permite el acceso a la misma, a través de solicitudes con formato. Luego, el software de acceso recibe los resultados de la base de datos, aún los mensajes de

error, y los pasa hacia la interfaz, y ésta a su vez, los pasa hasta los scripts de procesamiento. Ahora podemos hacer una definición y análisis de cada una de las herramientas que durante el desarrollo del proyecto vamos a utilizar, para poder proporcionar una visión general de los elementos que lo conforman.

1.6.1. Servidor APACHE.

Apache es un servidor Web de código libre, robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la Web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Estos voluntarios se conocen como el Apache Group. Además del Apache Group, cientos de personas han contribuido al proyecto con código, ideas y documentación.

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache.
- **Módulos Multiproceso:** Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los procesos hijos a atender a las peticiones.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Módulos Base y Módulos Multiproceso:

core: Funciones básicas del Apache que están siempre disponibles.

mpm_common: Colección de directivas que se implementan en más de un módulo multiproceso.

beos: Módulo de multiproceso optimizado para BeOS.

leader: Variable experimental de MPM.

mpm_netware: Módulo de multiproceso que implementa un servidor Web optimizado para Novell NetWare.

mpmt_os2: MPM híbrido, multiproceso y multihilo para OS/2.

perchild: Módulo multiproceso que permite a los procesos demonio servir las peticiones que se asignan a distintos id de usuario.

prefork: Implementa un servidor sin hilos.

threadpool: Variante experimental del módulo estándar de MPM.

mpm_winnt: Módulo multiproceso optimizado para Windows NT.

worker: Módulo multiproceso que implementa un híbrido multihilos y multiprocesos de servidor Web.

Módulos adicionales:

mod_access: Proporciona control de acceso basándose en el nombre del host del cliente, su dirección IP u otras características de la petición del cliente.

mod_actions: Este módulo se utiliza para ejecutar Scripts CGI, basándose en el tipo de medio o el método de petición.

mod_alias: Proporcionado para mapear diferentes partes del sistema de archivos del servidor en el árbol de documentos del servidor, y para redirección de URL's.

mod_asis: Envío de archivos que tienen sus propias cabeceras http.

mod_auth: Autenticación de usuario utilizando archivos de texto.

mod_auth_anon: Permite a usuarios anónimos acceder a áreas autenticadas.

mod_auth_dbm: Proporciona autenticación utilizando archivos DBM.

mod_auth_digest: Autenticación de usuario utilizando MD5.

mod_auth_ldap: Permite la utilización un directorio LDAP para almacenar la base de datos de autenticación.

mod_autoindex: Muestra los contenidos de un directorio automáticamente, parecido al comando ls de Unix.

mod_cache: Cache de contenidos indexados por URL's.

mod_cern_meta: Semántica de etiquetas meta del CERN.

mod_cgi: Ejecución de Scripts CGI.

mod_cgid: Ejecución de Scripts CGI utilizando un demonio CGI externo.

mod_charset_lite: Para la especificación del juego de caracteres de las traducciones.

mod_deflate: Comprime el contenido antes de ser enviado al cliente.

mod_dir: Proporcionado para redirecciones y para servir los archivos de listado de directorios.

mod_disk_cache: Cache para almacenar contenidos identificados por URL.

mod_echo: Un servidor simple de echo para ilustrar los módulos del protocolo.

mod_env: Modificación del entorno que se envía a los scripts CGI y las páginas SSI.

mod_expires: Generación de las cabeceras http Expires, de acuerdo de los criterios especificados por el usuario.

mod_ext_filter: Pasa el cuerpo de la respuesta a través de un programa antes de enviársela al cliente.

mod_file_cache: Almacena en memoria cache una lista estática de archivos en memoria.

mod_headers: Personalización de las peticiones HTTP y las cabeceras de las

respuestas.

mod_imap: Proceso de imágenes en el lado del servidor.

mod_include: Documentos HTML generados por el servidor (Server Side Includes).

mod_info: Proporciona una visión comprensiva de la configuración del servidor.

mod_isapi: Extensiones ISAPI en Apache para Windows.

mod_ldap: Pool de conexiones LDAP y cacheo de resultados para la utilización de otros módulos LDAP.

mod_log_config: Registro de las peticiones hechas al servidor.

mod_logio: Registro del número de bytes recibidos y enviados en cada respuesta.

mod_mem_cache: Cache de contenidos identificados por URI.

mod_mime: Asocia las extensiones de peticiones de los archivos con el comportamiento del archivo (manejadores y filtros) y contenido (tipos mime, idioma, juego de caracteres y codificación).

mod_mime_magic: Determina el tipo MIME de un archivo mirando unos pocos bytes del contenido.

mod_negotiation: Se proporciona para la negociación del contenido.

mod_proxy: servidor HTTP/1.1 proxy/gateway.

mod_proxy_connect: Extensión de mod_proxy para la gestión de las peticiones CONNECT.

mod_proxy_ftp: Soporte FTP para mod_proxy.

mod_proxy_http: Soporte HTTP para el módulo mod_proxy.

mod_rewrite: Proporciona un motor de reescritura basado en reglas que rescribe las peticiones de URL's al vuelo.

mod_setenvif: Permite la configuración de las variables de entorno basándose en las características de la petición.

mod_so: Carga del código ejecutable y los módulos en al iniciar o reiniciar el servidor.

mod_speling: Intenta corregir las URL mal puestas por los usuarios, ignorando las mayúsculas y permitiendo hasta una falta.

mod_ssl: Criptografía avanzada utilizando los protocolos Secure Sockets Layer y Transport Layer Security.

mod_status: Proporciona información en la actividad y rendimiento del servidor.

mod_suexec: Permite a los scripts CGI ejecutarse con un nombre y grupo específico.

mod_unique_id: proporciona variables de entorno y un identificador único para cada petición.

mod_userdir: Directorios específicos para usuarios.

mod_usertrack: Registro de actividad de un usuario en el sitio.

mod_vhost_alias: Proporcionado para configurar muchos servidores virtuales dinámicamente.

Básicamente, las configuraciones del servidor residen dentro de dos archivos, el de configuración principal que se encuentra dentro de la carpeta conf, con el nombre httpd.conf, o dentro de un archivo con el nombre .htaccess que se puede encontrar dentro de cualquier directorio que se encuentre mapeado dentro del servidor. Dentro de cada uno de estos archivos se ubican las directivas de configuración. Muchas de estas directivas de configuración se pueden encontrar tanto dentro del archivo de configuración principal como dentro de un archivo .htaccess. Los valores de las directivas que se encuentran dentro de un archivo .htaccess, prevalecen frente a los valores de configuración especificados dentro del archivo httpd.conf.

El archivo httpd.conf

El archivo httpd.conf es el archivo principal de configuración del Apache, se encuentra dentro del directorio Conf, en el directorio de instalación del Apache. En primer lugar hay que destacar que el archivo está dividido en tres secciones, que son:

1. Parámetros globales
2. Directivas de Funcionamiento
3. Host Virtuales

En el archivo se encuentran todos los parámetros de funcionamiento del Apache. Algunos parámetros son generales para la instalación y funcionamiento del Apache. Muchos otros de los parámetros se pueden configurar independientes para un conjunto de directorios y/o archivos. En estos casos los parámetros se encuentran ubicados dentro de secciones donde se indica el ámbito de aplicación del parámetro.

Las secciones más importantes son:

<Directory>: Los parámetros que se encuentran dentro de esta sección, sólo se aplicarán al directorio especificado y a sus subdirectorios.

<DirectoryMatch>: Igual que Directory, pero acepta en el nombre del directorio expresiones regulares.

<Files>: Los parámetros de configuración proporcionan control de acceso de los archivos por su nombre.

<FilesMatch>: Igual que Files, pero acepta expresiones regulares en el nombre del archivo.

<Location>: Proporciona un control de acceso de los archivos por medio de la URL.

<LocationMatch>: Igual que Location, pero acepta expresiones regulares en el nombre del archivo.

Algunas veces las directivas de funcionamiento de las secciones anteriores se pueden cruzar en cuyo caso tienen el siguiente orden de preferencia:

1. <Directory> y .htaccess (.htaccess prevalece frente a <Directory>)
2. <DirectoryMatch> y <Directory>
3. <Files> y <FilesMatch>
4. <Location> y <LocationMatch>

También hay que destacar, que el archivo contiene una serie de comentarios para su correcta utilización, las líneas comentadas aparecen con el símbolo #.

Httpd.conf: Parámetros globales

Todos los parámetros que se establecen dentro de esta sección son globales para el funcionamiento del servidor, por lo que no admiten estar dentro de ninguna directiva.

ServerRoot: especifica la ubicación del directorio raíz donde se encuentra instalado el Apache, a partir del cual se crea el árbol de directorios. Esta directiva no debería cambiar a no ser que se mueva la carpeta de instalación de apache a otro directorio. Se encuentra disponible a través del módulo Core.

PidFile: ubicación del archivo que contendrá el número de identificación del proceso cuando se encienda el servidor. Se encuentra disponible a través de varios módulos beos, leader, mpm_winnt, mpmt_os2, perchild, prefork, threadpool ó worker.

TimeOut: el valor se utiliza para configurar medido en segundos, tres parámetros:

1. El tiempo tal que puede tardar una petición en ser recibida entera
2. La cantidad de tiempo que espera entre recepción de paquetes TCP
3. La cantidad de tiempo entre ACK's en transmisiones TCP

Pasado este tiempo se produce un mensaje de error en el que se indica que se ha consumido el tiempo máximo de espera. Establecer un valor muy pequeño puede dar lugar a que los usuarios reciban este mensaje de error, y establecer un valor muy alto dará lugar a una sobrecarga de la máquina. Se encuentra disponible a través del módulo Core.

KeepAlive: especifica si se utilizarán conexiones persistentes, es decir, que todas las peticiones de un usuario se atenderán con la misma conexión. Se encuentra disponible a través del módulo Core.

MaxKeepAliveRequests: número máximo de conexiones persistentes. (Número máximo de usuarios concurrentes si KeepAlive esta en ON). Para establecer este parámetro, hay que tener en cuenta el ancho de banda de salida de nuestro servidor, por el cual deberá ser enviada toda la información, si se establece un valor muy grande respecto al ancho de banda, el tiempo de respuesta se verá incrementado para cada usuario. Se encuentra disponible a través del módulo Core.

KeepAliveTimeout: tiempo que espera en segundos entre peticiones de un usuario, antes de considerar que este ha terminado, y cerrar su conexión. Si el valor es muy pequeño provocará que algunos usuarios no puedan visualizar la página debido a que el número máximo de conexiones persistentes se ha superado, mientras que si se establece un valor muy grande se estarán utilizando muchos recursos de la máquina. Se encuentra disponible a través del módulo Core.

Listen: esta directiva permite especificar que puerto se utilizará para atender las peticiones. Por defecto se utiliza el puerto 80 (WWW), también permite especificar que direcciones IP atenderá, por defecto todas. Para atender dos direcciones IP distintas, con distintos puerto, se utilizaría:

Listen 192.168.255.5:80

Listen 192.168.255.8:8080

Se encuentra disponible a través de varios módulos `beos`, `leader`, `mpm_winnt`, `mpmt_os2`, `perchild`, `prefork`, `threadpool` ó `worker`.

LoadModule: Directiva que sirve para cargar módulos que incluyen distintas funcionalidades. La Sintáxis es:

`LoadModule nombreModulo ubicacionArchivo`

Se encuentra disponible a través del módulo `mod_so`.

Httpd.conf: Directivas de funcionamiento (1)

Esta es la sección principal de configuración del servidor, en ella podemos encontrar las siguientes opciones:

ServerAdmin: Especifica la dirección de correo electrónico del administrador, esta dirección aparece en los mensajes de error, para permitir al usuario notificar un error al administrador. No puede estar dentro de ninguna sección.

Se encuentra disponible a través del módulo `Core`.

ServerName: Especifica el nombre y el puerto que el servidor utiliza para identificarse, normalmente se determina automáticamente, pero es recomendable especificarlo explícitamente para que no haya problemas al iniciar el servidor.

Si el servidor no tiene un nombre registrado en las DNS, se recomienda poner su número IP. No puede estar dentro de ninguna sección.

La sintáxis es:

ServerName direccionIP:Puerto p.e. ServerName localhost:80

Se encuentra disponible a través del módulo Core.

DocumentRoot: La carpeta raíz que se ubica en el servidor, desde la que se servirán los documentos. Por defecto, todas las peticiones, tendrán como raíz esta carpeta, a no ser que se utilicen alias (directorios virtuales en IIS).

Por defecto, la carpeta raíz es la carpeta Htdocs, que se encuentra en la carpeta de instalación del Apache. No puede estar dentro de ninguna sección.

Si se cambia este directorio por otro, es muy importante que se ponga el nuevo valor, no solo en esta línea, sino también en la sección <Directory> en la que se establecen los parámetros de configuración de este directorio.

Esta línea empieza por " <Directory " seguido de la carpeta raíz que originalmente hay en DocumentRoot. Se encuentra disponible a través del módulo Core.

DirectoryIndex: Especifica el archivo por defecto que buscará en cada directorio, en caso de que no se especifique ninguno. Por defecto es index.html. Es decir, que si por ejemplo se pone en el navegador: www.miweb.com el servidor por defecto servirá www.miweb.com/index.html.

En esta directiva se pueden especificar más de un archivo, la Sintáxis es la siguiente:

DirectoryIndex archivo1 archivo2 archivo3

El orden con el que se especifica el nombre de archivo determinará la prioridad a la hora de decidir que archivo es el que se muestra. La directiva se puede

encontrar fuera de cualquier sección, dentro de una sección o dentro de un archivo .htaccess. Se encuentra disponible a través del módulo mod_dir.

AccessFileName: Es el nombre del archivo de configuración que se buscará en cada una de los directorios del servidor para conocer la configuración del mismo. Este archivo permite configurar el comportamiento de cada uno de los directorios individualmente. Para que esta configuración funcione, la directiva AllowOverride tiene que tener un valor que lo permita. No puede estar dentro de ninguna sección.

El nombre de archivo que se especifica por defecto es el del archivo ".htaccess".

Como medida de seguridad, la configuración del Apache establece que no se muestre la existencia de este archivo a ningún usuario, aunque este establecida la opción de listado de directorios. Si se decide cambiar al nombre, habrá que redefinir la seguridad para que no se muestre el contenido del nuevo archivo. Esto se hace en el archivo httpd.conf en una sección File como la que se presenta a continuación en la que se establece que todos los archivos que comiencen por .ht no se mostrarán.

```
<Files ~ "\.ht">  
    Order allow,deny  
    Deny from all  
</Files>
```

Se encuentra disponible a través del módulo Core.

Httpd.conf: Directivas de funcionamiento (2)

TypesConfig: Especifica el nombre del archivo que contiene la lista de tipos mime que conoce el servidor, y que determinará dependiendo de las extensiones para generar las cabeceras http. No puede estar dentro de ninguna sección.

Se encuentra disponible a través del módulo mod_mime.

DefaultType: Tipo mime que se servirá por defecto en caso de no conocer la extensión del archivo que se está sirviendo. Por defecto, se indicará que se sirve texto plano, con el valor text/plain. La directiva se puede encontrar fuera de cualquier sección, dentro de una sección o dentro de un archivo .htaccess.

Sintáxis:

```
DefaultType tipoMime
```

Se encuentra disponible a través del módulo Core.

HostnameLookups: Se utiliza en los archivos de registro. Por defecto cuando se produce un acceso, se guarda simplemente su número IP, si esta directiva se encuentra en On, el servidor buscará la correspondencia de ese número IP con su nombre, y almacenará el nombre. Establecer esta configuración en ON provocará que por lo menos se tenga que hacer una petición al servidor de nombres por cada una de las peticiones de usuario, por lo que el rendimiento de la máquina se puede verse en decremento. Esta directiva se puede encontrar dentro de una sección o fuera de cualquier otra. Se encuentra disponible a través del módulo Core.

ErrorLog: Especifica la ubicación del archivo que contiene el registro de errores, por defecto en la carpeta logs. Esta directiva sólo se puede encontrar fuera de cualquier sección. Se encuentra disponible a través del módulo Core.

LogLevel: Especifica el tipo de mensajes que se guardaran en el archivo de registro de errores, dependiendo de los valores especificados, se guardarán mas o menos. Esta directiva sólo se puede encontrar fuera de cualquier sección.

Valor de más a menos son: debug, info, notice, warn, error, crit, alert, emerg

Se encuentra disponible a través del módulo Core.

LogFormat: La directiva permite definir el formato que se utilizará para almacenar los registros. A cada formato se le puede asignar un nombre, utilizándolo luego para crear distintos tipos de archivos de registro. Pueden existir varios logFormat distintos.

Sintáxis:

LogFormat "configuraciónError" nombre

Esta directiva se encuentra fuera de cualquier sección.

Se encuentra disponible a través del módulo mod_log_config.

CustomLog: La directiva se utiliza para especificar la ubicación y el tipo de formato que se utilizará en un archivo de registro. Pueden existir varios archivos de registro distintos con configuraciones distintas. Para hacer esto, simplemente hay que poner varias líneas customlog.

Sintáxis:

CustomLog archivo formato

Esta directiva se encuentra fuera de cualquier sección.

Se encuentra disponible a través del módulo mod_log_config.

ServerTokens: Esta directiva establece la información que se devuelve dentro de

la cabecera http que envía el servidor. Posibles valores de menor a mayor información son:

- **Pord**
- **Min**
- **Os**
- **Full**

Esta directiva se encuentra fuera de cualquier sección. Se encuentra disponible a través del módulo Core.

IndexOptions: Esta directiva controla la apariencia de la página que se mostrará a un usuario cuando se pide la lista de archivos de un directorio.

Sintáxis:

IndexOptions [+|-]opción [[+|-]opción] ... (Apache 1.3.3 en adelante)

Entre las opciones que se pueden poner, destaca:

FancyIndexing: que muestra los nombres de los archivos, con iconos etc.

Se encuentra disponible a través del módulo mod_autoindex.

FoldersFirst: Hace que primero se muestren los directorios. Esta opción sólo se puede establecer en el caso de que FancyIndexing este activa. Esta directiva se puede encontrar dentro del archivo .htaccess, dentro de una sección <Directory> y fuera de cualquier otra.

Se encuentra disponible a través del módulo mod_autoindex.

Httpd.conf: Directivas de funcionamiento (3)

AddIconByEncoding: Esta directiva permite asociar un icono a un tipo mime, de forma que cuando la directiva fancyIndexing este activada, se mostrará al lado del archivo el icono correspondiente.

Sintáxis:

```
AddIconByEncoding icon MIME-encoding...
```

Ejemplo:

```
AddIconByEncoding/icons/compressed.gif x-compress
```

Esta directiva se puede encontrar dentro del archivo .htaccess, dentro de una sección <Directory> y fuera de cualquier otra. Se encuentra disponible a través del módulo mod_autoindex.

AddIconByType: Esta directiva asocia un icono a un archivo dependiendo del un tipo mime, de forma que cuando la directiva fancyIndexing este activada, se mostrará al lado del archivo el icono correspondiente.

Sintáxis:

```
AddIconByType icon MIME-encoding...
```

Ejemplo:

```
AddIconByType /icons/text.gif text/*
```

La diferencia entre AddIconByType y AddIconByEncoding reside en que mientras que en la primera se determina el tipo mime mediante basándose en la

codificación del archivo, mientras que `AddIconByType` determina el tipo mime basándose en el nombre del archivo.

Ambas directivas se pueden encontrar dentro del archivo `.htaccess`, dentro de una sección `<Directory>` o fuera de cualquier otra. Se encuentra disponible a través del módulo `mod_autoindex`.

AddDescription: Esta directiva permite asociar una descripción a un tipo de archivo, que se mostrará al listar un directorio. Esta directiva se puede encontrar dentro de el archivo `.htaccess`, dentro de una sección `<Directory>` o fuera de cualquier otra.

Sintáxis:

`AddDescription cadena, archivo`

Se encuentra disponible a través del módulo `mod_autoindex`.

AddDefaultCharset: Esta directiva define la codificación de caracteres que se utilizará de forma predeterminada para los documentos. Por defecto viene establecido el valor `ISO-8859-1`. Esta directiva se puede encontrar dentro de cualquier sección y en los archivos `.htaccess`. Se encuentra disponible a través del módulo `Core`.

ErrorDocument: Esta directiva establece la configuración del servidor para cuando se produce un error. Se pueden establecer cuatro configuraciones distintas:

1. Mostrar un texto de error
2. Redirigir a un archivo en el mismo directorio
3. Redirigir a un archivo en nuestro servidor

4. Redirigir a un archivo fuera de nuestro servidor

Hay que tener en cuenta que si el texto de error se envía a Internet Explorer, este tendrá que tener al menos 512 Bytes, porque sino Internet Explorer mostrará su propia página de error.

Sintáxis:

```
ErrorDocument NúmeroError Acción
```

Esta directiva se puede encontrar tanto dentro del archivo .htaccess, dentro de la sección <Directory> o fuera de cualquier otra sección.

Ejemplo:

```
ErrorDocument 404 /error404.html.
```

En caso de no encontrarse un archivo, se mostrará el archivo error404.html

Se encuentra disponible a través del módulo Core.

CacheRoot: Establece el directorio donde se encontrarán los archivos de la cache del Apache.

Se encuentra disponible a través del módulo mod_disk_cache

CacheSize: Tamaño de la cache en Kbytes. Se encuentra disponible a través del módulo mod_disk_cache

CacheGclInterval: Establece cada cuantas horas se verificará el tamaño de los archivos del cache para comprobar si se corresponden con el tamaño establecido dentro de CacheSize.

El valor acepta números flotantes, por lo que se pueden establecer los intervalos en minutos. Cuanto mayor sea el valor de esta directiva, más posibilidades existirán de que se sobrepase el valor establecido en CacheSize.

Se encuentra disponible a través del módulo `mod_disk_cache`

CacheMaxExpire: Máximo número de horas que los archivos permanecerán dentro de la cache.

Se encuentra disponible a través del módulo `mod_cache`.

CacheLastModifiedFactor: Sirve para calcular la caducidad de un archivo en la cache, que será el de la hora de la última modificación, multiplicado por este valor. Se encuentra disponible a través del módulo `mod_cache`.

CacheDefaultExpire: Número de horas por defecto a partir de las cuales un archivo caduca. Se aplica en aquellos casos en los que no se puede determinar la hora de creación del archivo. Todas las directivas de la caché, deben encontrarse fuera de cualquier sección. Esta directiva no se puede encontrar dentro de ninguna sección. Se encuentra disponible a través del módulo `mod_cache`.

Creación de directorios virtuales en Apache

Esta directiva sólo se encuentra dentro del archivo de configuración `httpd.conf`, y se trata aparte en este capítulo debido a su importancia.

Alias

Permite la definición de directorios virtuales, un directorio virtual es un directorio que se encuentra en un directorio distinto del que se mapea en la URL.

El directorio virtual no se tiene porqué encontrar dentro de árbol de directorios que se crea a partir de DocumentRoot, sino que se puede encontrar en cualquier otra ubicación, incluso se podría encontrar en otro servidor distinto.

Por ejemplo, cuando se escribe `www.miweb.com/manual/php` la carpeta `php` no se tiene que encontrar necesariamente dentro de la carpeta `manual`, que a su vez esta dentro de la carpeta raíz `miweb`, sino que puede estar en una ubicación distinta, y fuera del árbol de subdirectorios de la directiva `DocumentRoot`

Sintaxis:

```
Alias nombreFicticio ubicacionReal
```

Ejemplo:

```
Alias /manual/php "c:\php"
```

El directorio `php` no se encuentra dentro del directorio `manual` dentro de la carpeta `documentRoot`, sino en la carpeta `c:\php`. Por defecto vienen creados dos redirecciones con `Alias`.

Icons: para establecer la carpeta donde se encuentran los iconos que utilizará el Apache para mostrar el contenido de los directorios, y

Manual: que apunta a la carpeta donde está instalado el manual del Apache en caso de que se hubiera elegido la opción durante la instalación.

AliasMatch

La utilidad de esta directiva es idéntica a la de la directiva `Alias`, la única diferencia es que mientras `Alias` utiliza expresiones estándar regulares para especificar la URL que se va mapear.

Sintaxis:

AliasMatch Expresión regular ubicación

1.6.2. Lenguaje PHP.

PHP es el acrónimo de "PHP: Hypertext Preprocessor". Es un lenguaje multiplataforma de programación interpretado de alto nivel, pensado para desarrollos Web y embebido en páginas HTML. La facilidad de PHP es que tiene muchas funciones copiadas de C, Java y/o Perl. Lo que ve el cliente es el código ejecutado, es decir, el cliente solo ve lo que el código PHP ha producido, no la programación de éste, como pasaría con un Javascript.

PHP está llamado a sustituir a los CGI's (Common Gateway Interface) que fueron unas de las primeras maneras prácticas de crear contenido dinámico. En los CGI's el servidor Web pasa las solicitudes del cliente a un programa externo, como si fuese un script, y la salida es enviada al cliente como un archivo estático tradicional.

PHP 5 fue recientemente liberado con un soporte muy amplio para la programación orientada a objetos.

PHP no es un lenguaje que se interprete en el cliente o en el navegador, es un lenguaje que interpreta el servidor y manda la salida en el formato deseado (HMLT, XML...) al navegador. Esto quiere decir que no necesitamos que nuestro navegador tenga ninguna característica especial para poder ver páginas Web programadas en PHP.

PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

Existen tres campos en los que se usan scripts escritos en PHP.

- Scripts del lado del servidor. Este es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto funcione. El intérprete PHP (CGI ó módulo), un servidor Web y un navegador. Es necesario correr el servidor Web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectándose con el servidor Web.
- Scripts en la línea de comandos. Puede crear un script PHP y correrlo sin ningún servidor Web o navegador. Solamente necesita el intérprete PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en *nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesamiento de texto.
- Escribir aplicaciones de interfaz gráfica. Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas en programas clientes, se puede utilizar PHP-GTK para escribir dichos programas. También es posible escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores Web de hoy en día, incluyendo **Apache**, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, O'Reilly Website Pro Server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

De modo que, con PHP tenemos la libertad de elegir el sistema operativo y el servidor de su gusto. También tenemos la posibilidad de usar programación

procedimental o programación orientada a objetos. Aunque no todas las características estándares de la programación orientada a objetos están implementadas en la versión actual de PHP, muchas librerías y aplicaciones grandes (incluyendo la librería PEAR) están escritas íntegramente usando programación orientada a objetos. Con PHP no se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF y películas Flash.

También se pueden presentar otros resultados, como XHTML y archivos XML. PHP puede auto generar éstos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.

Quizá la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía Web para una base de datos es una tarea simple con PHP.

Las siguientes bases de datos están soportadas actualmente:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

También cuenta con una extensión DBX de abstracción de base de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. Adicionalmente, PHP soporta ODBC (el Estándar Abierto de Conexión con Bases de Datos), así que puede conectarse a cualquier base de datos que soporte tal estándar.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets puros. PHP soporta WDDX para el intercambio de datos entre lenguajes de programación en Web. Sobre interconexión, PHP puede utilizar objetos Java de forma transparente como objetos PHP y la extensión de CORBA puede ser utilizada para acceder a objetos remotos.

PHP tiene unas características muy útiles para el procesamiento de texto, desde expresiones regulares POSIX extendidas o tipo Perl hasta procesadores de documentos XML. Para procesar y acceder a documentos XML, soporta los estándares SAX y DOM. También puede utilizar la extensión XSLT para transformar documentos XML.

Para terminar, PHP cuenta con muchas otras extensiones muy interesantes, las funciones del motor de búsquedas mnoGoSearch, funciones para pasarelas de IRC, utilidades de compresión (gzip, bz2), conversión de calendarios, traducción, etc.

1.6.3. MySQL.

MySQL es un Sistema de Gestión de Base de Datos. Una Base de Datos es una colección estructurada de datos. Puede ser, desde una simple lista de artículos a las inmensas cantidades de información en una red corporativa.

MySQL es un gestor de bases de datos SQL (Structured Query Language). Es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías). Podemos agregar, acceder, y procesar datos grabados en una base de datos. Actualmente, el gestor de base de datos juega un rol central en la informática, como única utilidad, o como parte de otra aplicación. Es un Sistema de Gestión de Base de Datos Relacional. El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información debe estar

contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos. Esto añade velocidad y flexibilidad.

MySQL es un software de código abierto esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Podemos descargar MySQL desde Internet y usarlo sin pagar nada. De esta manera, cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades. MySQL usa el GPL (GNU Licencia Publica General) para definir que podemos y que no podemos hacer con el software en diferentes situaciones. Entre otras cuestiones esta licencia aclara que no cuesta dinero a menos que lo incluyamos en un software comercial y tengamos el código fuente.

MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace a MySQL altamente conveniente para acceder a bases de datos en Internet.

Información técnica

Principales Características:

- El principal objetivo de MySQL es velocidad y robustez.
- Escrito en C y C++, probado con GCC 2.7.2.1. Usa GNU autoconf para portabilidad.
- Clientes C, C++, JAVA, Perl, TCL.
- Multiproceso, es decir puede usar varias CPU si están disponibles.
- Puede trabajar en distintas plataformas y S.O. distintos.
- Sistema de contraseñas y privilegios muy flexibles y seguros.
- Todas las palabras de paso viajan encriptadas en la red.
- Registros de longitud fija y variable.

- 16 índices por tabla, cada índice puede estar compuesto de 1 a 15 columnas o partes de ellas con una longitud máxima de 127 bytes.
- Todas las columnas pueden tener valores por defecto.
- Utilidad (Isamchk) para chequeo, optimizar y reparar tablas.
- Todos los datos están grabados en formato ISO8859_1.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintas lenguas.
- Todos los comandos tienen -help o -? Para las ayudas.
- Diversos tipos de columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc.
- ODBC para Windows 95 (con fuentes), se puede utilizar ACCESS para conectar con el servidor.

Sistemas operativos soportados por MySQL:

- AIX 4.x con consulta nativa.
- BSDI 2.x con el paquete MIT-pthreads incluido.
- BSDI 3.0, 3.1 y 4.x con consulta nativa.
- DEC Unix 4.x con consulta nativa.
- FreeBSD 2.x con el paquete MIT-pthreads incluido.
- FreeBSD 3.x y 4.x con consulta nativa.
- HP-UX 10.20 con el paquete MIT-pthreads incluido.
- HP-UX 11.x con consulta nativa.
- Linux 2.0+ con LinuxThreads 0.7.1+ o glibc 2.0.7+.
- Mac OS X Server.
- NetBSD 1.3/1.4 Intel y NetBSD 1.3 Alpha (carece de la marca GNU).
- OpenBSD > 2.5 con consulta nativa. OpenBSD < 2.5 con el paquete MIT-pthreads incluido.
- OS/2 Warp 3, FixPack 29 y OS/2 Warp 4, FixPack 4.
- SGI Irix 6.x con consulta nativa.

- Solaris 2.5 y con consulta nativa sobre SPARC y x86.
- SunOS 4.x con el paquete MIT-pthreads incluido.
- SCO OpenServer con un puerto reciente al paquete de FSU Pthreads.
- SCO UnixWare 7.0.1.
- Tru64 Unix
- Win95, Win98, NT, y Win2000.

Una vez definida la arquitectura de un sistema Web, así como las herramientas que serán utilizadas en el presente proyecto, en el siguiente capítulo se aborda la etapa inicial del desarrollo de la aplicación, denominada como análisis.

Capítulo 2

Análisis y Diseño

En este apartado se presenta la problemática actual que se tiene durante el proceso de inscripciones a laboratorios de ingeniería de la Facultad en cuestión, así como también las áreas de mejora, el planteamiento de la propuesta que se sugiere para terminar con todos los aspectos dañinos que existen en dicho proceso y por último una descripción del diseño de la aplicación.

2.1. Problemática Actual.

En la actualidad, el proceso de inscripciones a laboratorios de ingeniería, se lleva a cabo bajo circunstancias poco ortodoxas que provocan que dicho proceso no sea el óptimo, a continuación se describe el proceso que actualmente se lleva a cabo.

2.1.1. Proceso de Inscripción.

El proceso comienza con el sorteo de inscripciones a laboratorios de ingeniería, mismo que es proporcionado por servicios escolares ya que es generado con relación a la base de datos que esta dependencia tiene a su cargo, dicho sorteo contiene el número de cuenta, nombre, fecha y hora de inscripción del alumno en cuestión. Estos datos en conjunto sirven para identificar el día y la hora en que cada uno de los alumnos puede llevar a cabo su inscripción a los diferentes grupos de laboratorios de ingeniería.

El listado correspondiente al sorteo, es publicado en medios impresos en las instalaciones de los laboratorios o en las áreas destinadas a la publicación de información de las diferentes carreras de ingeniería.

Una vez que el listado del sorteo de inscripciones ha sido publicado, cada alumno interesado en inscribirse en algún grupo de laboratorio, debe acudir a dichas áreas de publicación para consultar dicho listado y con ello, enterarse del día y la fecha en que podrá llevar a cabo su inscripción.

Llegada la fecha de inscripción, el alumno acude al edificio de los laboratorios, donde debe esperar la hora que tiene destinada para realizar el trámite, esto en compañía de muchas otras personas que realizan sus movimientos el mismo día e incluso a la misma hora que el alumno en cuestión.

La entrada al edificio permanece entreabierta y es controlada por personal de las mismas instalaciones, permitiendo el paso sólo a aquellas personas a las que les ha llegado su turno para realizar el trámite. Antes de que el interesado ingrese a las instalaciones para realizar su inscripción, éste debe tener debidamente llenado el formato correspondiente para poder llevarlo a cabo. Este formato contiene información sobre la materia inscrita en servicios escolares que requiere del curso de laboratorio, así como datos acerca del grupo al cual se quiere inscribir.

Una vez que el interesado ha ingresado a las instalaciones para llevar a cabo su inscripción, este debe dirigirse al área que le corresponde al laboratorio que se desea inscribir. Es en este lugar donde se lleva a cabo la inscripción, y es realizada por parte del personal de las instalaciones de laboratorios, los cuales verifican que el alumno tenga inscrita en su tira de materias proporcionada por servicios escolares, la materia a la cual corresponde el laboratorio, si los datos son correctos y el cupo del grupo al que se quiere inscribir, contiene lugares disponibles, la persona encargada realiza la inscripción, llevando el control sobre el número de personas inscritas en el grupo y sellando el comprobante de inscripción al interesado. Cabe señalar que dicho cupo es variable y es controlado por las personas encargadas de dirigir la inscripción a laboratorios.

Si no hay lugares disponibles en el grupo al cual se desea inscribir, entonces el alumno debe seleccionar algún otro grupo que le permita llevar a cabo el curso, si ya no hay ningún otro grupo que el usuario tenga interés, hay ocasiones en las que el interesado opta por hablar con el profesor encargado del grupo de teoría para que pueda cursar la materia sin hacer lo propio con el laboratorio, si no llegan

a ningún acuerdo, el alumno puede acudir a servicios escolares y dar de baja la materia en cuestión, aprovechando que el período de altas y bajas de materias, son las mismas fechas en las que se llevan a cabo las inscripciones a laboratorios.

En la figura 10 se muestra el diagrama UML del proceso de inscripciones a laboratorios de ingeniería describiendo gráficamente lo que se presentó en los párrafos anteriores.

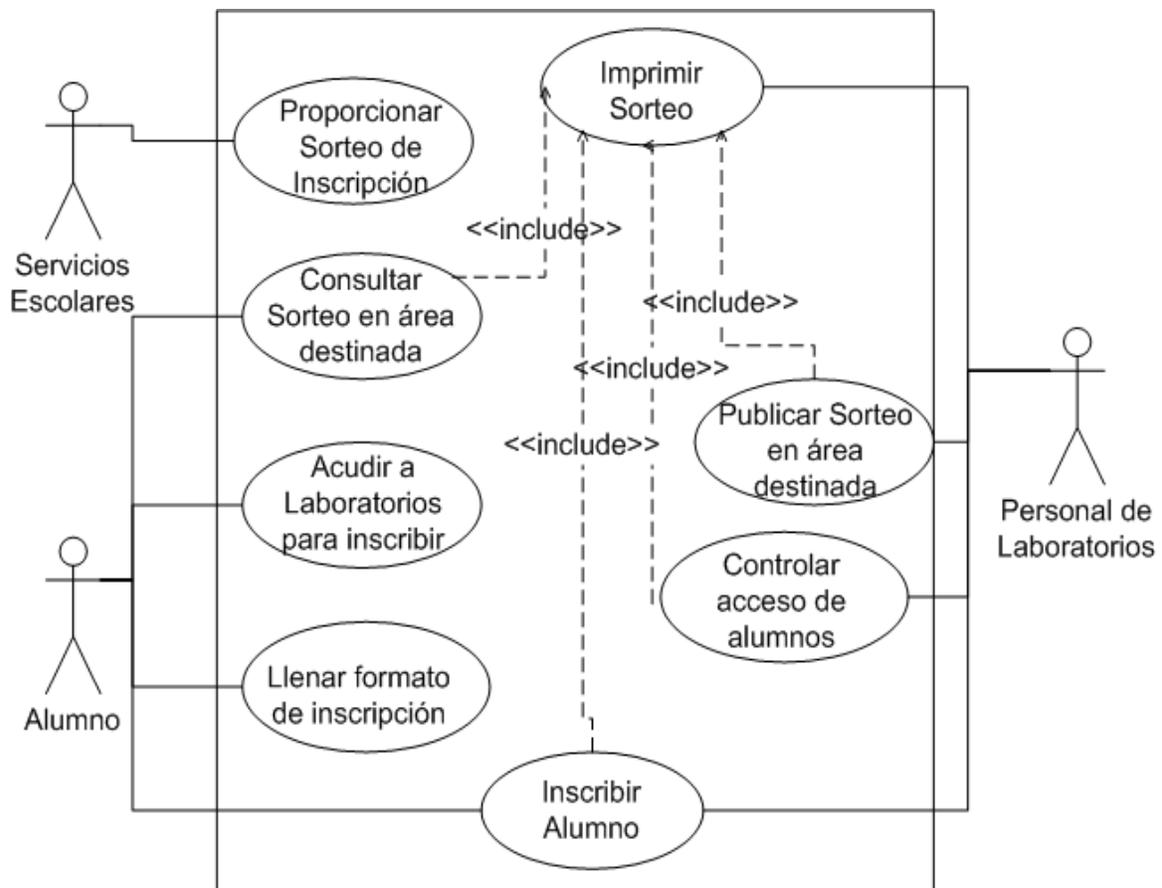


Figura 10. Diagrama UML del Proceso de Inscripciones.

En la figura 11, se muestra el diagrama UML del proceso de inscripción de un alumno a laboratorios de ingeniería.

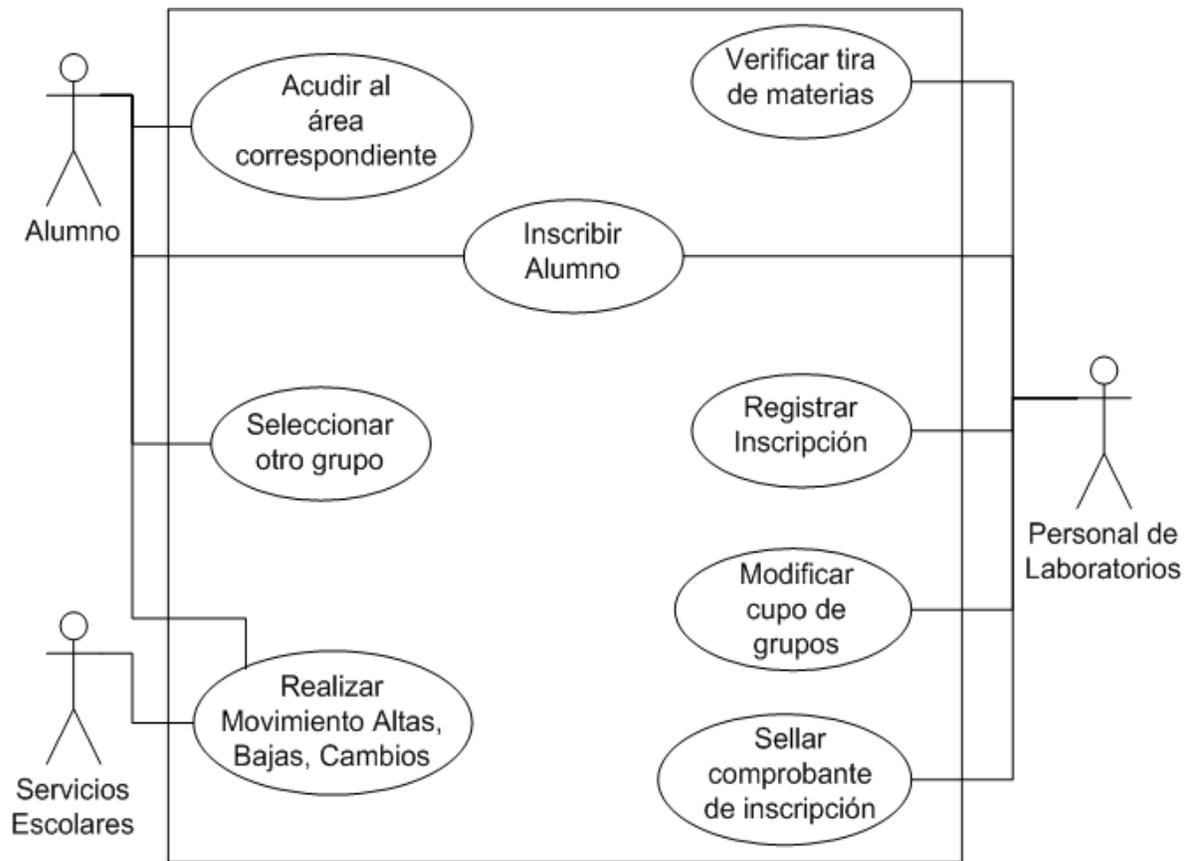


Figura 11. Inscripción de alumnos.

2.1.2. Problemas Durante el Proceso de Inscripción.

Como se puede ver en párrafos anteriores, el proceso de inscripciones a laboratorios de ingeniería es muy complicado en muchos aspectos debido a las condiciones en que es llevado a cabo.

Algunos de estos problemas son descritos a continuación.

- Pérdida de tiempo en la espera de la publicación del listado del sorteo.
- Incertidumbre sobre la fecha y la hora de inscripción del interesado, en caso de que el bloque del listado impreso donde se encuentra dicha información, no se encuentre en el área destinada para tal

efecto, ya sea porque no se publicó o incluso porque ha sido retirado del lugar.

- Pérdida de tiempo, incluso del turno correspondiente para realizar el trámite al desplazarse físicamente hasta las instalaciones de los laboratorios.
- Pérdida de tiempo al esperar el turno correspondiente afuera de las instalaciones de los laboratorios.
- Realización de actos que ponen en riesgo la integridad física de las personas que se encuentran esperando su turno para llevar a cabo su trámite, esto como resultado de empujones y agresiones que se presentan en estas condiciones.
- Incertidumbre sobre el status actual de los grupos de laboratorio en cuestión de lugares disponibles antes y durante la inscripción.
- Errores al realizar la inscripción por parte del personal encargado de llevar a cabo el trámite.

2.1.3. Áreas de Oportunidad.

La principal característica que se pretende proporcionar al proceso de inscripciones, es la de llevar a cabo el trámite desde cualquier punto geográfico desde el cual se tenga acceso mediante Internet. Aprovechando lo anterior, se puede dar al proceso de inscripciones la capacidad de realizar el trámite correspondiente vía remota.

La generación de listados para la administración del proceso es otra característica que puede ser agregada en el proceso de inscripciones. Una característica adicional en la parte post-inscripción es la publicación de documentos, así como la aplicación de calificaciones a cada uno de los alumnos de un grupo en particular que puedan ser consultados remotamente por los interesados.

2.1.4. Propuesta de la Solución.

Tomando en cuenta el proceso que se sigue actualmente para llevar a cabo las inscripciones a laboratorios de ingeniería, así como los problemas que esto presenta, se ha pensado en una solución capaz de terminar con estas propiedades dentro de dicho proceso, además de proporcionar nuevas características funcionales, todo ello con la oportunidad de crecimiento que todo sistema funcional requiere, aplicando en todo momento las reglas con que se opera actualmente.

La solución comprende a un sistema computacional con acceso remoto por los usuarios, por lo tanto, el sistema se convierte en una página Web la cual tiene como vía de acceso, el Internet desde cualquier equipo con conexión a esta red.

Considerando que una vez que el usuario en común entre al sistema, este será capaz de llevar el control sobre los contenidos que le serán mostrados, para lo cual es necesario tener en todo momento, información acerca del perfil del usuario en cuestión, para lo cual, el sistema tendrá soporte de sesiones y acceso a bases de datos.

Cualquier visitante a la página tendrá acceso a distintas opciones que no requieren que el usuario este firmado en el sistema, tal es el caso de las prácticas y apuntes de los distintos grupos de laboratorios. La contraparte de esta característica es que para tener acceso a las opciones que así lo requieran, el usuario debe firmarse dentro del sistema y después podrá hacer uso de dicha funcionalidad.

El sistema contará con la generación de reportes que ayudarán a la administración del proceso de inscripciones. Al generar cualquier reporte, el sistema solicita nuevamente la contraseña del usuario que requiere la información, en caso de que la contraseña sea incorrecta, el documento no será presentado al usuario. En caso

contrario y una vez que la contraseña es correcta, el documento podrá ser guardado en la máquina del usuario.

Cada vez que el documento que reside en la máquina del usuario, sea abierto, este solicitará la contraseña del administrador general del sistema o en su caso, del usuario que generó el documento, esto con la finalidad de dar o no ciertos privilegios al momento de abrir el archivo.

El sistema comprende un módulo administrador, al cual sólo tendrán acceso los usuarios con privilegios especiales para tal efecto.

Beneficios de la solución.

Las características que serán beneficiadas al momento de implementar el sistema de inscripciones a laboratorios de ingeniería son los siguientes:

- Consulta del sorteo de inscripción individual, sin tener que acudir físicamente a las instalaciones del campus, ya que podrá ser consultado desde la página Web del sistema una vez que el administrador ha realizado la carga del listado proporcionado por servicios escolares.
- Disponibilidad constante del sorteo de inscripción para ser consultado por el alumno las veces que le sean convenientes sin tener que estar firmado dentro del sistema.
- Consulta de la información en tiempo real, de este modo el alumno evita la pérdida de tiempo al desplazarse y formarse físicamente en la fila de inscripciones a laboratorios de ingeniería.
- Eliminar por completo las condiciones agresivas al estar formado en la fila de inscripciones a laboratorios de ingeniería.
- Eliminar la incertidumbre por completo con relación al número de plazas disponibles en cada uno de los grupos de laboratorio de ingeniería en los que el alumno puede estar interesado.

- Descentralizar la responsabilidad de las inscripciones de las personas encargadas de realizar el trámite delegando dicha responsabilidad a cada uno de los alumnos que llevan a cabo su inscripción de forma personal, en otras palabras, cada alumno es el responsable de realizar de forma correcta dicha inscripción.
- Aporta características adicionales que son de utilidad dentro de las actividades de los laboratorios. Una de estas características es la capacidad de descargar desde la página Web del sistema, cualquier práctica y/o apunte de los diferentes laboratorios impartidos en el campus. Otra característica adicional es el asentamiento de las calificaciones de cada uno de los integrantes de un grupo de laboratorio, mismo al que tendrán acceso los administradores del sistema y los maestros que tienen asignados grupos en el semestre en cuestión. La última característica de este tipo es la generación de diferentes reportes que ayudan a la operación de laboratorios de ingeniería.

Como se puede observar, todas estas características funcionales así como la razón misma de ser del sistema en cuestión, termina por completo con la problemática que se presenta actualmente en la operación de los laboratorios de ingeniería de la FES Aragón, motivo por el cual, se cumple con las expectativas de cualquier sistema de computación mediante el cual se busca la automatización de los procesos inmersos en las actividades de una organización.

Cabe señalar que se ha considerado en todo momento el facilitar el mantenimiento al sistema, diseñando la solución en capas y módulos específicos para tareas generales dentro de la operación, aplicando técnicas de programación orientada a objetos con lo que se obtiene la reutilización de código, herencia, etc., así como también las funciones genéricas de acceso y manipulación de la base de datos correspondiente.

2.2. Diseño de la Base de Datos.

Este apartado está dedicado al diseño de la base de datos del sistema de acuerdo a los requerimientos identificados durante el análisis. A continuación se muestran todos los puntos desarrollados en dicho diseño.

2.2.1. Diccionario de Datos.

Como resultado de los requerimientos identificados dentro de los laboratorios de ingeniería, así como la funcionalidad provista para alcanzar la operatividad necesaria al sistema, se presenta a continuación el diccionario de datos y posteriormente los diagramas que comprenden al diseño de la base de datos.

ERROR					
Descripción:	Esta entidad almacena los errores del sistema.				
Relaciones:	Ninguna.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
ERRORID	Identificador numérico.	INT			Y
CODIGO	Identificador numérico.	INT			Y
MENSAJE	Descripción del Error.	VARCHAR(200)			
SUGERENCIA	Sugerencia de Solución del Error.	VARCHAR(200)			
TIPO	Tipo de Error.	INT			

PRIVILEGIO					
Descripción:	Esta entidad almacena los tipos de usuarios del sistema.				
Relaciones:	Se relaciona con la entidad usuario.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
PRIVILEGIOID	Identificador numérico.	INT			Y
NOMBRE	Nombre del tipo de privilegio.	VARCHAR(20)	Administrador Maestro Alumno		
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

ESCUELA					
Descripción:	Esta entidad almacena los datos relacionados a la escuela.				
Relaciones:	Se relaciona con la entidad usuario.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
ESCUELAID	Identificador numérico.	INT			Y
NOMBRE	Nombre de la escuela.	VARCHAR(70)			
CLAVE	Identificador Alfanumérico.	CHAR(4)			Y
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

CARRERA					
Descripción:	Esta entidad almacena los datos relacionados a la carrera.				
Relaciones:	Se relaciona con la entidad usuario y con la entidad escuela.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
CARRERAID	Identificador numérico.	INT			Y
NOMBRE	Nombre de la carrera.	VARCHAR(50)			
CLAVE	Identificador Alfanumérico.	CHAR(4)			Y
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

USUARIO					
Descripción:	Esta entidad almacena los datos del usuario.				
Relaciones:	Se relaciona con la entidad privilegio, carrera y consigo misma.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
USUARIOID	Identificador numérico.	INT			Y
NOMBRE	Nombre del usuario.	VARCHAR(50)			
APEPATERNO	Apellido Paterno del usuario.	VARCHAR(50)			
APEMATERNO	Apellido Materno del usuario.	VARCHAR(50)			

USUARIO (Continuación)					
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
NCUENTA	Número de Cuenta en la UNAM.	VARCHAR(9)			Y
TELEFONO	Teléfono del usuario.	INT			
DIRECCION	Dirección del usuario.	VARCHAR(80)			
EMAIL	E-mail del usuario.	VARCHAR(50)			
MSN	Messenger del usuario.	VARCHAR(50)			
USUARIO	Nombre de usuario.	VARCHAR(50)			Y
CONTRASENA	Contraseña del usuario.	VARCHAR(32)			
USUARIONUEVO	Identifica si el usuario es nuevo en el sistema.	INT	0, 1	0	
CONFIRMACION	Identifica si el usuario ha confirmado su registro en el sistema.	CHAR(1)	N, Y	N	
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

CICLOESCOLAR					
Descripción:	Esta entidad almacena los datos del semestre.				
Relaciones:	Se relaciona con la entidad usuario.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
CICLOESCOLARID	Identificador numérico.	INT			Y
SEMESTRE	Nombre del Semestre.	VARCHAR(8)			Y
FECHAINICIO	Fecha en que comienza el Semestre.	DATE		0000-00-00	
FECHAFIN	Fecha en que termina el Semestre.	DATE		0000-00-00	
SEMANACIERRE	Fecha del primer y último día de semana de cierre.	VARCHAR(21)		0000-00-00 – 0000-00-00	

CICLOESCOLAR (Continuación)					
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
HORA0	Primer Hora de Inscripción.	VARCHAR(5)		00:00	
HORA1	Segunda Hora de Inscripción.	VARCHAR(5)		00:00	
HORA2	Tercer Hora de Inscripción.	VARCHAR(5)		00:00	
HORA3	Cuarta Hora de Inscripción.	VARCHAR(5)		00:00	
HORA4	Quinta Hora de Inscripción.	VARCHAR(5)		00:00	
HORA5	Sexta Hora de Inscripción.	VARCHAR(5)		00:00	
HORA6	Séptima Hora de Inscripción.	VARCHAR(5)		00:00	
HORA7	Octava Hora de Inscripción.	VARCHAR(5)		00:00	
HORA8	Novena Hora de Inscripción.	VARCHAR(5)		00:00	
HORA9	Décima Hora de Inscripción.	VARCHAR(5)		00:00	
DIAINSCRIPCION1	Primer día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION2	Segundo día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION3	Tercer día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION4	Cuarto día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION5	Quinto día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION6	Sexto día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION7	Séptimo día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION8	Octavo día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION9	Noveno día de Inscripción.	DATE		0000-00-00	
DIAINSCRIPCION10	Décimo día de Inscripción.	DATE		0000-00-00	
ALTASBAJAS	Último día de altas y bajas	DATE		0000-00-00	
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

SORTEO					
Descripción:	Esta entidad almacena los datos del listado del sorteo de inscripción.				
Relaciones:	Se relaciona con la entidad usuario y con la entidad cicloescolar.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
SORTEOID	Identificador numérico.	INT			Y
NCUENTA	Número de Cuenta del Alumno.	VARCHAR(9)			
NOMBRE	Nombre del Alumno.	VARCHAR(200)			
HORAINSCRIPCION	Hora de Inscripción.	VARCHAR(5)		00:00	
DIAINSCRIPCION	Día de Inscripción.	DATE		0000-00-00	
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

MAESTRO					
Descripción:	Esta entidad almacena los datos del maestro.				
Relaciones:	Se relaciona con la entidad usuario.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
MAESTROID	Identificador numérico.	INT			Y
NOMBRE	Nombre del Maestro.	VARCHAR(50)			
APEPATERNO	Apellido Paterno del Maestro.	VARCHAR(50)			
APEMATERNO	Apellido Materno del Maestro.	VARCHAR(50)			
CLAVE	Clave del Maestro.	CHAR(4)			Y
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

MATERIA					
Descripción:	Esta entidad almacena los datos de las materias.				
Relaciones:	Se relaciona con la entidad usuario.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
MATERIAID	Identificador numérico.	INT			Y
NOMBRE	Nombre de la Materia.	VARCHAR(70)			
CLAVE	Clave de la Materia.	CHAR(4)			Y
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

MATERIAINSCRITA					
Descripción:	Esta entidad almacena los datos de las materias inscritas del usuario en el semestre.				
Relaciones:	Se relaciona con la entidad usuario y con la entidad materia.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
MATERIAINSCRITAID	Identificador numérico.	INT			Y
NCUENTA	Número de Cuenta del Alumno.	VARCHAR(9)			
CLAVE	Clave de la Materia.	CHAR(4)			
GRUPO	Grupo de la Materia Inscrita.	INT			
SEMESTRE	Nombre del Semestre.	VARCHAR(8)			
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

LABORATORIO					
Descripción:	Esta entidad almacena los datos de los grupos de laboratorio.				
Relaciones:	Se relaciona con la entidad usuario, maestro, materia y cicloescolar.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
LABORATORIOID	Identificador numérico.	INT			Y
HORARIO	Horario en que se imparte el curso.	VARCHAR(10)			
UBICACION	Ubicación física donde se imparte el curso.	VARCHAR(50)			
DIASEMANA	Día de la semana en que se imparte el curso.	VARCHAR(50)	Lunes, Martes, Miércoles, Jueves, Viernes, Sábado, Domingo		
CUPO	Número de alumno del grupo.	INT			
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

INSCRIPCION					
Descripción:	Esta entidad almacena los datos de los alumnos inscritos en cada grupo.				
Relaciones:	Se relaciona con la entidad usuario, laboratorio, materia inscrita y cicloescolar.				
Atributos					
Nombre	Descripción	Tipo(Long)	Dominio	Defecto	Identificador
INSCRIPCIONID	Identificador numérico.	INT			Y
SEMESTRE	Nombre del semestre.	VARCHAR(8)			
CALIFICACION	Calificación obtenida por el alumno.	CHAR(2)			
FECHACREACION	Fecha de creación.	DATE		0000-00-00	
FECHAMODIFICACION	Fecha de Modificación.	DATE		0000-00-00	

2.2.2. Diagrama E/R (Modelo Conceptual).

El diagrama E/R de la figura 12, tiene todas las entidades que intervienen en el proceso de inscripciones a laboratorios y sus relaciones.

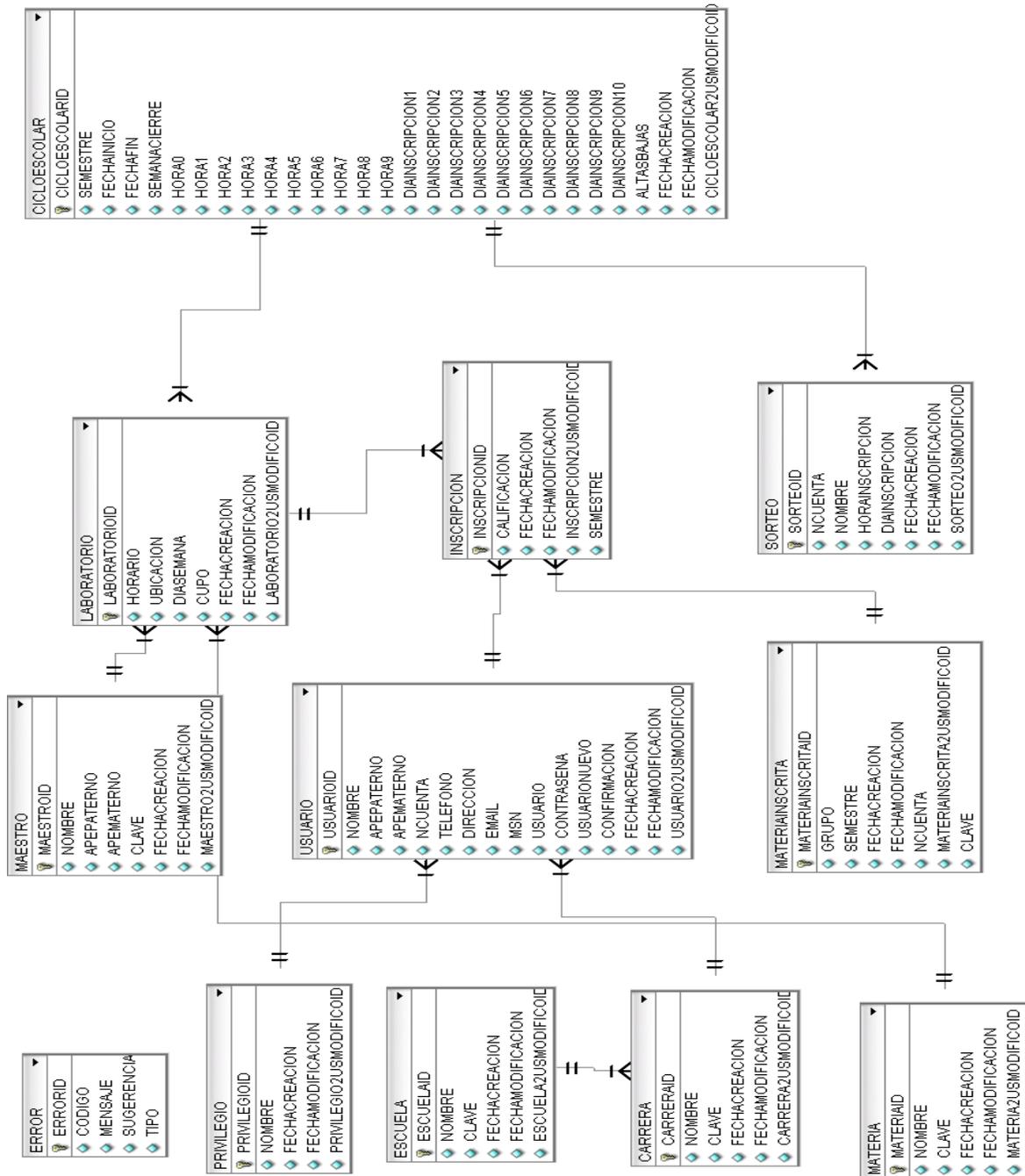


Figura 12. Diseño DB Modelo Conceptual.

2.3. Identificación de Módulos.

Se han identificado un conjunto de módulos para terminar por completo con la problemática identificada en el proceso de inscripciones a laboratorios de ingeniería. Dichos módulos serán tratados con más detalles en los siguientes puntos de este escrito. En la figura 15 se muestra el diagrama conceptual del sistema, en él se describen cada uno de los módulos que intervienen en el proceso y sus componentes correspondientes.

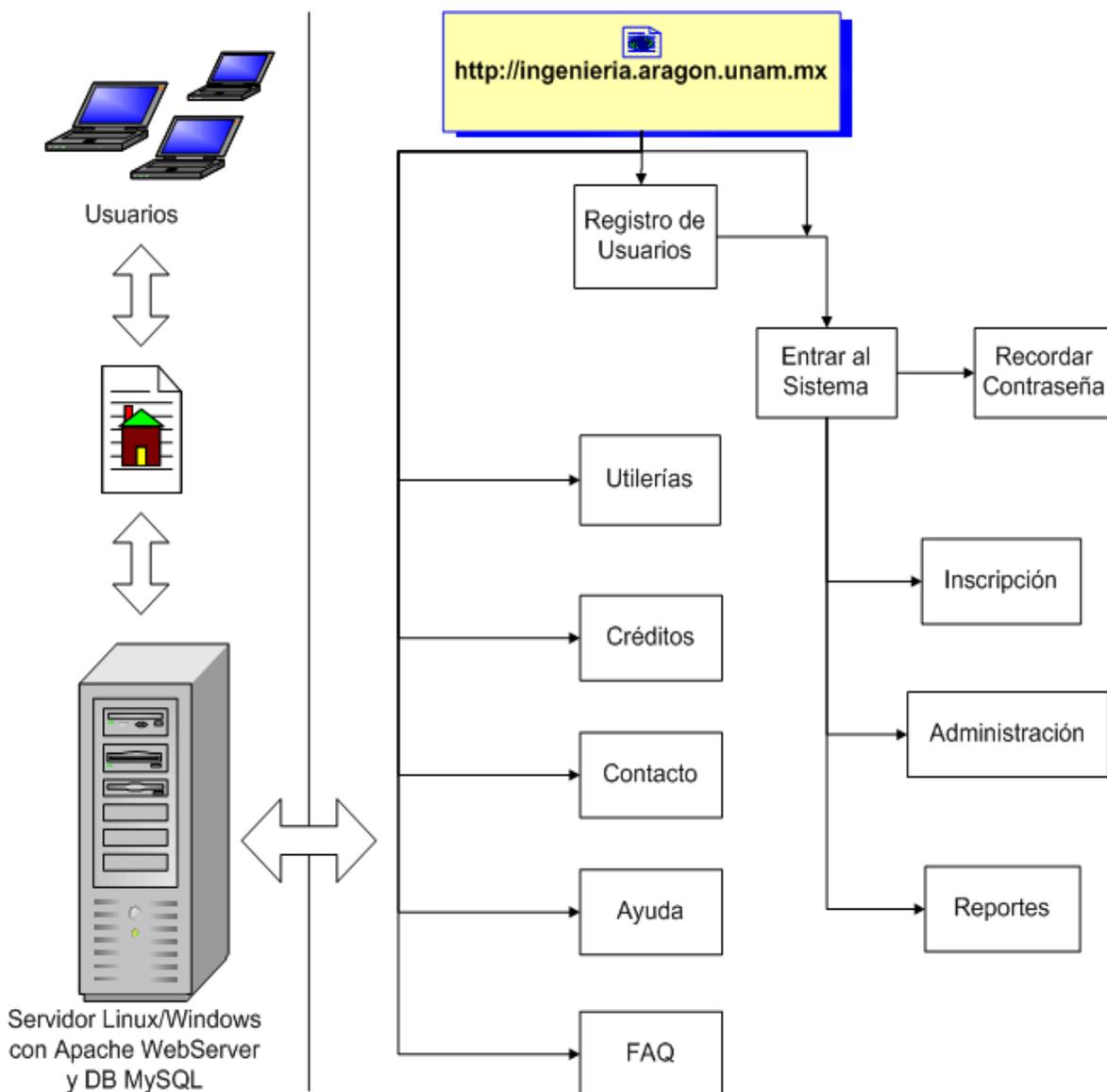


Figura 15. Diagrama Conceptual del Sistema.

- **Módulo de Acceso e Inscripción al Sistema.-** En este módulo se encuentra la funcionalidad necesaria para que un usuario se pueda inscribir, entrar y salir del sistema de Laboratorios. Si el usuario en cuestión ha olvidado su contraseña, existe una liga dentro del formulario de acceso al sistema, el cual permite al sistema enviar la contraseña al usuario que así lo requiera vía electrónica. Todos los usuarios tendrán acceso a este módulo dentro del sistema.
- **Módulo de Utilerías.-** En este punto se encuentran herramientas útiles para usuarios del sistema, como son las prácticas y apuntes referentes a los diferentes laboratorios de ingeniería, así como el sorteo de inscripción de cada alumno. Todos los usuarios tendrán acceso a este módulo dentro del sistema.
- **Módulo de Inscripción a Laboratorios y Llenado y/o Consulta de calificaciones.-** En este módulo esta la funcionalidad para que un usuario se inscriba a los laboratorios de ingeniería. Una vez que el ciclo escolar de los laboratorios de ingeniería ha terminado, el profesor tendrá que vaciar en el sistema las calificaciones obtenidas por cada persona del grupo, por otro lado, el alumno podrá consultar las calificaciones obtenidas en dicho ciclo escolar. Todos los usuarios con privilegios de “Alumno” tendrán acceso a este módulo para inscribirse a los diferentes grupos de laboratorios de ingeniería, así como también para consultar la calificación correspondiente de cada grupo inscrito, pero no tendrá acceso a la parte correspondiente al llenado de calificaciones, a esta parte tendrán acceso solo profesores y administradores del sistema.
- **Módulo de Administración del Sistema.-** Desde este punto se lleva a cabo la administración del sistema, por esta razón, solo tienen acceso los administradores y/o encargados del sistema.
- **Módulo de Reportes.-** Este módulo es capaz de generar los reportes que se requieran para la toma de decisiones y/o administración del

sistema. Los usuarios con privilegios de “Maestro” ó “Administrador” tendrán acceso a este módulo.

2.4. Descripción de Módulos dentro del sistema.

Se ha diseñado un menú horizontal para almacenar de forma gráfica cada uno de los módulos del sistema, se tiene una opción por cada módulo y estos a su vez han generado más entradas para el funcionamiento de cada una de las acciones relacionadas a cada módulo. Los nombres de las entradas del menú principal son:

- **Sistema.**
- **Utilerías.**
- **Inscripciones y Calificaciones.**
- **Administración.**
- **Reportes.**

En los siguientes apartados se muestra la estructura que tiene cada una de las entradas del menú principal.

2.4.1. Sistema.

Este es el primer menú del sistema al cual tienen acceso todos los usuarios. Dentro de él se encuentran las siguientes opciones.

- **Inicio.-** Lleva al usuario a la página principal del sistema.
- **Alta en el sistema.-** Comienza con el proceso para inscribirse al sistema de laboratorios de ingeniería.
- **Acceso al sistema.-** Lleva al usuario al formulario para entrar al sistema de laboratorios de ingeniería.
- **Salir.-** Termina con la sesión del usuario dentro del sistema.

En la figura 16 se muestra el diagrama de flujo global del sistema.

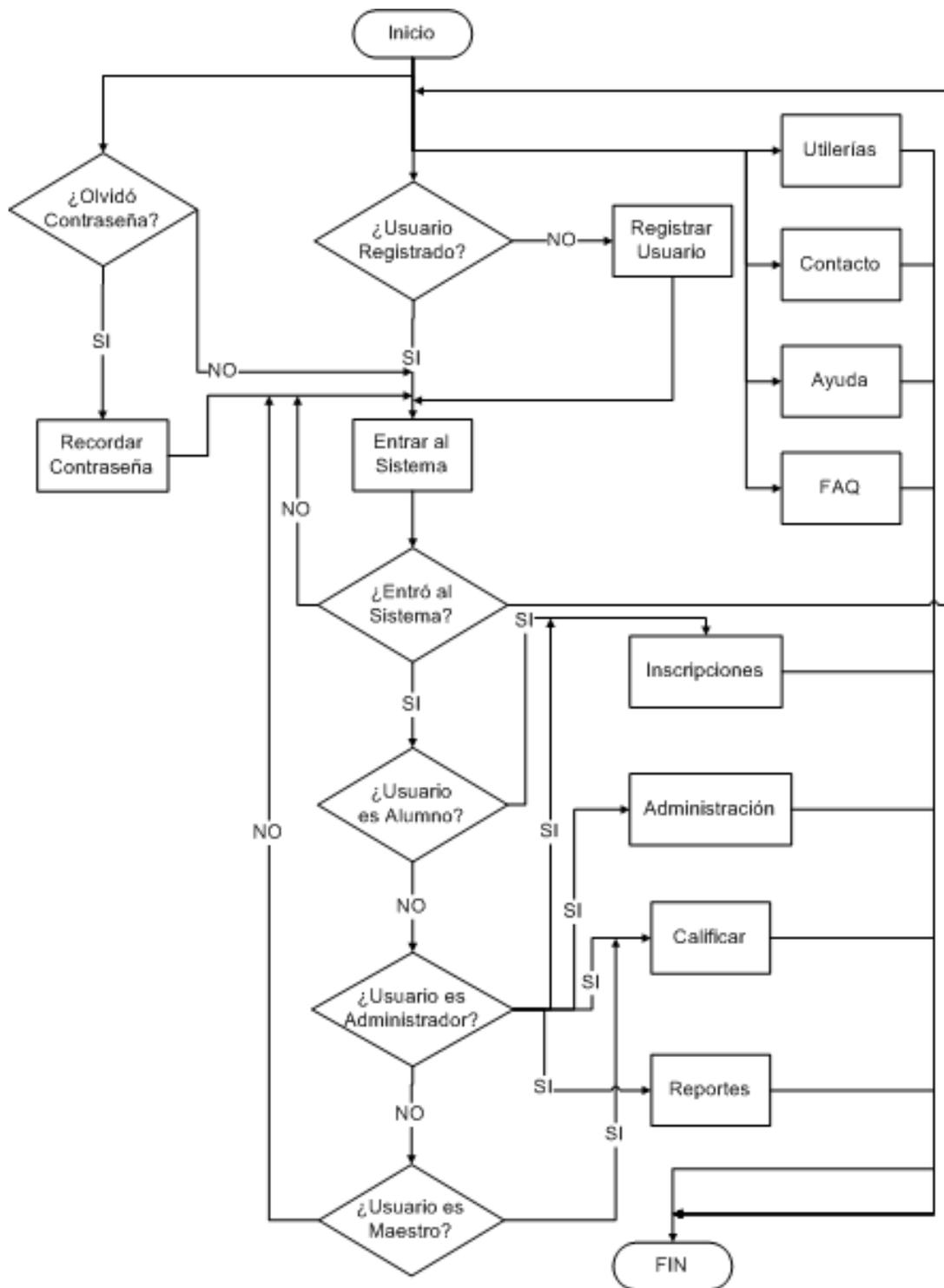


Figura 16. Diagrama de Flujo Global del Sistema.

En la figura 17 se muestra el diagrama de flujo correspondiente al módulo de registro de usuarios.

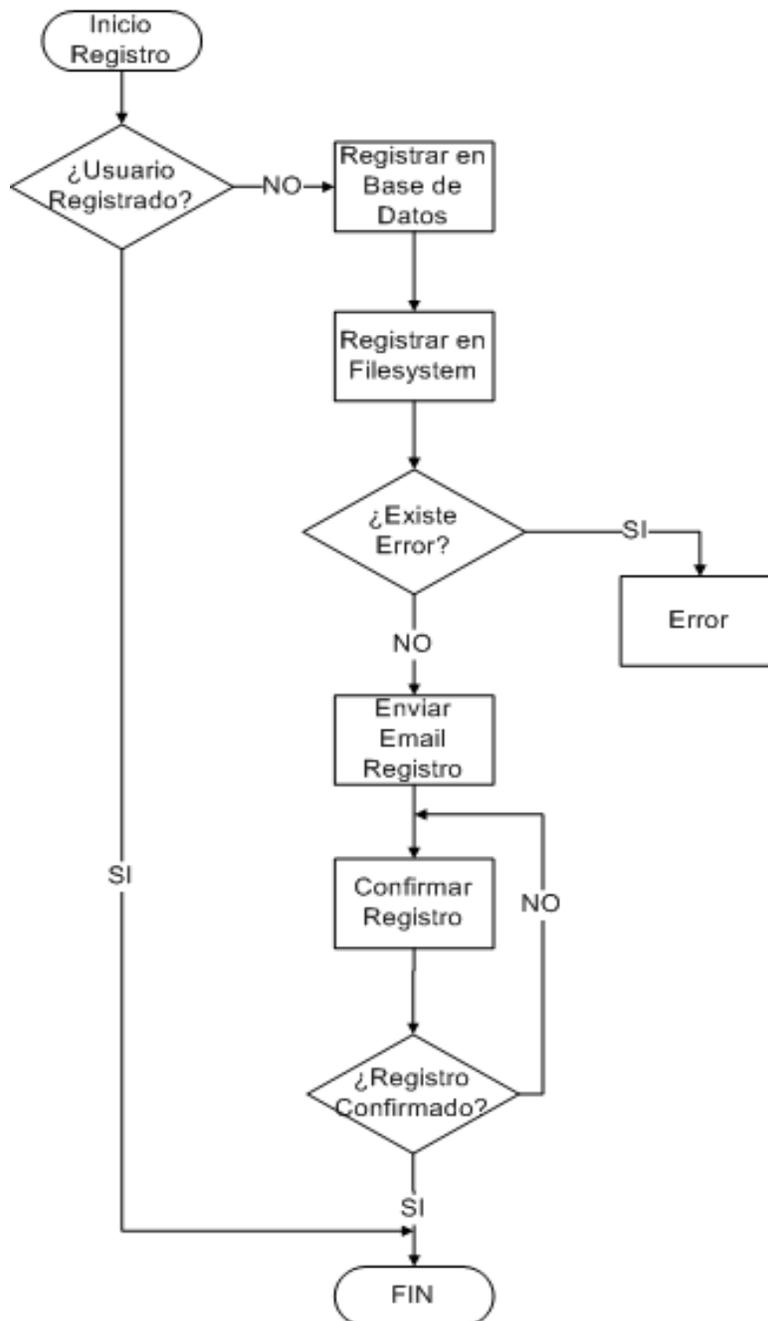


Figura 17. Diagrama de Flujo del Módulo de Registro de Usuarios.

En la figura 18 se muestra el diagrama de flujo correspondiente al módulo de acceso al sistema.

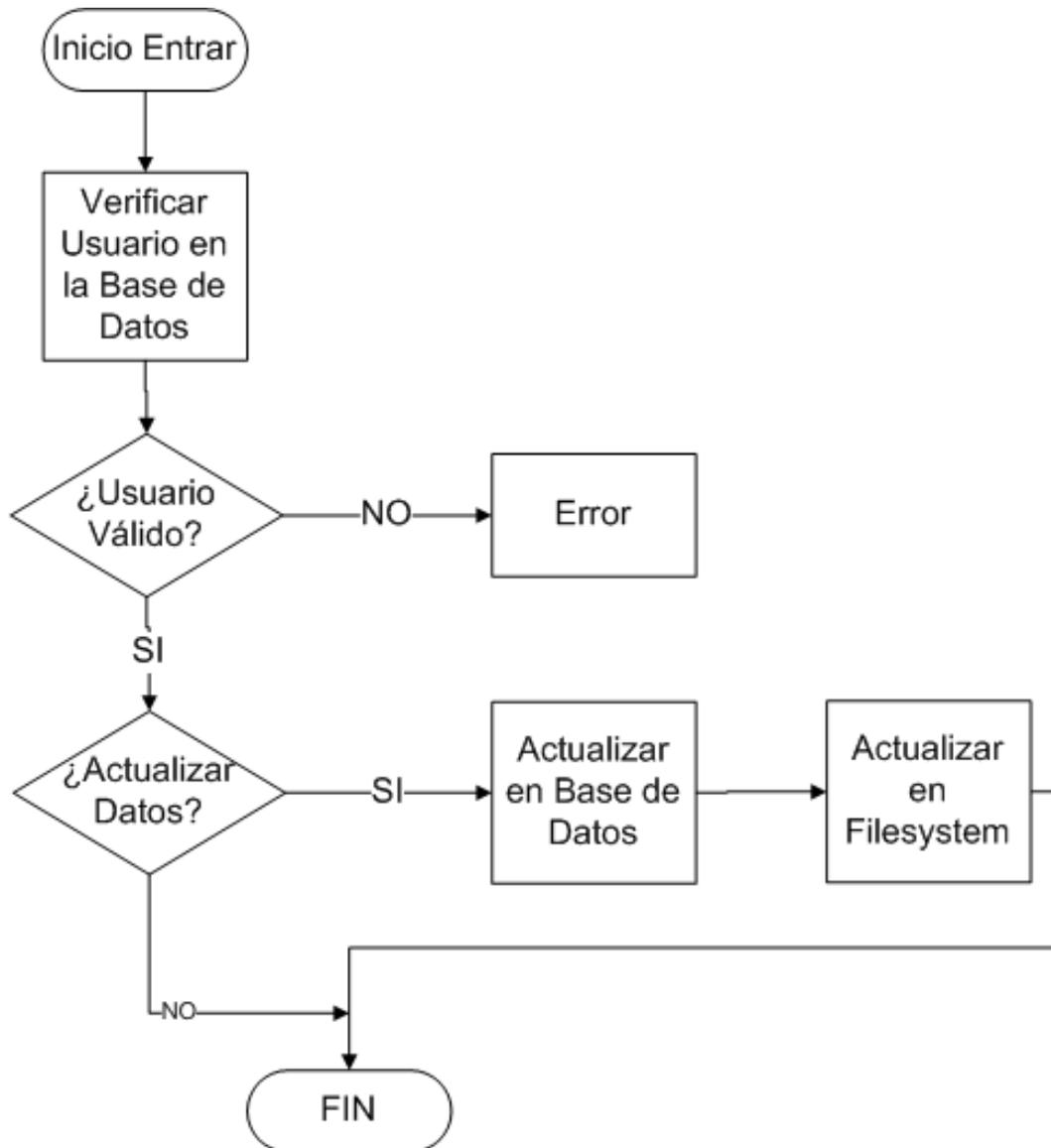


Figura 18. Diagrama de Flujo del Módulo de Acceso al Sistema.

En la figura 19 se muestra finalmente el diagrama de flujo correspondiente al módulo dentro del sistema para que el usuario pueda recuperar la contraseña en caso de que la haya olvidado.

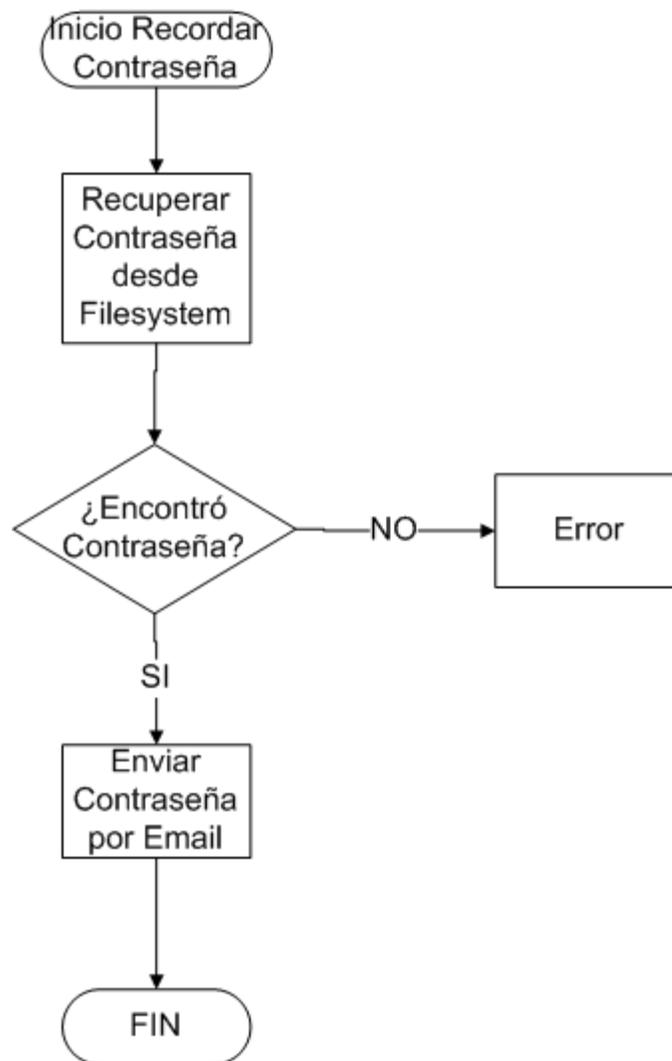


Figura 19. Diagrama de Flujo del Módulo de Recordar Contraseña.

2.4.2. Utilerías.

Es el segundo menú del sistema en el que se tienen opciones que le son de utilidad a cualquier visitante de la página sin estar autenticado dentro de la aplicación. Se tienen las siguientes opciones. Cabe señalar que existe una validación interna para evitar que el usuario intente cargar en el sistema un listado con un formato y orden de la información incorrectos.

- **Prácticas.-** Bajo esta opción se pueden obtener los documentos referentes a las prácticas de cada uno de los laboratorios de ingeniería.

Estos documentos pueden estar en formato Word y/o PDF y dependerá del formato en que el administrador haya publicado dicha información.

- **Apuntes.-** En esta opción se encuentran apuntes de diferentes laboratorios de ingeniería que ayudan al alumno para entender mejor lo visto en clase.
- **Consultar Sorteo.-** En esta opción se puede consultar el día y la hora en que el alumno podrá realizar su inscripción a laboratorios.

En la figura 20 se muestra el diagrama de flujo correspondiente al módulo de utilerías.

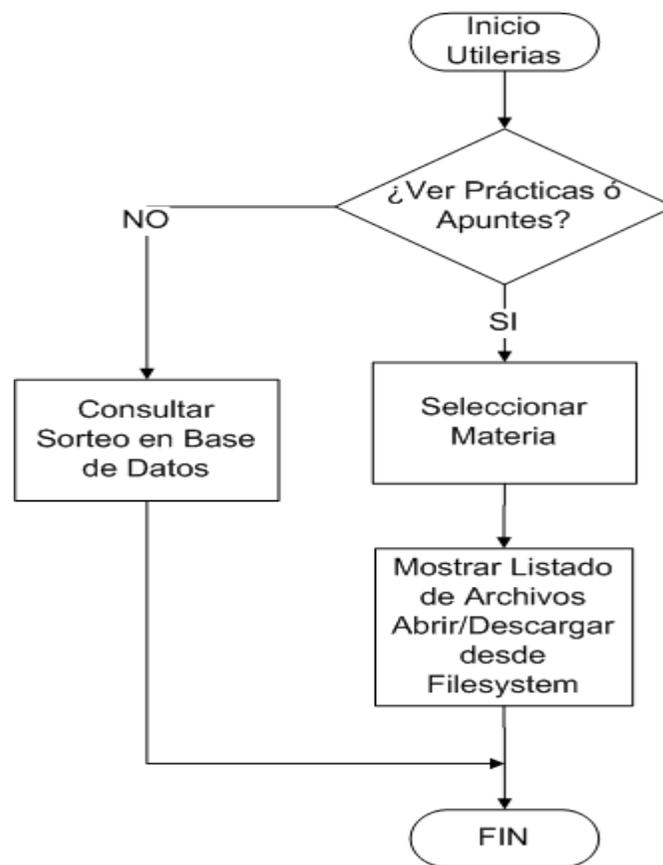


Figura 20. Diagrama de Flujo del Módulo Utilerías.

2.4.3. Inscripciones y Calificaciones.

Este es el tercer menú del sistema y para tener acceso al mismo, se debe estar autenticado en la aplicación, debido a que depende de los privilegios con que cuenta cada usuario para tomar un comportamiento específico a cada rol. Por ejemplo, si el usuario tiene privilegios de “Alumno”, este sólo podrá inscribirse a los grupos de las materias de teoría en que se haya inscrito en servicios escolares y obtener el comprobante de dicha inscripción y en su caso las calificaciones obtenidas en cada uno de los laboratorios cursado. Por otra parte si el usuario tiene privilegios de “Administrador”, este podrá asentar la calificación a cada uno de los alumnos de cualquier grupo de laboratorio o en su caso, inscribir por medio de su cuenta, a varios usuarios de tipo “Alumno”. El usuario con privilegios de “Maestro” solo podrá asentar las calificaciones a cada uno de los alumnos de los grupos de laboratorio que tiene asignados durante el semestre.

- **Inscripciones.-** Comienza el proceso de inscripción a los diferentes grupos de laboratorios de ingeniería. Cuando el alumno entre a esta opción para inscribirse a grupos de laboratorios de ingeniería, sólo aparecerán los grupos de laboratorios que correspondan a las materias de teoría inscritas por parte del alumno, de esta forma, no podrá inscribirse a grupos de laboratorios de materias que no tenga en su tira de materias.
- **Calificaciones.-** Este apartado sirve para ver o grabar las calificaciones obtenidas por cada uno de los alumnos que se inscribieron a un grupo de laboratorios. Si el usuario en cuestión tiene privilegios de “Maestro”, entonces podrá grabar y/o ver las calificaciones de los alumnos que estén inscritos a los grupos de laboratorio que tiene a su cargo. Si el usuario tiene privilegios de “Alumno”, solo podrá ver las calificaciones de los laboratorios que esta inscrito.

En las figuras 21 y 22 se muestran los diagramas de flujo correspondientes a los módulos de inscripciones y calificaciones respectivamente.

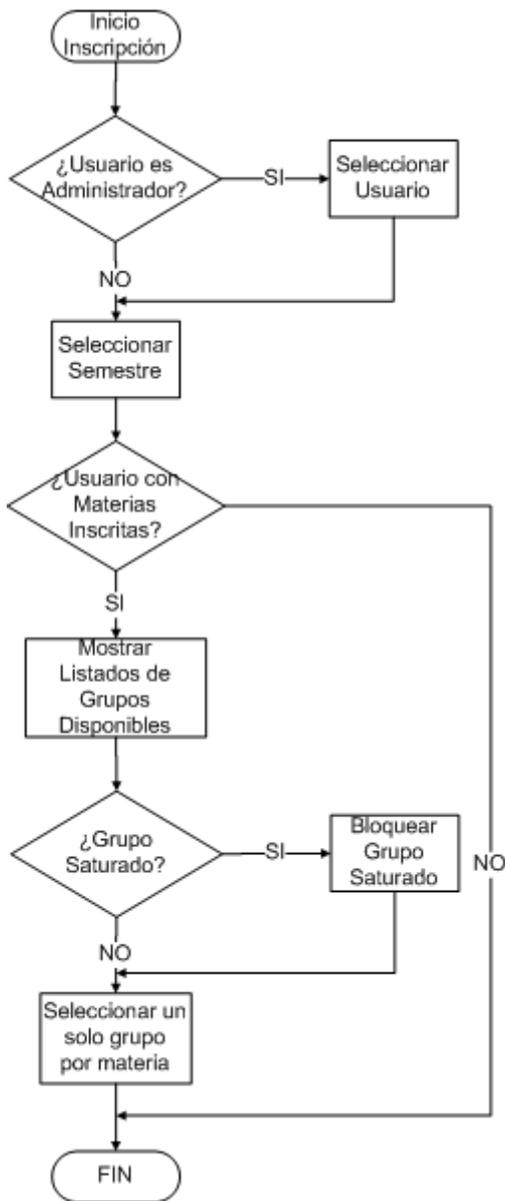


Figura 21. Diagrama de Flujo del Módulo de Inscripciones.

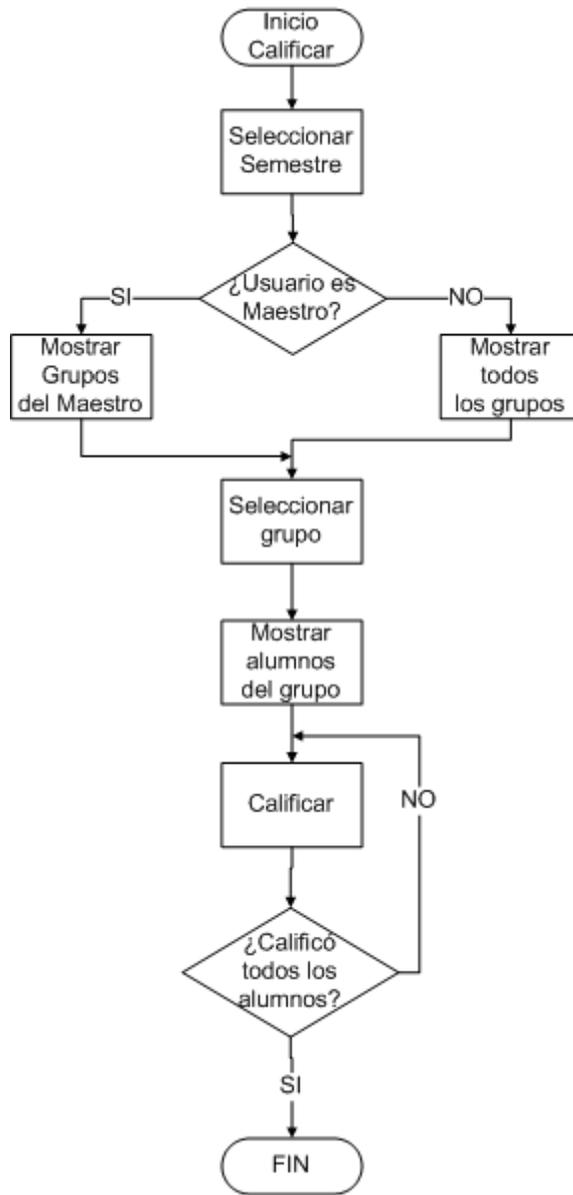


Figura 22. Diagrama de Flujo del Módulo de Calificaciones.

2.4.4. Administración.

Este menú sólo aparece a los usuarios con privilegios de “Administrador” para tener el control de los cambios que se hacen en el sistema. En la figura 23 se muestra el diagrama de flujo correspondiente al módulo de administración del sistema.

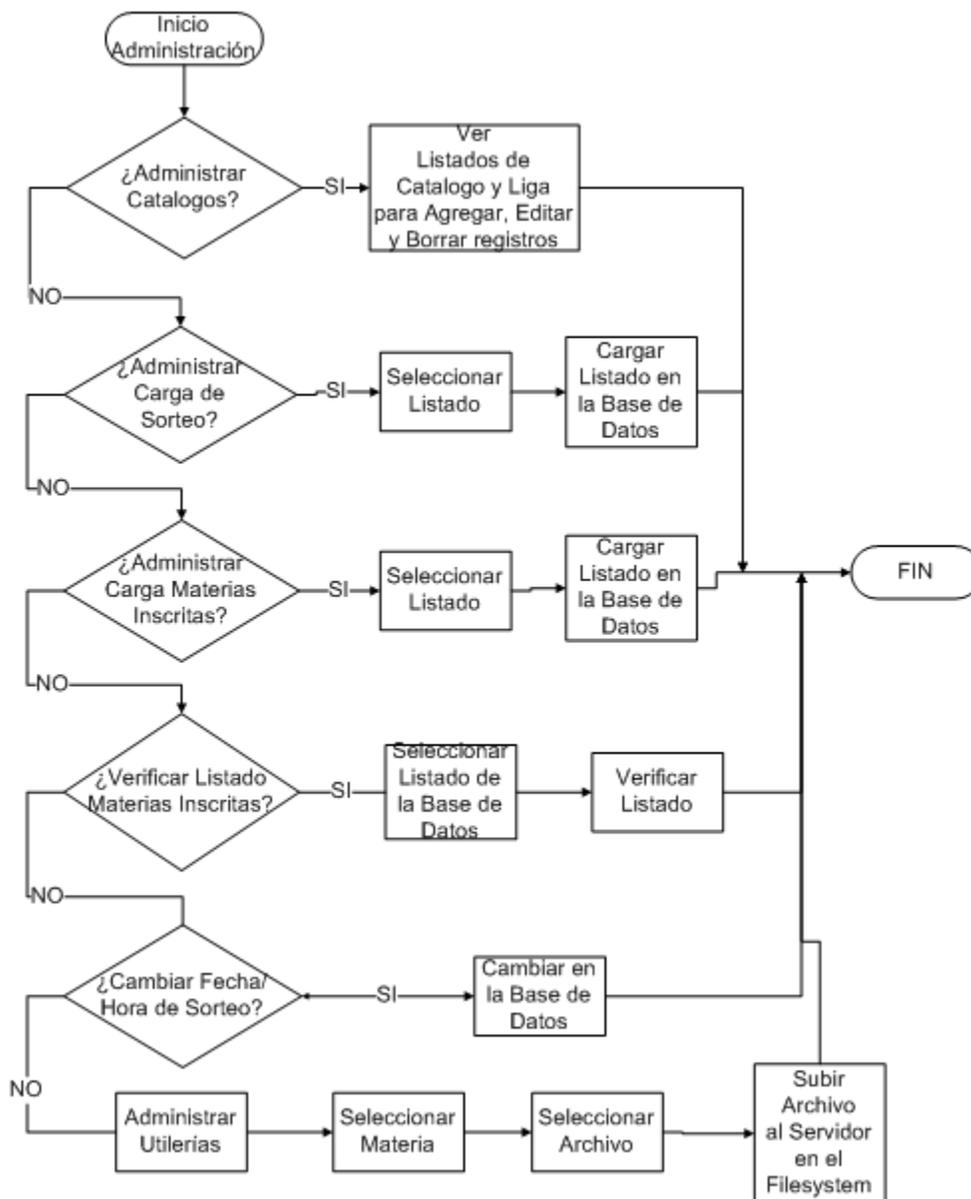


Figura 23. Diagrama de Flujo del Módulo de Administración del Sistema.

- **Catálogos.-** En esta opción se encuentran varios submenús para administrar los catálogos que intervienen para la inscripción a laboratorios de ingeniería de la siguiente forma. Se muestra un listado de los registros contenidos en un determinado catálogo y se muestran también una serie de ligas para agregar, editar o borrar cada uno de los registros del catálogo según se requiera en la operación. Dichos catálogos son los siguientes.
 - **Ciclo Escolar.-** Permite la administración del catálogo de ciclo escolar. Este catálogo es muy importante, ya que sobre el se basan muchas de las funcionalidades del sistema.
 - **Usuarios.-** Permite la administración del catálogo de usuarios.
 - **Maestros.-** Permite la administración del catálogo de Maestros.
 - **Materias.-** Permite la administración del catálogo de Materias.
 - **Grupos.-** Permite la administración del catálogo de grupos de laboratorios.
- **Carga del Sorteo de Inscripción.-** Carga toda la información referente al proceso de inscripción, como fecha y día de inscripción, así como los datos del alumno (nombre, número de cuenta, materias de teoría inscritas, etc.).
- **Carga de Materias de Teoría Inscritas.-** Realiza la carga de un listado proporcionado por servicios escolares donde informa las materias de teoría que tiene inscritas cada uno de los usuarios del sistema y que requieren de laboratorio, esto con la finalidad de mostrar al “Alumno” los grupos de laboratorio de las materias que lo requieren y que aparecen en su tira de materias correspondiente al semestre en cuestión.
- **Verificación del listado de Materias de Teoría Inscritas.-** Debido a que las fechas de inscripción a laboratorios y de los movimientos de altas, bajas y cambios de materias son las mismas, puede darse el caso en que un alumno se inscriba a un determinado grupo de laboratorio y posteriormente realice un movimiento de baja de la materia de teoría

que requería del laboratorio en cuestión. En este caso, este alumno estaría ocupando un lugar dentro de los laboratorios de forma innecesaria, motivo por el cual, una vez que han pasado las fechas de altas y bajas, servicios escolares deberá proporcionar al administrador del sistema, un listado de todas las materias de teoría inscritas por cada alumno, en este momento, si un alumno ya no cuenta con la materia referente al laboratorio en que esta inscrito, entonces el sistema dará de baja la inscripción en laboratorios de dicho alumno. En caso contrario, si un alumno, dio de alta una nueva materia que requiere laboratorio, el sistema tendrá disponible el mecanismo de inscripción a laboratorios después de las fechas destinadas para tal caso, únicamente para casos especiales y bajo el criterio del administrador del sistema.

- **Cambiar Fecha/Hora de Inscripción del Alumno.-** Esta opción ha sido diseñada para que el administrador del sistema y bajo su propio criterio, pueda cambiar la fecha y/o hora de inscripción a laboratorios de un alumno en específico, esto con la finalidad de que dicho alumno pueda ser inscrito después del tiempo destinado para hacerlo, por ejemplo, si el alumno se inscribe a una materia que requiere laboratorio después de las fechas de altas y bajas o incluso si ha pasado la fecha de inscripción sin que el interesado haya realizado el trámite.
- **Copiar grupos.-** Al comienzo de cada semestre, el administrador del sistema debe armar uno a uno, todos los grupos de laboratorios que se impartirán durante el curso, tomando en cuenta la plantilla de maestros y horarios disponibles de los mismos. Normalmente dichos grupos no tienen afectación alguna en los diferentes semestres, motivo por el cual, esta opción ha sido diseñada. Básicamente sirve para copiar todos los grupos que permanecerán constantes de un semestre a otro, quedando disponibles para las inscripciones del nuevo curso en cuestión.

2.4.5. Reportes.

Este es el último menú del sistema al cual tienen acceso los usuarios con privilegios de “Administrador” y “Maestro”. Su función principal es la de obtener información relevante para la administración y/o uso del sistema. En la figura 24 se muestra el diagrama de flujo correspondiente al módulo generador de reportes del sistema.

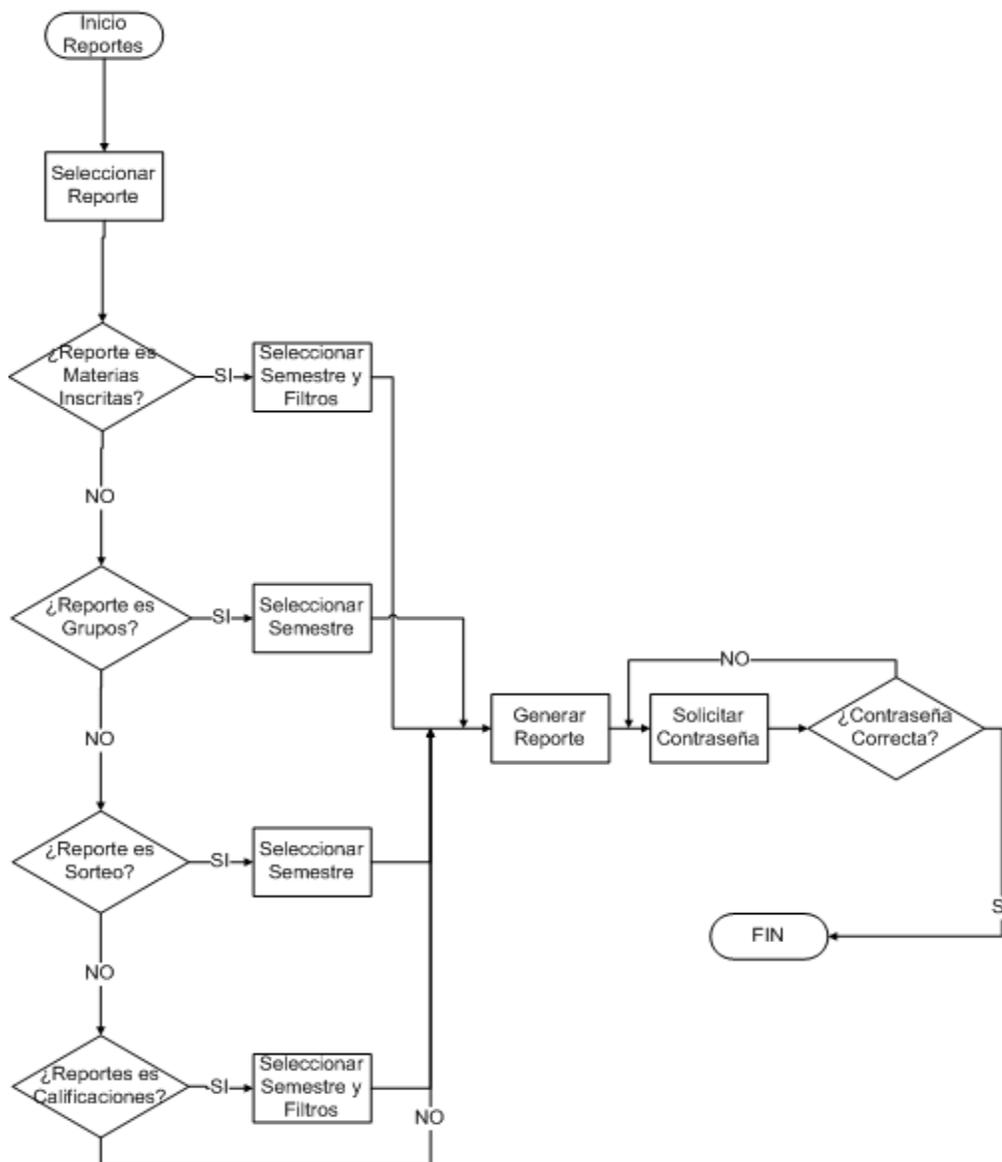


Figura 24. Diagrama de Flujo del Módulo de Reportes del Sistema.

- **Usuarios.-** Genera un listado con toda la información referente a los usuarios del sistema, nombre, dirección, e-mail, etc. En un documento PDF para que no se hagan alteraciones al documento.
- **Materias.-** Genera un listado con toda la información de las materias de teoría que tienen su correspondiente grupo de laboratorio.
- **Materias de Teoría Inscritas.-** Genera un listado con toda la información de cada una de las materias de teoría que requiere de laboratorio en las que esta inscrito el usuario con privilegios de “Alumno”.
- **Maestros.-** Genera un listado con toda la información de los maestros que tienen asignados grupos de laboratorios.
- **Grupos.-** Genera un listado con toda la información acerca de los grupos de laboratorios.
- **Inscripciones y Calificaciones.-** Genera un reporte de los grupos con las personas inscritas en cada uno de ellos. Si el período escolar ha terminado, este reporte tendrá las calificaciones de cada uno de los alumnos de cada grupo, una vez que el maestro o el administrador del sistema ha cargado dichas calificaciones al sistema mismo.
- **Sorteo.-** Genera un listado con toda la información acerca del sorteo de inscripciones a laboratorios de Ingeniería.

2.4.6. Opciones Generales.

Se desarrollaron 4 módulos adicionales para los cuales no es necesario que el usuario este firmado dentro del sistema y por lo tanto pueden ser consultados en cualquier momento y por cualquier visitante del sitio de la aplicación.

Dichos módulos son los siguientes:

- **Contacto.-** Mediante esta opción es posible enviar mensajes de correo al administrador del sistema.
- **Créditos.-** Con esta opción, se abre una nueva ventana que contiene la información de los créditos del sistema.

- **Ayuda.-** En este punto se muestra la documentación del sistema disponible en tres formatos, DOC. PDF y HTML.
- **FAQ.-** Mediante esta opción se muestran las preguntas frecuentes de los usuarios del sistema.

En las figuras 25, 26, 27 y 28 se muestran los correspondientes diagramas de flujo de los módulos generales mencionados anteriormente.

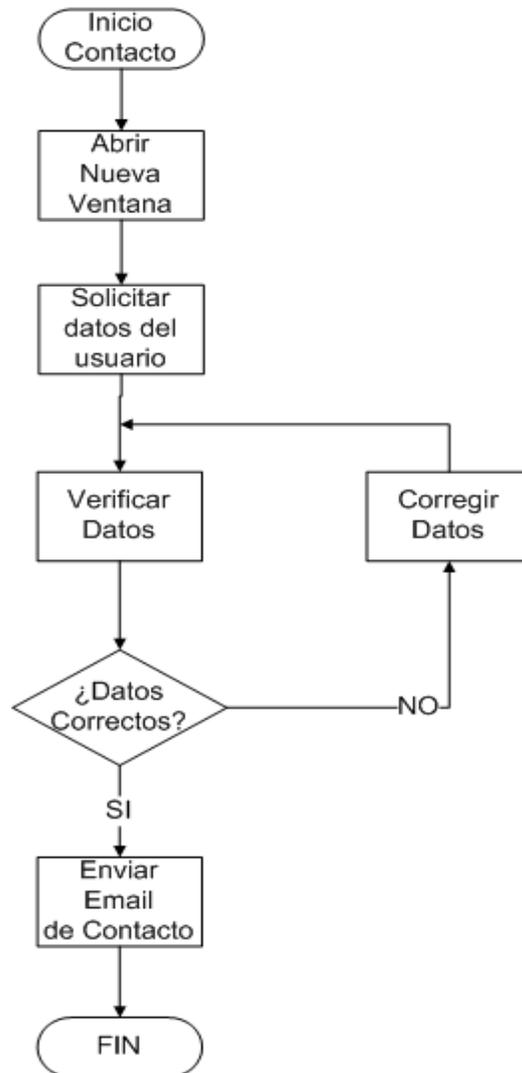


Figura 25. Diagrama de Flujo del Módulo de Contacto.

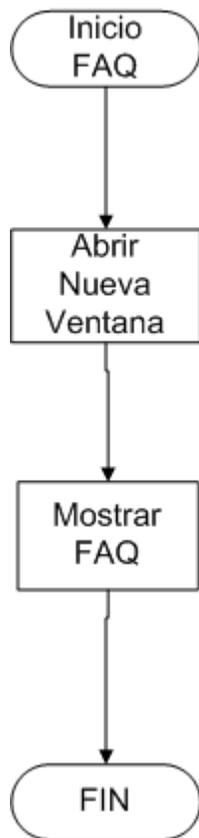


Figura 26. Diagrama de Flujo del Módulo de FAQ

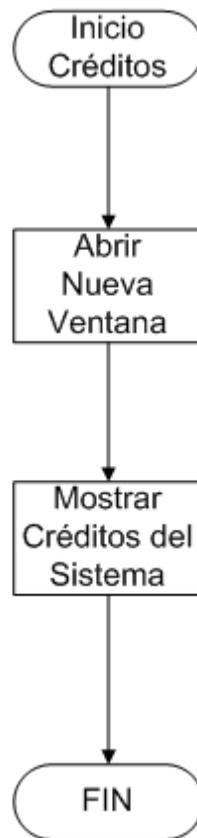


Figura 27. Diagrama de Flujo del Módulo de Créditos.



Figura 28. Diagrama de Flujo del Módulo de Ayuda

2.4.7. Reglas de negocio.

Como en cualquier sistema desarrollado para automatizar las tareas que se llevan a cabo de forma manual en ciertos procesos, es importante considerar las reglas con las que cuenta la entidad que aplica dicho proceso, en este caso el sistema de laboratorios de Ingeniería. A continuación se presentan algunas de las reglas de negocio con que se cuenta actualmente y algunas adicionales que se proponen al utilizar el sistema.

- Un alumno puede inscribirse a un grupo de laboratorios una vez que este ha sido registrado en el sistema por el mismo o que un usuario administrador haya realizado esta operación.

- Un alumno sólo puede inscribirse a un grupo de laboratorio que corresponda con la materia de teoría inscrita en su tira de materias una vez que se haya cumplido el horario resultante del sorteo de inscripciones, imposibilitando al usuario de inscribirse si la fecha y hora actual es menor a la fecha y hora de dicho sorteo.
- Un alumno puede inscribirse al grupo de laboratorio que más le convenga, siempre y cuando dicho grupo tenga lugares disponibles de inscripción, definiendo el número de alumnos por grupo solo los usuarios administradores del sistema en cualquier momento.
- Una vez que el alumno se ha inscrito en algún grupo de laboratorio, puede obtener del sistema el comprobante de inscripción y en su caso de calificaciones obtenidas durante el curso. Dicho comprobante será proporcionado en medio electrónico con un archivo en formato PDF que el usuario puede imprimir. Para que este documento tenga validez oficial, se recomienda que al momento de presentarse el maestro ante su respectivo grupo, este último aplique un sello oficial en cada uno de los comprobantes de los alumnos. Para asentar dicho sello, el maestro en turno deberá verificar contra la lista que le proporcione el sistema la inscripción de cada uno de los alumnos.
- El usuario con privilegios de Maestro deberá acceder al sistema al finalizar el curso para asentar las calificaciones a cada uno de los integrantes del grupo en cuestión. En este punto, dicho usuario deberá asentar forzosamente las calificaciones a todos los integrantes del grupo. Una vez que dichas calificaciones han sido asentadas, ya no podrán ser modificadas por ningún usuario.
- Cualquier reporte que sea generado incluyendo los comprobantes de inscripción y calificaciones, pedirá siempre la contraseña del usuario que lo esta solicitando o en su caso la contraseña del administrador general del sistema para determinar los privilegios que tiene dicho usuario y considerar si se presenta o no la información solicitada y proveer al

archivo de los privilegios correspondientes. Por ejemplo. Un usuario que no sea administrador del sistema, sólo podrá guardar e imprimir dichos archivos, en cambio un usuario administrador, podrá copiar, guardar, imprimir, etc. El archivo en cuestión.

- El usuario administrador puede realizar inscripciones a grupos de laboratorios a diferentes usuarios “Alumno” siempre que el grupo en cuestión tenga cupo y que el usuario tenga inscrita en su tira de materias la materia de teoría correspondiente, así mismo podrá desempeñar las mismas tareas que un usuario “Maestro” para poder cubrir la posible baja de este durante el curso.
- El usuario administrador es el encargado de mantener la información contenida en los diferentes catálogos del sistema, así como de los archivos de prácticas y apuntes y los listados de sorteo y materias de teoría inscritas proporcionados por servicios escolares.
- El listado de sorteo deberá tener exactamente el siguiente formato.
 - Un encabezado en la fila 1 con el siguiente orden.
 - **Nombre.-** El nombre del alumno (texto).
 - **Cuenta.-** El número de cuenta del alumno (ej. 094115196).
 - **D.-** El día en que se tiene que inscribir (numérico 1-9).
 - **Plantel.-** La clave del plantel.
 - **Carr.-** La clave de la carrera.
 - **Plan.-** El plan de estudios.
 - **Mdinsc.-**
 - **Horainsc.-** La hora de inscripción (numérico 0-9).
 - **Ultinsc**
 - **Inscribió**
 - Los datos a partir de la segunda fila.
- El listado de las materias de teoría inscritas deberá tener exactamente el siguiente formato.

- Un encabezado en la fila 1 con el siguiente orden.
 - **Nombre.-** El nombre del alumno (texto).
 - **Cuenta.-** El número de cuenta del alumno (ej. 094113245).
 - **Clave.-** La clave de la materia inscrita.
 - **Grupo.-** El grupo donde se ha inscrito.
- Como las fechas de inscripción a laboratorios son las mismas fechas que el período de Altas, Bajas, Cambios de servicios escolares y el sistema de laboratorios depende directamente de las materias que requieren laboratorio y que han sido inscritas por un alumno, existirá una inconsistencia en los datos del listado de materias de teoría inscritas, por lo tanto se recomienda se separe en cuanto a las fechas, las inscripciones a laboratorios y el período de Altas, Bajas y Cambios, de otra forma se podría omitir que servicios escolares envíe dicho listado, pero en su lugar, este último debería conectarse a la base de datos del sistema de laboratorios para informar en línea los movimientos realizados en el período de Altas, Bajas y Cambios y con ello se garantizaría que la información es consistente por ambas partes.

Hasta este punto, se han presentado los aspectos más importantes que han sido considerados dentro del diseño del sistema. Cabe señalar que el sistema ha sido diseñado de tal forma que el agregar nuevas características funcionales, sea de manera rápida y sin complicaciones, para ello, se puede consultar la documentación API del sistema.

Capítulo 3

Configuración del Sistema

En este apartado, se mencionan los pasos a seguir para poder instalar y configurar las herramientas que se necesitan dentro del servidor para que el sistema funcione. Se detalla también la instalación y configuración de la aplicación misma.

3.1. Requerimientos Mínimos.

En la siguiente tabla se muestran los requerimientos mínimos con que debe contar el servidor que contendrá la aplicación. Se sugiere seguir al pie de la letra todas las indicaciones que en este capítulo se describen, ya que el sistema ha sido probado siguiendo todas estas indicaciones, aunque no se descarta alguna otra configuración que el administrador del sistema decida aplicar bajo su propia responsabilidad.

Propiedad	Windows	Linux
Sistema Operativo.	Microsoft Windows 2000, XP.	*Linux Red Hat 7.2.
Procesador.	Celeron 300 Mhz. o superior.	Celeron 300 Mhz. o superior.
Espacio en Disco Duro.	500 Mb. espacio disponible en disco duro.	**100Mb en filesystem /usr. **400 Mb en filesystem /var.
Memoria Ram	128 Mb	**128 Mb de memoria física. **256 Mb en filesystem swap.
Conexión a Internet con IP Fija		
Servidor de Correos Interno.		

* Se puede utilizar alguna otra distribución de sistema operativo unix, en este caso la ubicación de archivos dentro del filesystem puede ser diferente a lo descrito en este documento.

** Se recomienda la configuración anterior aunque no es la única posible de utilizar. Seguir otra configuración es decisión con relación al conocimiento del administrador del equipo y/o sistema.

3.2. Estructura del CD de instalación.

El disco de instalación tiene la siguiente estructura.



En el directorio docs se encuentra toda la documentación del sistema (API, Manual, Diseño, Instalación).

En el directorio scripts, se encuentran los archivos útiles para la instalación como el script de la base de datos.

En el directorio src se encuentran los archivos fuentes propios del sistema.

En el directorio install se encuentran las herramientas para que funcione el sistema, tales como Apache, MySQL, PHP, etc.

3.3. Arquitectura de la aplicación.

La aplicación puede residir en un servidor con los requerimientos mínimos que se mencionaron anteriormente como lo son el sistema operativo Windows o Linux, incluso cualquier otro sistema basado en Unix, además de ello, el servidor de la aplicación debe tener instalado el servidor Web, en este caso Apache, el manejador de bases de datos MySQL, el interprete de PHP y las utilerías del proyecto de PEAR basadas en lenguaje PHP para que el sistema pueda operar.

En la figura 29 se muestra el diagrama de la arquitectura de la aplicación.

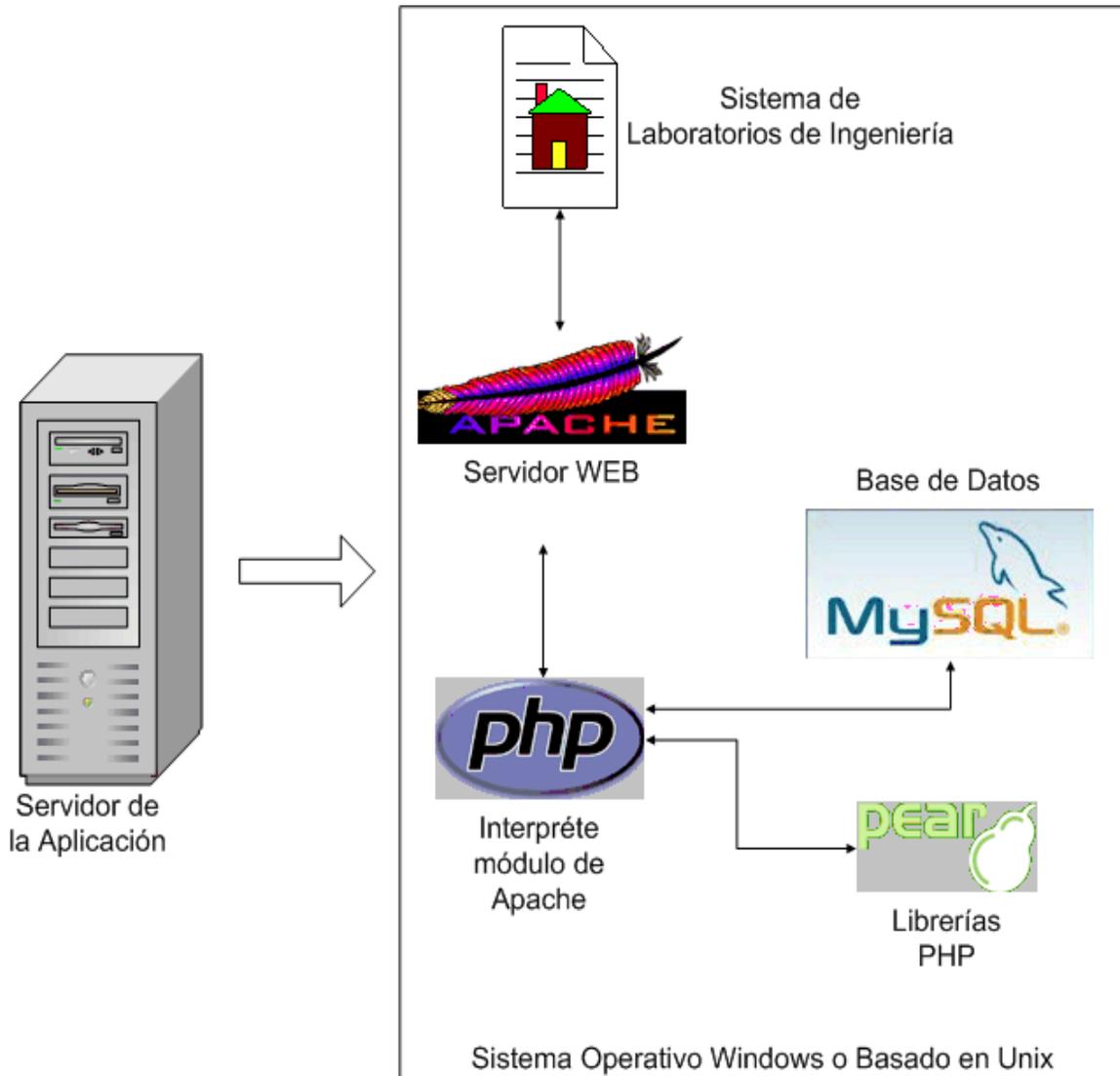


Figura 29. Arquitectura de la Aplicación.

3.4. Instalación en plataforma Windows.

A continuación se describen a detalle los pasos a seguir para llevar a cabo la instalación del sistema en la plataforma Windows. El proceso consiste en seguir el asistente de instalación como se muestra en las figuras.

3.4.1. Instalación de MySQL.

1. Abrir el archivo Setup.exe que se encuentra en la ruta install/Windows/MySQL del CD de instalación y se muestra una ventana como la de la figura 30.

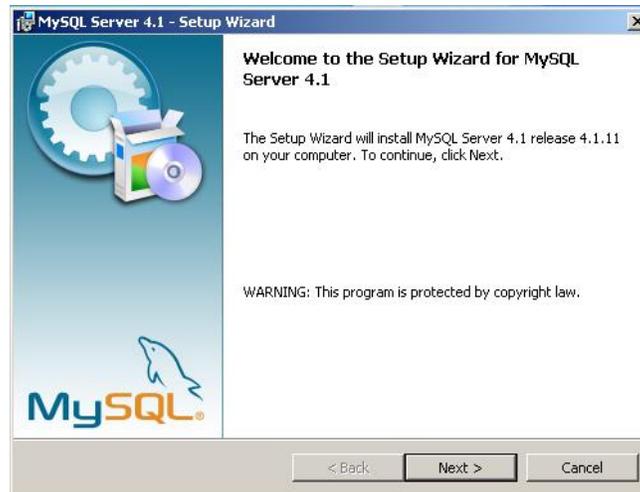


Figura 30. Paso 1 de Instalación de MySQL en Windows.

2. En este punto se debe dar click en el botón Next y se muestra la ventana de la figura 31.



Figura 31. Paso 2 de Instalación de MySQL en Windows.

3. Seleccionar la opción Complete y dar click en el botón Next de la figura 32.

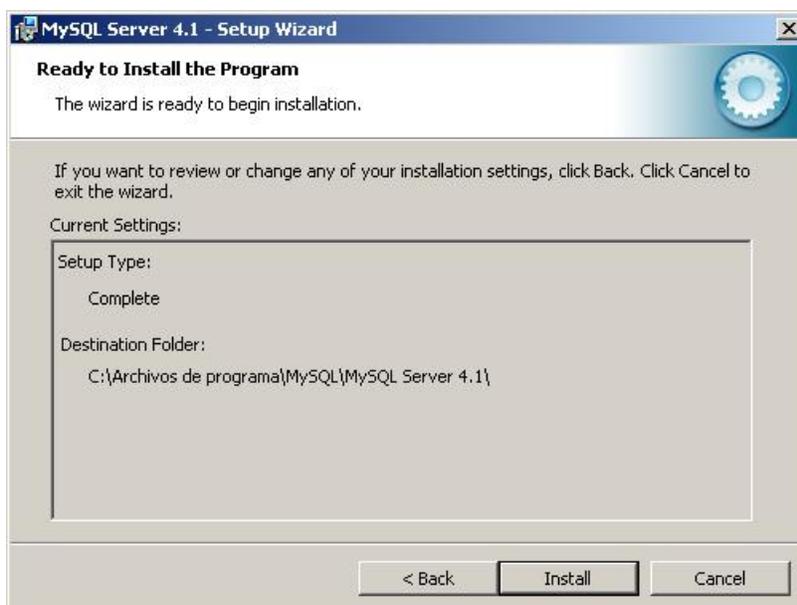


Figura 32. Paso 3 de Instalación de MySQL en Windows.

4. Dar click en el botón Install de la figura 33.



Figura 33. Paso 4 de Instalación de MySQL en Windows.

- Una vez que el asistente ha terminado, seleccionar la opción Skip Sign-Up y dar click en el botón Next de la figura 34.



Figura 34. Paso 5 de Instalación de MySQL en Windows.

- Seleccionar la opción Configure the MySQL Server now y dar click en el botón Finish de la figura 35.



Figura 35. Paso 6 de Instalación de MySQL en Windows.

7. Dar click al botón Next de la figura 36.



Figura 36. Paso 7 de Instalación de MySQL en Windows.

8. Seleccionar la opción Detailed Configuration y dar click al botón Next de la figura 37.

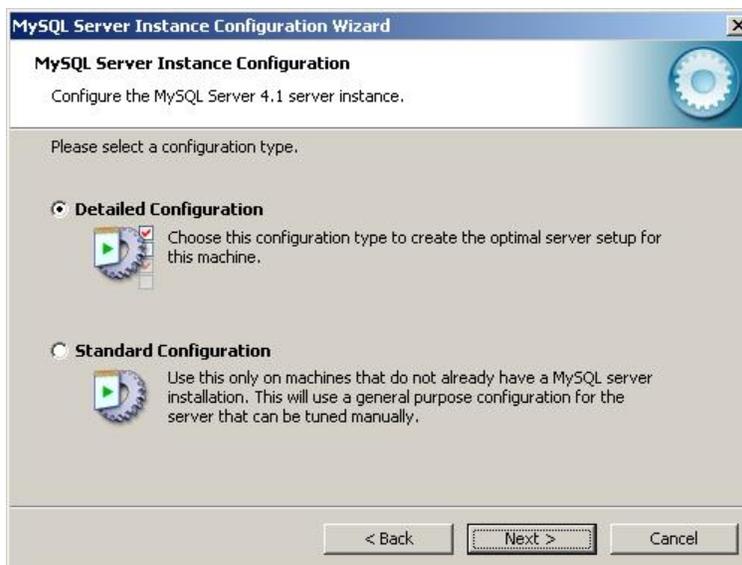


Figura 37. Paso 8 de Instalación de MySQL en Windows.

9. Seleccionar la opción Server Machine y dar click en el botón Next de la figura 38.



Figura 38. Paso 9 de Instalación de MySQL en Windows.

10. Seleccionar la opción Multifunctional Database y dar click al botón Next de la figura 39.



Figura 39. Paso 10 de Instalación de MySQL en Windows.

11. Especificar la ruta en el disco duro donde se instalará la base de datos y dar click en el botón Next de la figura 40.

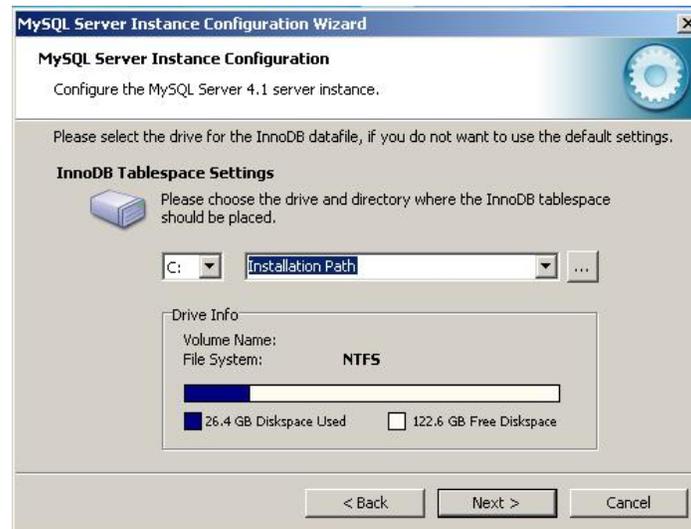


Figura 40. Paso 11 de Instalación de MySQL en Windows.

12. Seleccionar la opción Online Transaction Processing (OLTP) y dar click en el botón Next de la figura 41.



Figura 41. Paso 12 de Instalación de MySQL en Windows.

13. Seleccionar la opción Enable TCP/IP Networking en el puerto 3306 y dar click al botón Next de la figura 42.

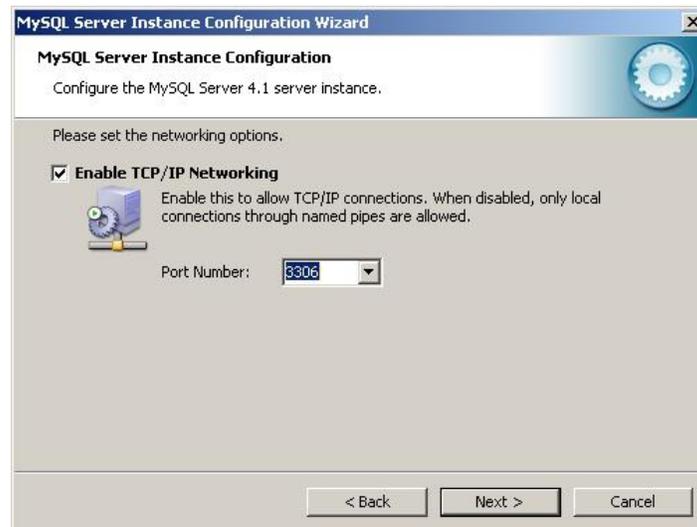


Figura 42. Paso 13 de Instalación de MySQL en Windows.

14. Seleccionar la opción Standard Carácter Set y dar click al botón Next de la figura 43.



Figura 43. Paso 14 de Instalación de MySQL en Windows.

15. Seleccionar la opción las tres opciones disponibles y dar click al botón Next de la figura 44.



Figura 44. Paso 15 de Instalación de MySQL en Windows.

16. Seleccionar la opción Modify Security Settings y teclear una contraseña para la administración de MySQL y dar click al botón Next de la figura 45.



Figura 45. Paso 16 de Instalación de MySQL en Windows.

17. Dar click en el botón Execute de la figura 46.



Figura 46. Paso 17 de Instalación de MySQL en Windows.

18. El sistema comienza a aplicar la configuración y se muestra una ventana como la de la figura 47.



Figura 47. Paso 18 de Instalación de MySQL en Windows.

- Una vez que el sistema ha terminado de configurar, dar click al botón Finish que se muestra en la figura 48.

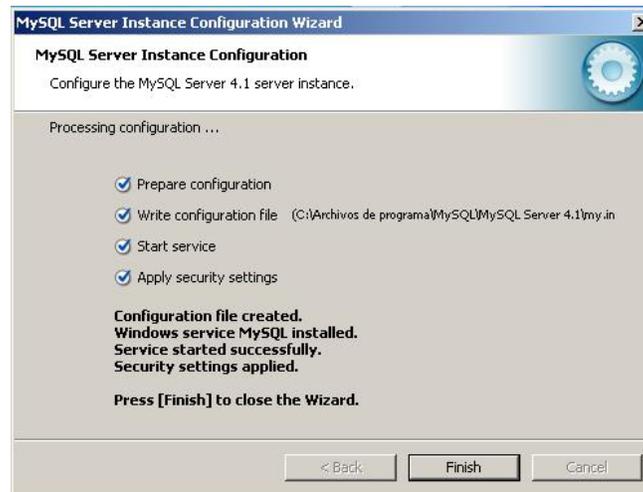


Figura 48. Paso 19 de Instalación de MySQL en Windows.

3.4.2. Instalación de Apache Web Server.

- Abrir el archivo ejecutable apache_1.3.33-win32-x86-no_src.exe que se encuentra en la ruta install/Windows/Apache, dar click al botón Next de la figura 49.

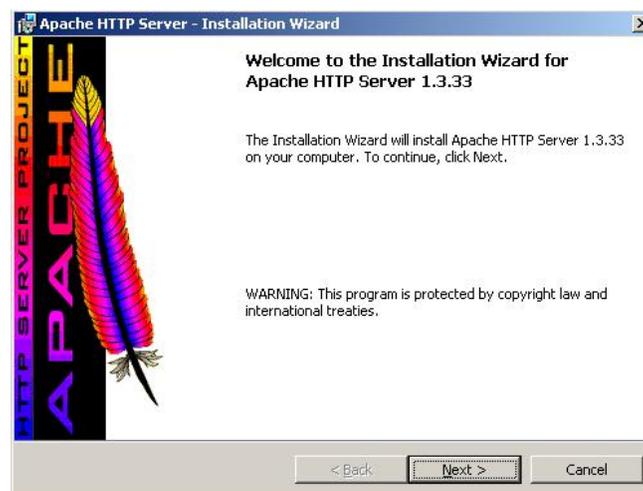


Figura 49. Paso 1 de Inscripción de Apache en Windows.

2. Aceptar el contrato de licencia y dar click al botón Next, véase la figura 50.



Figura 50. Paso 2 de Inscripción de Apache en Windows.

3. Dar click al botón Next de la pantalla como muestra la figura 51.



Figura 51. Paso 3 de Inscripción de Apache en Windows.

4. Escribir los valores de dominio, nombre del servidor Web y la dirección de correo electrónico del administrador del servidor, luego seleccionar la opción recomendada y dar click al botón Next como muestra la figura 52.



Figura 52. Paso 4 de Inscripción de Apache en Windows.

5. Seleccionar la opción Complete y dar click al botón Next, veáse figura 53.



Figura 53. Paso 5 de Inscripción de Apache en Windows.

6. Seleccionar el directorio donde se instalará Apache y dar click al botón Next, que muestra la figura 54.

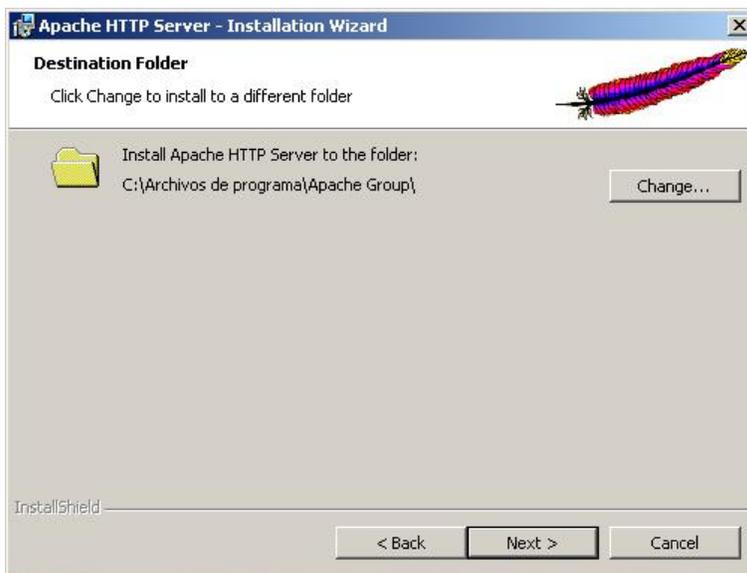


Figura 54. Paso 6 de Inscripción de Apache en Windows.

7. De la siguiente pantalla dar click al botón Install que muestra la figura 55.

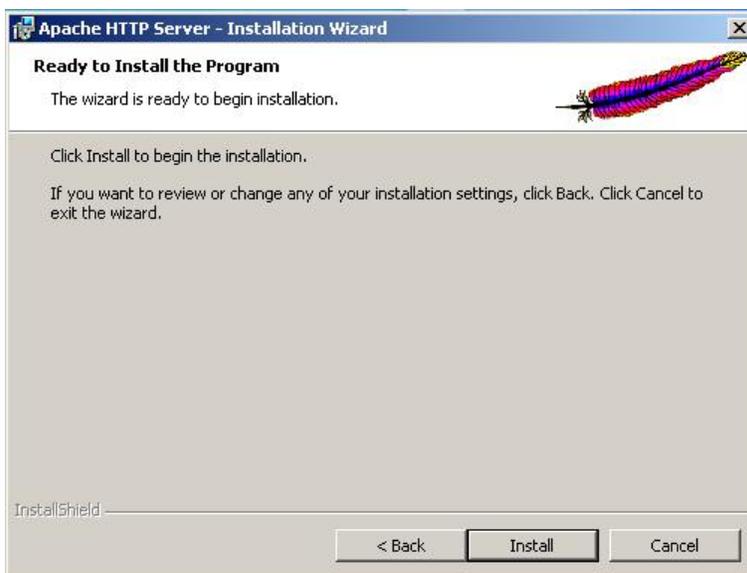


Figura 55. Paso 7 de Inscripción de Apache en Windows.

8. Durante la instalación aparece la ventana que muestra la figura 56.

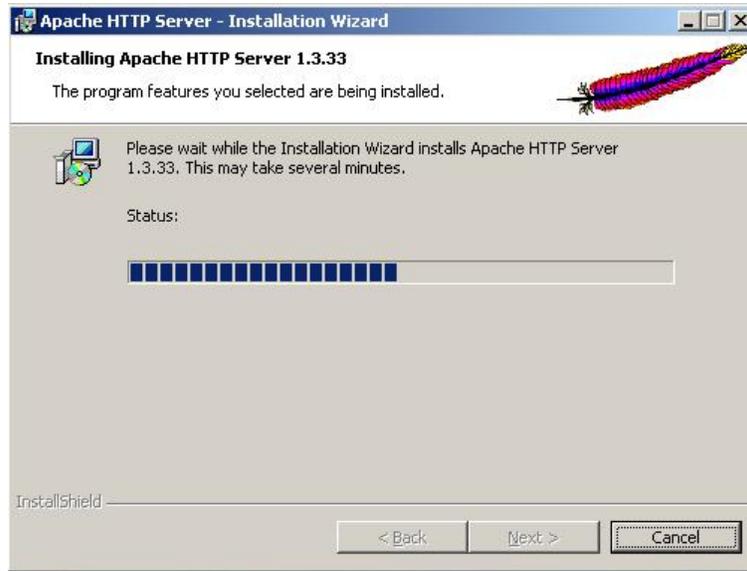


Figura 56. Paso 8 de Inscripción de Apache en Windows.

9. Cuando la instalación ha terminado, dar click al botón Finish, como la que muestra la figura 57.

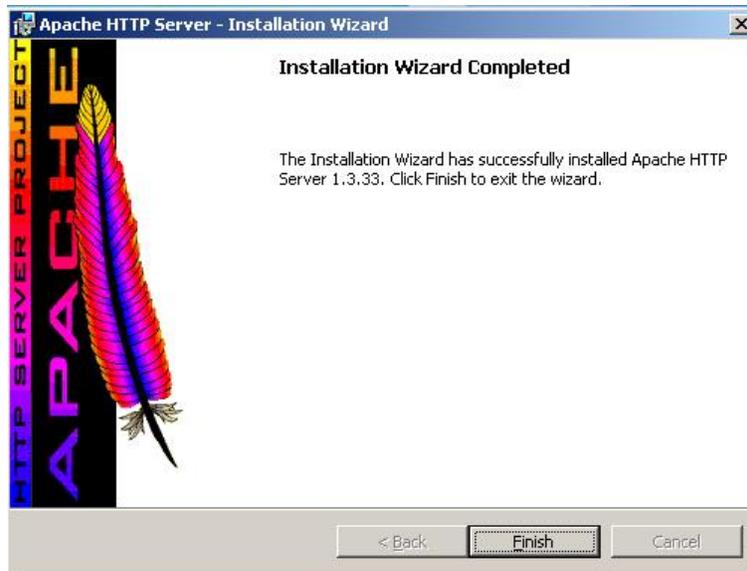


Figura 57. Paso 9 de Inscripción de Apache en Windows.

3.4.3. Instalación de PHP + PEAR.

1. Crear una carpeta en el disco duro llamada php y otra llamada tmp.
2. Descomprimir el archivo de instalación de PHP php-5.0.4-Win32.zip que se encuentra en el directorio install/Windows/PHP del CD de instalación en el disco duro del equipo en el directorio php.
3. Copiar la carpeta PEAR del CD de instalación en el directorio php.
4. Copiar todos los archivos con extensión .dll del directorio del disco duro php al directorio del sistema.(C:/Windows/system32 para Windows XP y C:/WINNT/system32 para Windows 2000).
5. Copiar el archivo php.ini-recomended que se encuentra en la carpeta php del disco duro al directorio C:/Windows o C:/WINNT según sea el caso.
6. Renombrar el archivo php.ini-recomended por php.ini.
7. Editar el archivo php.ini
8. Buscar la línea register_globals y asignarle el valor de On. Por ejemplo:

```
register_globals = On.
```

9. Buscar la línea extensión_dir y asignarle la ruta donde se encuentra el directorio ext en la carpeta php del disco duro. Por ejemplo:

```
extensión_dir = "C:/php/ext".
```

10. Buscar la línea ;extensión=php_mysql.dll y quitar el comentario con el carácter ';' quedando de la siguiente forma

```
extensión=php_mysql.dll
```

11. Buscar la línea session.save_path y asignarle el valor de la ruta donde se encuentra el directorio tmp. Por ejemplo

```
session.save_path = "C:/tmp"
```

12. Agregar al final del archivo la siguiente línea

```
include_path=".;C:\php\PEAR"
```

nótese que el valor de la propiedad, es la ruta en el disco duro del directorio donde se instaló PEAR en el punto 3.

13. Guardar y cerrar el archivo.
14. Abrir el archivo httpd.conf que se encuentra en el directorio conf donde se instaló Apache, por ejemplo: C:/Archivos de Programa/Apache Group/Apache/conf.
15. Buscar el bloque de carga de módulos y agregar la siguiente línea

```
LoadModule php5_module C:/php/php5apache.dll.
```

16. Buscar la sección de ClearModuleList y agregar la siguiente línea:

```
AddModule mod_php5.c
```

17. Buscar la línea de DirectoryIndex y agregar al final de la línea index.php, por ejemplo:

```
DirectoryIndex index.html index.php.
```

18. Buscar la sección de AddType y agregar las siguientes líneas:

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

19. Guardar y cerrar el archivo.
20. Abrir Panel de Control//Herramientas Administrativas/Servicios y reiniciar el servicio de Apache.

21. Para probar que la instalación ha tenido éxito, crear un nuevo archivo de texto llamado info.php en el directorio de instalación de apache/htdocs por ejemplo: C:/Archivos de Programa/Apache Group/Apache/htdocs con el siguiente contenido.

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

22. Abrir un navegador de Internet con la dirección http://localhost/info.php, si se muestra una página con el contenido de la figura 58, entonces la instalación ha tenido éxito, en caso contrario, es necesario verificar la configuración del sistema.

PHP Version 5.0.5



System	Windows NT PERODRIGUEZA 5.1 build 2600
Build Date	Sep 5 2005 15:50:13
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS\php.ini
PHP API	20031224
PHP Extension	20041030
Zend Extension	220040412
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v2.0.5, Copyright (c) 1998-2004 Zend Technologies



PHP Credits

Figura 58. Prueba de Instalación de PHP.

3.4.4. Instalación de la base de datos.

1. Copiar el archivo laboratorio.sql que se encuentra en la ruta scripts del CD de instalación al directorio tmp del disco duro.
2. Abrir una ventana de MS-dos y entrar al directorio donde se instaló MySQL. Por ejemplo: cd c:/Archivos de Programa/MySQL/MySQL Server 4.1/bin y ejecutar el siguiente comando:

```
mysql -u root -p.
```

3. Cuando el sistema solicite el password, teclear la contraseña que fue proporcionada en el punto 15 de la instalación de MySQL.
4. Una vez dentro del gestor, teclear el siguiente comando:

```
mysql>source c:/tmp/laboratorio.sql;
```

5. Para comprobar que la base de datos se ha instalado correctamente en el gestor, teclear show databases; y debe aparecer la base de datos laboratorio.

3.4.5. Instalación del Sistema de Laboratorios de Ingeniería.

1. Copiar la carpeta laboratorio que se encuentra en el directorio src del CD de instalación al directorio destinado para Apache, por ejemplo:

```
C:/Archivos de Programa/Apache Group/Apache/htdocs
```

3.5. Instalación en plataforma Linux.

A continuación se describen a detalle los pasos a seguir para llevar a cabo la instalación del sistema en la plataforma Linux.

3.5.1. Instalación de MySQL.

1. Crear el directorio `/root/install`
2. Copiar el archivo `mysql-4.1.11.tar.gz` que se encuentra en la ruta `install/unix/MySQL` del CD de instalación al directorio `/root/install`
3. Teclear los siguientes comandos como usuario `root`.

```
# groupadd mysql
# useradd -g mysql mysql
# tar -zxvf mysql-4.1.11.tar.gz
# cd mysql-4.1.11
# ./configure --prefix=/var/lib/mysql
# make
# make install
# cp support-files/my-medium.cnf /etc/my.cnf
# cd /var/lib/mysql
# bin/mysql_install_db --user=mysql
# chown -R root .
# chown -R mysql var
# chgrp -R mysql .
# bin/mysqld_safe --user=mysql &
# bin/mysqladmin -u root password NUEVA_CONTRASEÑA
```

3.5.2. Instalación de APACHE + PHP + PEAR.

1. Copiar el archivo `apache_1.3.33.tar.gz` que se encuentra en la ruta `install/unix/Apache` del CD de instalación al directorio `/root/install`
2. Copiar el archivo `php-5.0.4.tar.gz` que se encuentra en la ruta `install/unix/PHP` del CD de instalación al directorio `/root/install`
3. Teclear los siguientes comandos como usuario `root`.

```
# tar -zxvf apache_1.3.33.tar.gz
# tar -zxvf php-5.0.4.tar.gz
```

```
# cd apache_1.3.33
# ./configure
# cd ../php-5.0.4
# ./configure --with-mysql=/var/lib/mysql --with-apache=../apache_1.3.33 --disable-libxml
# make
# make install
# cd ../apache_1.3.33
# ./configure --activate-module=src/modules/php5/libphp5.a
# make
# make install
```

4. Copiar el contenido del directorio install/unix/PEAR del CD de instalación a /usr/local/lib/php.

```
# cp php.ini-dist /usr/local/lib/php.ini
```

5. Buscar la línea de DirectoryIndex y agregar al final de la línea index.php, por ejemplo: DirectoryIndex index.html index.php.
6. Buscar la sección de AddType y agregar las siguientes líneas:

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

7. Guardar y cerrar el archivo
8. Editar el archivo php.ini
9. Buscar la línea register_globals y asignarle el valor de On. Por ejemplo:

```
register_globals = On.
```

10. Buscar la línea extension_dir y asignarle la ruta donde se encuentran las librerías del sistema. Por ejemplo: extension_dir = /usr/local/lib/php.

11. Buscar la línea `;extension=php_mysql.dll` y quitar el comentario con el carácter `;` quedando de la siguiente forma

```
extensión=php_mysql.dll
```

12. Buscar la línea `session.save_path` y asignarle el valor de la ruta donde se encuentra el directorio `tmp`. Por ejemplo

```
session.save_path = /tmp
```

13. Agregar al final del archivo la siguiente línea

```
include_path=".:usr/local/lib/php".
```

14. Guardar y cerrar el archivo.
15. Teclear el siguiente comando:

```
# /usr/local/apache/bin/apachectl start &
```

16. Para probar que la instalación ha tenido éxito, crear un nuevo archivo de texto llamado `info.php` en el directorio `/usr/local/apache/htdocs`.

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

17. Abrir un navegador de Internet con la dirección `http://localhost/info.php`, si se muestra una página con el contenido de la figura 59, entonces la instalación ha tenido éxito, en caso contrario, es necesario verificar la configuración del sistema.

PHP Version 5.0.5


System	Windows NT PERODRIGUEZA 5.1 build 2600
Build Date	Sep 5 2005 15:50:13
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS\php.ini
PHP API	20031224
PHP Extension	20041030
Zend Extension	220040412
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v2.0.5, Copyright (c) 1998-2004 Zend Technologies



PHP Credits

Figura 59. Prueba de Instalación de PHP.

3.5.3. Instalación de la base de datos.

1. Copiar el archivo laboratorio.sql que se encuentra en la ruta scripts del CD de instalación al directorio /tmp.
2. Entrar al gestor de la base de datos con el comando `mysql -u root -p`.
3. Cuando el sistema solicite el password, teclear la contraseña que fue proporcionada al instalar MySQL.
4. Una vez dentro del gestor, teclear el siguiente comando:

```
mysql>source /tmp/laboratorio.sql;
```

5. Para comprobar que la base de datos se ha instalado correctamente en el gestor, teclear `show databases;` y debe aparecer la base de datos laboratorio.

3.5.4. Instalación del Sistema de Laboratorios de Ingeniería.

- Copiar la carpeta laboratorio que se encuentra en el directorio src del CD de instalación al directorio destinado para Apache, por ejemplo: `/usr/local/apache/htdocs`
- Cambiar los permisos de los archivos del sistema con el siguiente comando:

```
# chown -R nobody:nobody laboratorios
# chmod -R 755 laboratorios
```

3.6. Configuración del Sistema.

El sistema cuenta con un **archivo de configuración** llamado `config.properties` que se encuentra bajo el directorio `/include` del directorio donde se instaló el sistema, a continuación se detallan las propiedades de dicho archivo de configuración.

Propiedad	Descripción
<code>dsn</code>	Cadena de conexión a la base de datos, las partes que pueden ser configuradas son <code>user</code> , <code>password</code> , y <code>host</code> . Estas partes corresponden a los usuarios de la base de datos. <code>mysql://user:password@host/</code>
<code>database</code>	El nombre de la base de datos donde radica el sistema.
<code>tableuser</code>	El nombre de la tabla donde se almacenan los usuarios para autenticarse en el sistema.
<code>usernamecol</code>	El nombre de la columna en la base de datos donde se almacena el nombre de usuario.
<code>passwordcol</code>	El nombre de la columna en la base de datos donde se almacena la contraseña del usuario.
<code>db_fields</code>	Los campos de la base de datos que serán incluidos en la sesión.

Propiedad	Descripción
cryptType	El tipo de encriptamiento del campo de contraseña del usuario.
sessionName	El nombre de la sesión del usuario.
emailFrom *	La dirección de correo electrónico desde la cual el sistema envía los correos.
emailTo *	La dirección de correo electrónico a la cual los usuarios envían mensajes de correo al administrador del sistema.
emailHost *	El servidor de correos que será utilizado para enviar los mensajes.
emailPort *	El puerto por el cual se envían los mensajes de correo.
emailAuth	Determina si se necesita autenticar en el servidor de correos para leer los mensajes..
emailUsername *	El nombre de usuario dentro del servidor de correos.
emailPassword *	La contraseña del usuario dentro del servidor de correos.
hostname *	El nombre del servidor donde radica el sistema.
savePassword	La ruta relativa dentro del servidor donde se almacenan las contraseñas del sistema.
savePasswordReportes	La ruta relativa dentro del servidor en el directorio de reportes donde se almacenan las contraseñas del sistema.
directoryPractice	La ruta relativa dentro del servidor donde se almacenan las prácticas de los laboratorios.
directoryApunte	La ruta relativa dentro del servidor donde se almacenan los apuntes de los laboratorios.
directorySorteo	La ruta relativa dentro del servidor donde se almacenan los listados de sorteo del sistema.
directoryMatriasInscritas	La ruta relativa dentro del servidor donde se almacenan los listados de las materias inscritas del sistema.
contactName *	El nombre del administrador y contacto del sistema.
contactEmail *	La dirección de correo electrónico del administrador y contacto del sistema.
passAdmin	La contraseña del administrador para ver cualquier reporte del sistema con todos los privilegios.

Los campos marcados con * son los mínimos requeridos a ser configurados para el funcionamiento del sistema, los restantes también pueden ser modificados, pero

se deja a consideración del administrador del sistema, una configuración errónea, podría provocar que el sistema dejara de funcionar.

3.7. Configuración de Seguridad.

En este apartado se explican las configuraciones básicas de seguridad en el servidor donde reside el sistema.

3.8. Seguridad en Windows.

Considerando que el servidor donde reside el sistema es de tipo Web Server, será necesario tener una cuenta con privilegios restringidos para monitorear los logs y hacer cambios solo en los archivos que comprenden al sistema mismo. El usuario administrador del sistema operativo deberá ser usado sólo cuando se requiera instalar o desinstalar alguna aplicación o en su caso cuando se realice alguna configuración que necesite de todos los privilegios del sistema, además de la actualización constante de parches de seguridad del sistema operativo para lo cual es recomendable utilizar Windows update, así como también la instalación de un programa firewall para denegar el acceso a usuarios y/o aplicaciones no deseadas.

3.9. Seguridad en Linux.

Bajo los sistemas Linux, se recomiendan tres puntos fundamentales para tomar en cuenta en la seguridad del sistema operativo. A continuación se describe la configuración que se aplica con dichos puntos a los sistemas Linux.

El primer aspecto a considerar en la seguridad de este tipo de sistemas, es en cuanto a los puertos que se tendrán abiertos en el servidor, para lo cual se trabaja sobre el archivo `/etc/services`, a continuación se muestra parte de un contenido típico de este archivo.

```

http      80/tcp    www www-http # WorldWideWeb HTTP
http      80/udp    www www-http # HyperText Transfer Protocol
Kerberos  88/tcp    kerberos5 krb5 # Kerberos v5
Kerberos  88/udp    kerberos5 krb5 # Kerberos v5
Supdup    95/tcp
Supdup    95/udp
Hostname  101/tcp   hostnames    # usually from sri-nic
Hostname  101/udp   hostnames    # usually from sri-nic
iso-tsap  102/tcp   tsap         # part of ISODE.
csnet-ns  105/tcp   cso          # also used by CSO name server
csnet-ns  105/udp   cso
# unfortunately the poppassd (Eudora) uses a port which has already
# been assigned to a different service. We list the poppassd as an
# alias here. This should work for programs asking for this service.
# (due to a bug in inetd the 3com-tsmux line is disabled)
#3com-tsmux 106/tcp   poppassd
#3com-tsmux 106/udp   poppassd

```

En este archivo se encuentran listados todos los puertos conocidos mediante los cuales el sistema tendrá comunicación directa de entrada – salida hacia cualquier elemento de la red LAN, incluso WAN. En este punto se recomienda editar este archivo y poner comentario al inicio de todas las líneas del archivo mediante un carácter '#' y dejar sin comentario, todos los puertos a los cuales el sistema tendrá acceso, para lo cual, este archivo deberá tener desbloqueados los puertos para ssh, http, imap3, pop, pop2, pop3, smtp, mysql.

El segundo punto importante sobre el tema de seguridad en Linux es mediante el uso de los archivos /etc/hosts.allow y /etc/hosts.deny. El primero de estos archivos determina los servicios que estarán disponibles a determinados equipos de la red identificados por su dirección IP, en tanto que el segundo archivo se encarga de denegar los servicios listados en el mismo los diferentes equipos de la red que sean descritos en dicho archivo.

A continuación se muestra un contenido típico de estos dos archivos.

```
#
# hosts.allow This file describes the names of the hosts which are
#           allowed to use the local INET services, as decided
#           by the '/usr/sbin/tcpd' server.
#
sendmail : unam.mx/192.168.* : allow
```

El cual quiere decir que todos los usuarios o elementos que se encuentren bajo el dominio de unam.mx y/o pertenezcan al segmento de red 192.168, tendrán acceso al correo que proporcione dicho servidor. En tanto que el siguiente archivo niega el acceso a todos los servicios que no han sido contemplados en el archivo `/etc/hosts.allow`

```
#
# hosts.deny This file describes the names of the hosts which are
#           *not* allowed to use the local INET services, as decided
#           by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow. In particular
# you should know that NFS uses portmap!

portmap:ALL
lockd:ALL
mountd:ALL
rquotad:ALL
statd:ALL
```

El cual quiere decir que se deniega el acceso a los Servicios de portmap, lockd, mountd, rquotad y statd a todos los usuarios que no estén comprendidos para utilizar dichos servicios en el archivo `/etc/hosts.allow`.

El último punto importante a considerar en la seguridad de servidores Linux es el uso de firewall, esto se logra mediante la configuración de las iptables.

Como se ha descrito anteriormente, en los párrafos del presente capítulo, la instalación y configuración de las herramientas y el sistema mismo para su buen funcionamiento y aunque no es la única forma de hacerlo, se recomienda seguir estos pasos.

Se recomienda consultar la documentación de cada una de las herramientas en lo que a instalación y configuración se trata, en caso de que el administrador del servidor y/o del sistema, decida aplicar una configuración diferente a la establecida líneas atrás.

Capítulo 4

Manual de Usuario y Administrador

Este capítulo tiene la finalidad de mostrar al usuario del sistema de laboratorios de ingeniería, la forma en que deberá ser usado para llevar a cabo las actividades contempladas dentro del mismo. A continuación se describe paso a paso cada uno de los módulos que comprenden a dicho sistema para poder atacar de esta forma la problemática actual que presenta la Facultad de Estudios Superiores Aragón en el proceso de inscripciones a laboratorios del área de ingeniería y en general toda la gestión correspondiente. Para lo cual, se presenta en como preámbulo, la forma en que se lleva a cabo la navegación en cualquier sitio Web para posteriormente, entrar de lleno a la descripción de la operabilidad del sistema de laboratorios de ingeniería.

4.1. Navegación básica Web.

En primer lugar, se debe abrir el programa navegador o browser mediante el cual serán visualizadas las páginas consultadas. Existen diversos programas de este tipo que cuentan con un funcionamiento básico muy similar entre si y las variaciones encontradas son debido a la plataforma del sistema operativo y/o a los estándares que soportan en cuanto a la lectura de los diferentes comandos pertenecientes a los lenguajes de programación como JavaScript y VBScript. Entre dichos navegadores se encuentran Internet Explorer, Netscape, Mozilla, FireBox y Opera como los más comunes y comerciales entre los usuarios de internet.

La forma en que se abre dicho programa navegador, depende directamente del sistema operativo que tenga instalado la computadora desde la cual se intenta navegar dentro de internet. Normalmente se debe ingresar al modo gráfico de dicho sistema operativo para poder visualizar la información contenida en las páginas Web, aunque existen comandos o aplicaciones propias del sistema operativo para llevar a cabo la navegación en modo texto, como el comando lynx en sistemas unix. Para los usuarios de windows no es necesario ingresar en algún modo en específico debido a que este sistema operativo abre por default la

sesión en su forma gráfica. Estando dentro de la sesión gráfica del sistema operativo, basta con dar doble click al icono del programa que se encuentra en el escritorio del sistema o desde el menú de inicio, en la sección de programas. Una vez abierto el navegador, hay que escribir la dirección de la página de internet en el espacio destinado para ello y dar click en el botón “Ir” o teclear enter, normalmente dicho espacio se encuentra en la parte superior del programa con la etiqueta de Dirección, Address ó URL. La dirección normalmente se encuentra compuesta por el nombre del equipo seguido del nombre del dominio separado por un punto y en ocasiones el nombre del archivo de alguna página, por ejemplo: <http://www.google.com.mx> ó <http://ingenieria.aragon.unam.mx/index.php>.

The image shows a screenshot of a web browser window titled "Servicios en Línea: Solicitud de Tarjeta de Crédito - Microsoft Internet Explorer". The page content is a form for a credit card application, divided into two main sections: "Identificación del Cliente" and "Datos Personales e Información Domiciliaria".

Identificación del Cliente: This section contains a question "¿Es Cliente Banamex?" with radio buttons for "NO" and "SI". A "RadioButton" label points to the "SI" button. A "Link" label points to the "Banamex" text.

Datos Personales e Información Domiciliaria: This section contains various input fields and controls:

- Text input fields for "Apellido Paterno", "Apellido Materno", and "Nombre(s)". An "InputText" label points to the "Apellido Paterno" field.
- Text input fields for "Teléfono Celular" with options "con 044" and "sin 044".
- A text input field for "Correo Electrónico".
- A "Checkbox" labeled "Quiero recibir promociones en mi e-mail" which is checked.
- Dropdown menus for "Sexo", "Nacionalidad", "País", "Tipo de Vivienda", and "Escolaridad".
- Text input fields for "Domicilio: Calle", "Nº Ext.", and "Nº Int.". A "Select" label points to the "Estado" dropdown menu.
- Text input fields for "Colonia ó Fraccionamiento" and "C.P.". A "Select" label points to the "Estado" dropdown menu.
- Text input fields for "Deleg./Mpo./Población" and "Estado".
- Text input fields for "Fecha de Nacimiento" with dropdowns for "Día" and "Mes".
- Text input fields for "Tiempo de Residencia" (in "Años") and "Dependientes Económicos".

Figura 60. Formulario de ejemplo.

Hasta este momento, el navegador muestra en pantalla la página Web contenida en la dirección especificada anteriormente, dicha página esta compuesta por diferentes controles alineados entre si a lo que se le conoce dentro de internet como formulario. En la figura 60 se muestra un formulario de ejemplo que compone la información contenida en una página.

En las siguientes líneas, se describen los controles encontrados en el formulario anterior, que son los mínimos por comprender en cuanto a su forma de navegación, para poder llevar a cabo la operación dentro del sistema de laboratorios de ingeniería al que comprende este escrito.

4.1.1. Control de Entrada de Texto.

Este es el control más sencillo dentro del formulario y como su nombre lo indica, se trata de una caja destinada para que el usuario escriba algún dato para ser enviado posteriormente al servidor. También es conocido como `InputText`. En la figura 61 se encuentra un control `InputText` para escribir datos del usuario.



Figura 61. Control `InputText`.

4.1.2. Control de Lista de Selección.

El control de Lista de Selección o `Select`, se trata de una lista de posibles valores que el usuario puede seleccionar para llevar a cabo alguna tarea en específico, como puede ser el registro de datos en el servidor de la aplicación. La forma de navegar dentro de este control es dar un click sobre la punta de flecha de color negro que se encuentra al lado derecho del control y en ese momento, se abre la lista de valores contenidos por el mismo, por último, basta con dar otro click en el valor que el usuario desea. En la figura 62, se muestra un control de este tipo.



Figura 62. Control `Select`.

4.1.3. Control de Casilla de Verificación.

Se trata de un recuadro o casilla, mediante la cual, el usuario puede marcar algún valor de forma opcional, es decir, si la casilla está seleccionada, entonces el usuario cuenta con el dato que representa dicho control, en caso contrario, si la casilla no está seleccionada, el usuario no tiene asignado el dato en cuestión. La forma de navegar dentro de este control, es simplemente dar un click en la casilla de verificación para poner o quitar la selección a este control. En la figura 63 se muestra un control de casilla de selección.



Figura 63. Control Checkbox.

4.1.4. Control de Botón de Radio.

Este control es una casilla redonda que funciona de forma muy similar al control checkbox, con la diferencia de que el control RadioButton ó botón de radio, no es posible quitar la selección una vez que ha sido marcada. Para seleccionar un control de este tipo, basta con dar un click sobre dicho control. En la figura 64 se muestra un control de tipo RadioButton.

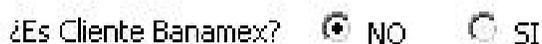


Figura 64. Control RadioButton.

4.1.5. Control de Enlace.

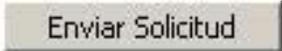
Este control sirve para reenviar al usuario a otra página, sitio o sección dependiendo al destino asociado al texto que representa dicho control, para lo cual, el usuario simplemente da click al texto que aparece de color azul y subrayado o en el cual, el cursor del ratón cambia de forma. En la figura 65 se muestra un control de tipo link para enviar al usuario a otra página.



Figura 65. Control Link.

4.1.6. Control de Botón.

Este último control, es el encargado de enviar toda la información recabada dentro del formulario, hacia el servidor para que sea procesada, almacenada, consultada, etc., para lo cual simplemente el usuario da un click en dicho botón y la información será enviada. En la figura 66 se muestra un control de tipo Button.



Enviar Solicitud

Figura 66. Control Button.

4.1.7. Navegación Especial.

Hasta este punto, se han mencionado los controles de navegación básicos dentro de una página Web y aunque no son los únicos que existen, son al menos los mínimos necesarios para la operación del sistema de laboratorios de ingeniería, aunado a esto, existen dentro de internet algunos sitios con cierto grado de dinamismo que proporcionan al usuario los medios necesarios para interactuar con el sistema paso a paso dentro del mismo. Un ejemplo claro de esto es cuando se presenta un formulario donde es necesario seleccionar el estado del donde reside el usuario, si la página es consultada solo por usuarios de un mismo país, no existe mayor problema que el llenado de la forma, sin embargo, el proceso se complica si es visitada por usuarios de distintos países, en este caso, el usuario debe seleccionar en primer instancia, el país al que pertenece, para después desplegar el listado de estados correspondientes a dicho país. Esto se logra con ciertas propiedades asignadas a los controles básicos mencionados anteriormente, con lo cual, al navegar sobre ellos, la página comienza a

recargarse con el nuevo valor seleccionado para poder desplegar la información subsecuente. Otra variante dentro de la navegación de páginas Web, se presenta con los objetos embebidos en dichos sitios, por ejemplo, cuando en la página se despliega un documento de Word en el navegador mismo sin que se abra la ventana del programa nativo, es decir Microsoft Word. En este momento se trata de un documento de Word inmerso en la página Web, la navegación de dicho documento, se lleva a cabo de la misma forma en que se manipula un documento desde el programa nativo.

4.2. Acceso al Sistema.

El primer paso para ingresar al sistema es abrir un navegador de páginas Web como Internet Explorer o Netscape y escribir en el campo de dirección lo siguiente: <http://informatica.aragon.unam.mx/laboratorios> como se muestra en la figura 67.



Figura 67. Página Inicial del Sistema.

4.3. Menús del Sistema.

Como se puede observar, se tiene acceso a dos menús principales sin tener que estar firmado dentro del sistema, en la figura 68 se muestran las opciones contenidas en dichos menús, así como una descripción de las acciones que se disparan al seleccionar dichas opciones.



Figura 68. Menú Sistema.

4.3.1. Sistema.

Este es el primer menú del sistema y contiene las opciones de “Inicio”, “Alta en el Sistema”, “Acceso al Sistema” y “Salir”.

4.3.2. Inicio.

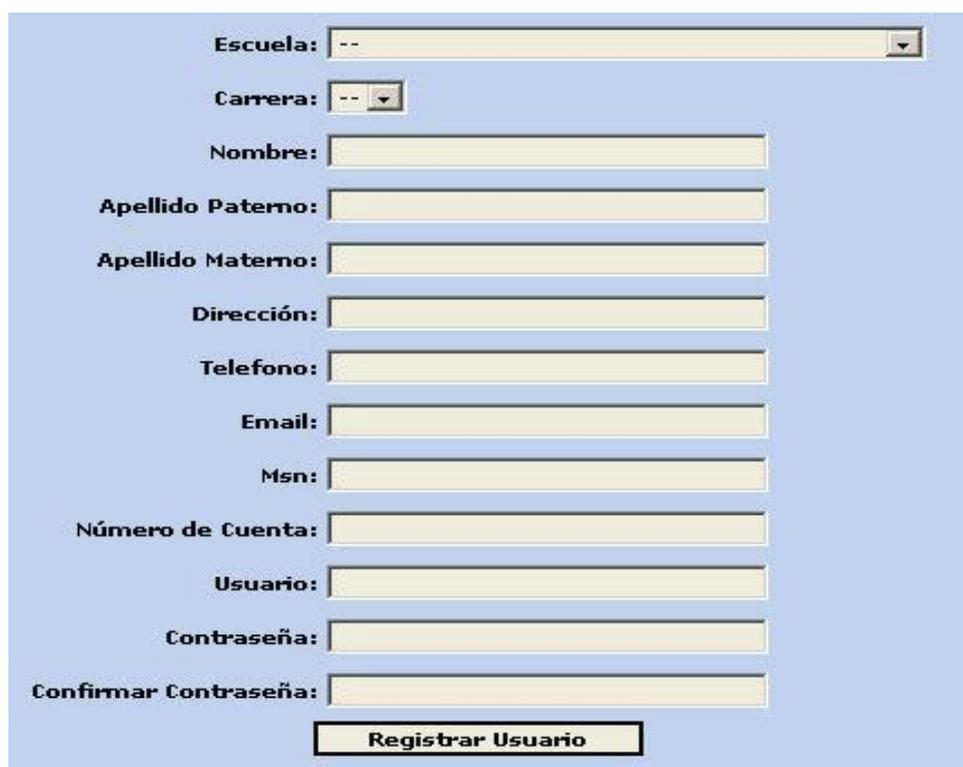
Como su nombre lo indica, esta opción se encarga de llevar al usuario a la página inicial del sistema, sin importar en que módulo del mismo se encuentre dicho usuario.

4.3.3. Alta en el Sistema.

Al seleccionar esta opción, el usuario podrá darse de alta en el sistema para poder tener acceso a las opciones donde se necesita que sea registrado antes de hacer uso de las mismas. En la figura 69 se muestra el formulario de registro de usuarios.

Es importante señalar que desde este punto del documento y hasta el final del mismo, para todos los formularios de los que el sistema obtiene la información necesaria para su operación, el mismo sistema se encarga de realizar una serie de validaciones de los datos que le son proporcionados. Esta validación puede ir desde la verificación de un dato obligatorio, el formato de algún campo específico (si el campo debe ser numérico o alfanumérico), el formato de la dirección de correo electrónico, etc.

Una vez que el sistema ha llevado a cabo esta validación y ha encontrado algún error en dicho proceso, habilita una ventana donde se informa de los errores encontrados y al mismo tiempo, pone los campos erróneos en color amarillo para su pronta ubicación.



Formulario de Registros de Usuarios. El formulario contiene los siguientes campos:

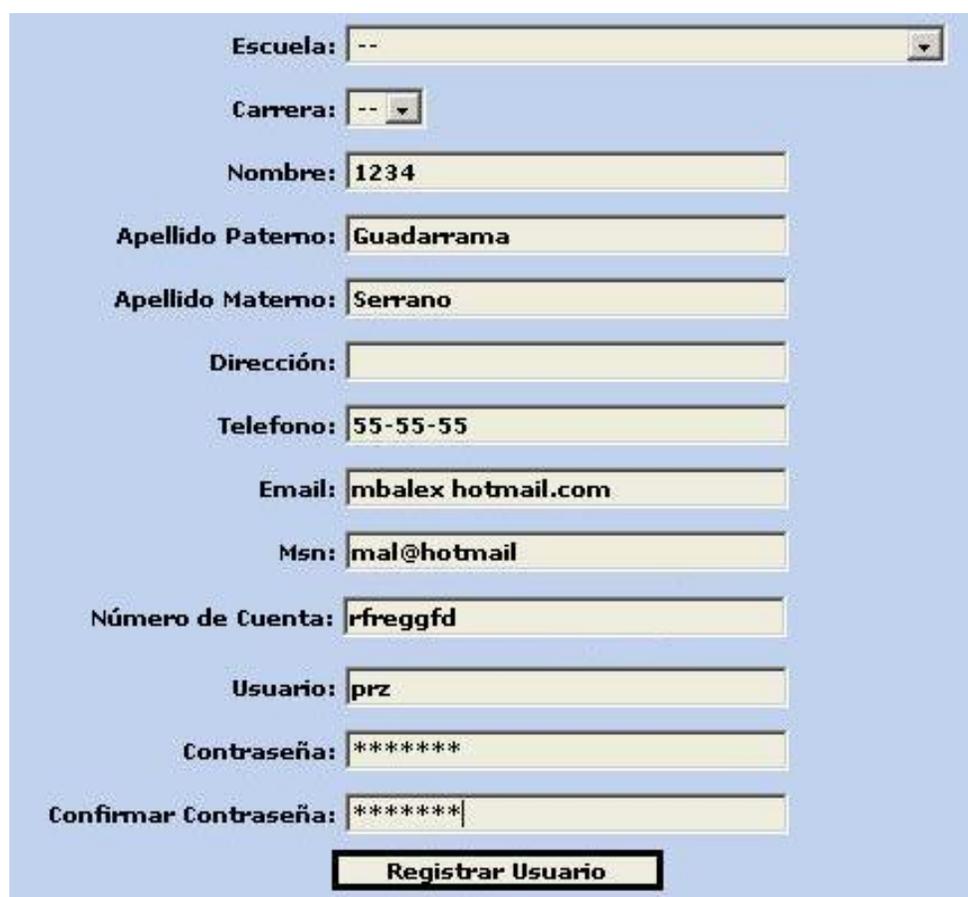
- Escuela: -- (menú desplegable)
- Carrera: -- (menú desplegable)
- Nombre: (campo de texto)
- Apellido Paterno: (campo de texto)
- Apellido Materno: (campo de texto)
- Dirección: (campo de texto)
- Telefono: (campo de texto)
- Email: (campo de texto)
- Msn: (campo de texto)
- Número de Cuenta: (campo de texto)
- Usuario: (campo de texto)
- Contraseña: (campo de texto)
- Confirmar Contraseña: (campo de texto)

Botón: Registrar Usuario

Figura 69. Formulario de Registros de Usuarios.

Cabe mencionar que esta opción esta disponible para ser usada sólo por los usuarios tipo “Alumno”, en caso contrario para dar de alta en el sistema a usuarios con privilegios de “Administrador” o “Maestro”, estos deberán ser registrados por un usuario “Administrador” ya existente en el sistema.

En la figura 70, se muestra el formulario anterior con datos erróneos y posteriormente el resultado que da el proceso de validación del sistema.



The image shows a user registration form with the following fields and values:

- Escuela: --
- Carrera: --
- Nombre: 1234
- Apellido Paterno: Guadarrama
- Apellido Materno: Serrano
- Dirección: (empty)
- Telefono: 55-55-55
- Email: mbalex hotmail.com
- Msn: mal@hotmail
- Número de Cuenta: rfreggfd
- Usuario: prz
- Contraseña: *****
- Confirmar Contraseña: *****

At the bottom of the form is a button labeled "Registrar Usuario".

Figura 70. Formulario de Registro de Usuarios con datos erróneos.

Ahora se muestra el error en la figura 71 que genera el sistema con los datos erróneos del formulario anterior

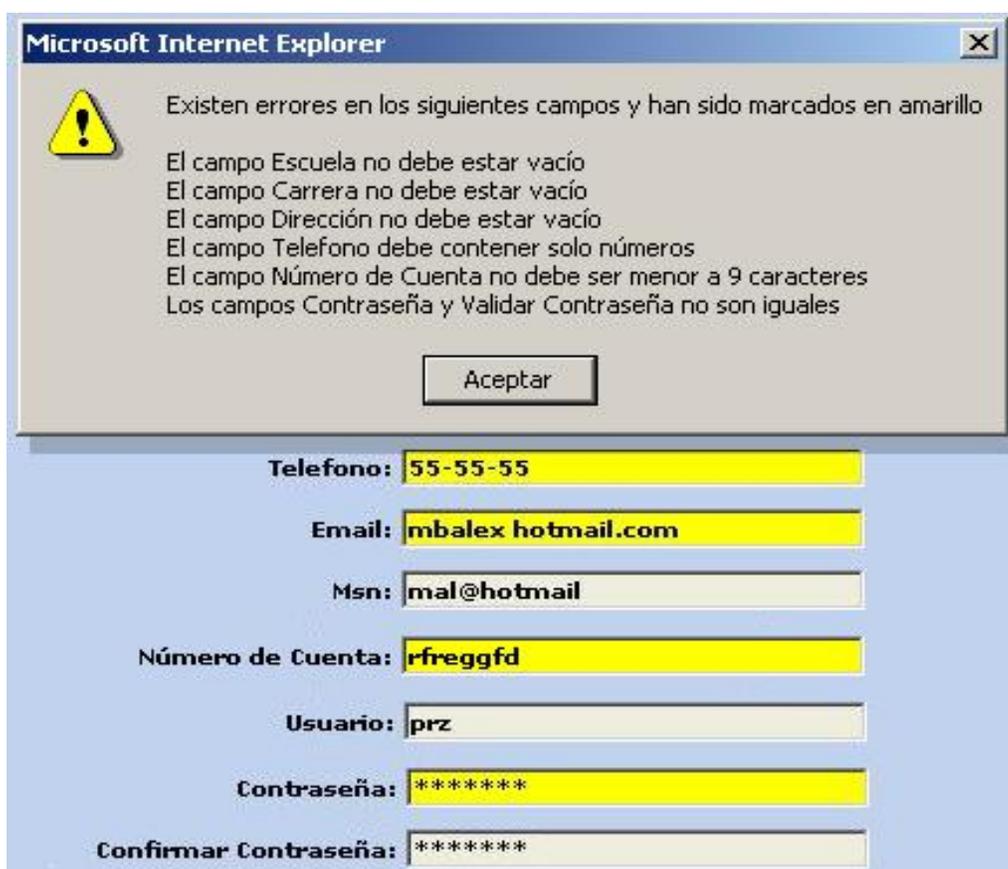


Figura 71. Error en el formulario de Registro de Usuarios.

Por último, en la figura 72, se muestra el resultado que arroja el sistema cuando se proporcionan datos correctos

Usted ha sido registrado en el sistema, recibirá un email para realizar la confirmación de inscripción y pueda hacer uso del servicio.

Figura 72. Mensaje de registro correcto al sistema.

El usuario recibe un e-mail de confirmación a la cuenta proporcionada durante el registro y para completar su inscripción al sistema, es necesario que siga las instrucciones contenidas en dicho mensaje, una vez hecho esto, se puede hacer uso del sistema.

4.3.4. Acceso al Sistema.

Una vez que el proceso de inscripción al sistema ha sido terminado, el usuario puede hacer uso del mismo, para lo cual es necesario ir al menú Acceso al Sistema e ingresar el nombre de usuario y contraseña proporcionados en el formulario de registro. En la figura 73 se muestra el formulario para ingresar al sistema.

El formulario de acceso al sistema tiene un fondo azul claro. En la parte superior izquierda, hay un campo de texto etiquetado "Usuario:" con un cursor visible. Debajo de él, hay otro campo de texto etiquetado "Contraseña:". A la derecha de cada campo hay un botón "Entrar" con un borde negro. En la parte inferior del formulario, hay un enlace de texto que dice "¿Olvidó su contraseña?" en color magenta.

Figura 73. Formulario de Acceso al Sistema.

Si el usuario no recuerda su contraseña para ingresar al sistema, deberá dar click donde dice ¿Olvidó su Contraseña?, en ese momento aparece la ventana de la figura 74.

Esta es una captura de pantalla de un navegador web. El título de la ventana es "Recordar Contraseña - Microsoft Internet Explorer". El contenido principal de la página es un campo de texto etiquetado "Nombre de Usuario:" con un cursor visible. Debajo del campo hay un botón "Entrar" con un borde negro. En la parte inferior de la ventana, se puede ver la barra de estado del navegador con el texto "Listo" y "Intranet local".

Figura 74. Formulario para recordar contraseña.

En la cual se debe ingresar el nombre de usuario proporcionado en el formulario de registro y dar click en el botón Entrar, el sistema entonces enviará la contraseña correspondiente al e-mail del usuario.

Una vez que el usuario ingresa su nombre de usuario y contraseña, el sistema presenta la siguiente pantalla que muestra la figura 75.



Figura 75. Página Inicial del Sistema cuando se firma un Usuario.

La anterior es la pantalla de bienvenida donde se muestra el nombre completo del usuario, así como los privilegios del mismo. Como se puede observar, en la barra de menús se muestra un nuevo bloque llamado Inscripciones y Calificaciones, mismo que será descrito más adelante e este mismo documento. Dependiendo de los privilegios de cada usuario, la barra de menús puede mostrar más o menos opciones, aunque para los usuarios con privilegio “Alumno”, son siempre las que se muestran en la imagen anterior. En este punto, si se da click en el nombre completo del usuario o en la imagen del lápiz que se encuentra del lado izquierdo de dicho nombre, el usuario puede modificar parte de la información que le corresponde a la cuenta con la que entró al sistema.

En la figura 76 se muestra el formulario para cambiar dichos datos.



Dirección: Dirección

Telefono: 55555555

Email: myblackalex@hotmail.com

Msn: myblackalex@hotmail.com

Contraseña:

Confirmar Contraseña:

Editar Usuarios

Figura 76. Formulario para actualizar los datos del usuario.

Siguiendo dicho proceso, simplemente se hacen los cambios correspondientes en cada campo del formulario y se da click en el botón para aplicar dichos cambios en el sistema.

4.3.5. Salir.

Como su nombre lo indica, la función de salir, termina con la sesión actual del usuario.

4.4. Utilerías.

Este es el segundo menú del sistema y cabe mencionar que en el se encuentran herramientas y/o información necesaria por el usuario sin tener que estar registrado o autenticado en el mismo. Por el momento este menú contiene las opciones de “Prácticas” y “Apuntes”. En la figura 77, se muestra dicho menú.



Figura 77. Menú Utilerías.

4.4.1. Prácticas.

Cuando el usuario selecciona esta opción, se muestra en la figura 78, el siguiente formulario, en el cual se debe seleccionar la materia de la cual se desea abrir o descargar las prácticas existentes en ese momento.

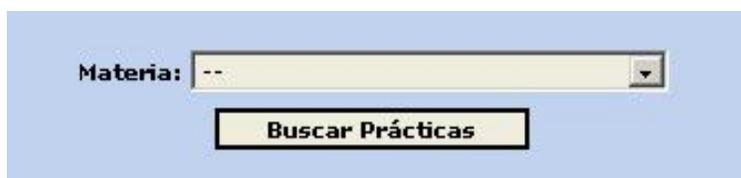
El formulario muestra un campo de selección etiquetado como 'Materia:' con el valor '--' y un botón 'Buscar Prácticas'.

Figura 78. Formulario para buscar Prácticas.

La imagen de la figura 79 muestra sólo una práctica para la materia seleccionada, para ver la información de dicha práctica, es necesario hacer click sobre la imagen del disquete o sobre el nombre del archivo, en ese momento aparece una ventana donde se pregunta si el usuario quiere abrir o guardar el archivo seleccionado. Si el usuario selecciona abrir, entonces se abre una nueva ventana con la información del archivo, en caso contrario, se abre otra ventana para especificar la ruta en la máquina de dicho usuario para guardar el archivo.

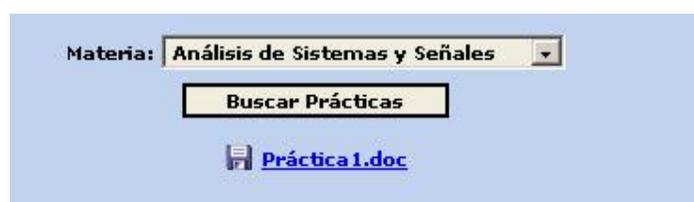
El formulario muestra el campo 'Materia:' con el valor 'Análisis de Sistemas y Señales' y el botón 'Buscar Prácticas'. Debajo del botón se muestra un ícono de disquete y el texto 'Práctica 1.doc'.

Figura 79. Listado de Prácticas de la materia seleccionada.

4.4.2. Apuntes.

Cuando el usuario selecciona esta opción, se muestra en la figura 80, el siguiente formulario, en el cual se debe seleccionar la materia de la cual se desea abrir o descargar los apuntes existentes en ese momento.

Un formulario web con un fondo azul claro. A la izquierda, el texto "Materia:" precede a un menú desplegable que muestra "--". Debajo del menú, hay un botón rectangular con el texto "Buscar Apuntes".

Figura 80. Formulario para buscar Apuntes.

La figura 81, muestra sólo un apunte para la materia seleccionada, para ver la información de dicho apunte, es necesario hacer click sobre la imagen del disquete o sobre el nombre del archivo, en ese momento aparece una ventana donde se pregunta si el usuario quiere abrir o guardar el archivo seleccionado. Si el usuario selecciona abrir, entonces se abre una nueva ventana con la información del archivo, en caso contrario, se abre otra ventana para especificar la ruta en la máquina de dicho usuario para guardar el archivo.

Un formulario web con un fondo azul claro. A la izquierda, el texto "Materia:" precede a un menú desplegable que muestra "Análisis de Sistemas y Señales". Debajo del menú, hay un botón rectangular con el texto "Buscar Apuntes". Debajo del botón, hay un ícono de un disquete y el texto "Apuntes.doc".

Figura 81. Listado de Apuntes de la materia seleccionada.

4.4.3. Consultar Sorteo.

Cuando el usuario selecciona esta opción, se muestra el formulario de la figura 82, en el cual se debe seleccionar el semestre y el número de cuenta del usuario que desea conocer el día y la hora de inscripción a laboratorios.

Un formulario web con un fondo azul claro. A la izquierda, el texto "Semestre:" precede a un menú desplegable que muestra "2006-II". Debajo del menú, el texto "Número de Cuenta:" precede a un campo de entrada de texto vacío. Debajo del campo de entrada, hay un botón rectangular con el texto "Consultar".

Figura 82. Formulario para consultar sorteo individual de alumnos.

Si el número de cuenta proporcionado se encuentra sorteado en el semestre especificado, el sistema muestra una leyenda como lo que se presenta en la figura 83.



Semestre: 2006-II ▾
Número de Cuenta: 094355707
Consultar

Tu inscripción a laboratorios es el día **09 de Feb de 2006** a las **09:00 hrs.**

Figura 83. Sorteo del alumno en el semestre seleccionado.

De lo contrario, el sistema muestra el mensaje de la figura 84.



Semestre: 2006-II ▾
Número de Cuenta: 094115196
Consultar

No fuiste sorteado para el semestre seleccionado.

Figura 84. Mensaje de error cuando no se encuentra al alumno.

4.5. Inscripciones y Calificaciones.

Es el tercer menú del sistema y es precisamente en este punto donde el usuario puede inscribirse a los diferentes grupos de laboratorio correspondientes a las materias inscritas en el correspondiente ciclo escolar.

Otra de las tareas que se llevan a cabo en este menú es, el de obtener el comprobante de inscripción y calificaciones obtenidas por el Alumno durante el curso.

4.5.1. Inscripciones.

Como se mencionó antes, es en esta opción donde los usuarios de tipo “Alumno” pueden llevar a cabo la inscripción a cada uno de los grupos de laboratorios que le corresponden a cada materia de teoría inscrita por dicho usuario, al seleccionar esta opción, el sistema muestra el formulario de la figura 85:

El formulario muestra un campo de texto con el texto "Semestre:" a la izquierda y un menú desplegable a la derecha que contiene "--" y un símbolo de flecha hacia abajo. Debajo de este campo hay un botón rectangular con el texto "Inscribir".

Figura 85. Formulario para seleccionar el semestre de inscripción.

Si el usuario, no tiene ninguna materia de teoría inscrita en el semestre especificado, el sistema envía el mensaje de la figura 86 y el proceso termina.

No tienes ninguna materia de teoría inscrita que requiera de laboratorio.

Figura 86. Mensaje de error al realizar la inscripción.

En caso contrario, el sistema muestra una lista de los grupos de laboratorios disponibles para las materias de teoría inscritas por el usuario que requieren de laboratorio.

Como se puede observar en la figura 87, el sistema muestra los grupos que tiene disponibles para las materias de teoría que el usuario tiene inscritas en el semestre en cuestión, se muestra también información relacionada con el grupo de laboratorio como es el nombre del maestro, horario, semestre, salón o ubicación, cupo, y lugares disponibles para que el usuario pueda tomar la decisión sobre cual grupo elegir. Si el grupo ya no tiene lugares disponibles para

inscribirse, el sistema muestra la leyenda SATURADO y desactiva la opción para inscribirse a dicho grupo de laboratorio.

Diseño Lógico							
Maestro	Días	Horario	Semestre	Ubicación	Cupo	Plazas Libres	Seleccionar Grupo
David Gonzalez Maxinez	Martes	09:00-11:00	2005-I	L3 Segundo Piso	10	10	<input type="radio"/>
Fortunato Cerecedo Hernández	Viernes	09:30-11:30	2005-I	L3 Segundo Piso	10	10	<input type="radio"/>

Análisis de Circuitos Eléctricos							
Maestro	Días	Horario	Semestre	Ubicación	Cupo	Plazas Libres	Seleccionar Grupo
Francisco Arista Patiño	Lunes	11:30-13:30	2005-I	CE1	0	SATURADO	<input type="radio"/>
Francisco Arista Patiño	Miercoles	11:30-13:30	2005-I	CE3	10	10	<input type="radio"/>

Inscribir

Figura 87. Listado de grupos de la materia a inscribir.

Una vez que el usuario ha decidido sobre el o los grupos a los cuales será inscrito, debe hacer click sobre la casilla correspondiente a cada grupo a inscribir y por último dar click en el botón inscribir y el proceso termina con el mensaje de la figura 88.

Se ha realizado con éxito la inscripción.

Figura 88. Mensaje del sistema al finalizar la inscripción.

4.5.2. Comprobante de Inscripción y Calificaciones.

Mediante esta opción el usuario puede obtener el comprobante de inscripción y de las calificaciones obtenidas durante el curso, para lo cual es necesario seleccionar el semestre del cual se desea obtener la información, como se muestra en la figura 89.

The image shows a screenshot of a web form. It features a light blue background. At the top, there is a label 'Semestre:' followed by a dropdown menu showing '--'. Below this, there is a rectangular button with a black border and the text 'Ver Calificaciones' in black.

Figura 89. Formulario para generar el comprobante de inscripción.

Una vez seleccionado el semestre, el sistema comienza a generar el comprobante correspondiente en formato PDF y posteriormente pedirá una contraseña para poder visualizarlo, el usuario debe escribir la contraseña que proporcionó en su registro para poder abrir el archivo.

Una vez que el usuario proporciona su contraseña, el documento es abierto para guardarlo o imprimirlo por dicho usuario.

Tipos de Usuario.

Existen dentro del sistema tres tipos de usuario en los que se tienen distintos privilegios que se describen a continuación.

- **Alumno.-** Cuenta con las operaciones básicas del sistema (Inicio, Alta en el Sistema, Acceso al Sistema, Salir, Prácticas, Apuntes), además de la funcionalidad para inscribirse a diversos grupos de laboratorio, así como también el obtener el respectivo comprobante de inscripción y calificaciones de dicha inscripción, todas ellas descritas anteriormente en este documento.
- **Maestro.-** Cuenta con las operaciones básicas a excepción de Alta en el Sistema, por lo tanto, el registro de este tipo de usuarios deberá hacerse por medio del usuario Administrador del Sistema. Adicionalmente cuenta con las operaciones de Calificar y una serie de reportes que pueden ser de utilidad para este tipo de usuarios. Estas últimas operaciones se describen en los siguientes apartados de este documento.

- **Administrador.-** Este tipo de usuario se encarga de administrar el sistema y cuenta con la misma funcionalidad que el usuario “Maestro”, además de que este tipo de usuario es capaz de realizar la inscripción a laboratorios de otros usuarios de tipo “Alumno” y un módulo de Administración del sistema. El registro de este tipo de usuarios debe hacerse por medio de un usuario “Administrador”. Este tipo de operaciones serán descritas más adelante en este mismo documento.

4.5.3. Calificar.

Como se comentó anteriormente, esta operación es utilizada para calificar los respectivos grupos de laboratorio. Para poder asentar la calificación a cada uno de los alumnos del grupo, es necesario seguir los siguientes pasos.

Especificar el semestre dentro del cual se esta trabajando como muestra la figura 90.

The image shows a screenshot of a web form. It features a label 'Semestre:' followed by a dropdown menu containing the text '2005-I'. Below this is a rectangular button with the text 'Ver Grupos' in a bold, black font.

Figura 90. Formulario para generar el listado de los grupos a calificar.

Una vez especificado el semestre, el sistema muestra los grupos que tiene asignado el maestro en cuestión y en caso de que el usuario que va a calificar sea administrador, el sistema muestra la lista de todos los grupos del semestre en cuestión.

La figura 91, muestra los grupos antes mencionados.

Semestre:

Materia	Horario	Ubicacion	Dia	Cupo	Semestre	Maestro
Análisis de Circuitos Eléctricos	11:30-13:30	CE1	Lunes	0	2005-I	Francisco Arista Patiño 
Análisis de Circuitos Eléctricos	11:30-13:30	CE3	Miercoles	10	2005-I	Francisco Arista Patiño 
Diseño Lógico	09:00-11:00	L3 Segundo Piso	Martes	10	2005-I	David Gonzalez Maxinez 
Diseño Lógico	09:30-11:30	L3 Segundo Piso	Viernes	10	2005-I	Fortunato Cerecedo Hernández 

Figura 91. Listado de grupos a calificar.

Para asignar las calificaciones a los alumnos del grupo se da click en el icono del lápiz que se encuentra en el lado derecho de cada uno de los grupos de la lista y el sistema entonces muestra la imagen de la figura 92.

Materia: Análisis de Circuitos Eléctricos	Horario: 11:30-13:30	Dia: Miercoles
Maestro: Francisco Arista Patiño	Semestre: 2005-I	Ubicacion: CE3
No Cuenta	Alumno	Calificación
094115196	Pedro Eduardo Rodríguez Alvarez	-- 
094324579	Alejandro Guadarrama Serrano	-- 

Figura 92. Listado de Alumnos del grupo a calificar.

Por último se selecciona la calificación de cada uno de los usuarios y se da click en el botón Calificar, cabe mencionar que se debe asignar forzosamente una calificación a cada alumno del grupo, en otro caso el sistema no asignará ninguna calificación, además de que la calificación establecida en este punto, no podrá ser modificada posteriormente.

4.6. Reportes.

El sistema ofrece la funcionalidad requerida para generar una serie de reportes que pueden ser de utilidad para el usuario con privilegios de “Maestro” ó “Administrador”; por lo tanto, lo que a continuación se describe aplica para ambos

tipos de usuario. Cabe mencionar que dichos reportes son generados en formato PDF y al momento de ser presentados en pantalla, el sistema requiere la contraseña del usuario que esta generando el reporte en cuestión, de lo contrario, no se genera el documento correspondiente.

Una vez que el usuario ha guardado el documento, siempre que el documento correspondiente sea abierto, éste seguirá pidiendo la contraseña del usuario que está intentando abrirlo. Todo lo anterior por razones de seguridad. Un usuario puede ver sólo la información que el sistema ha generado para él y algunas cuestiones de privilegios sobre los documentos PDF.

En la figura 93 se muestra la apariencia que tiene el menú de reportes.

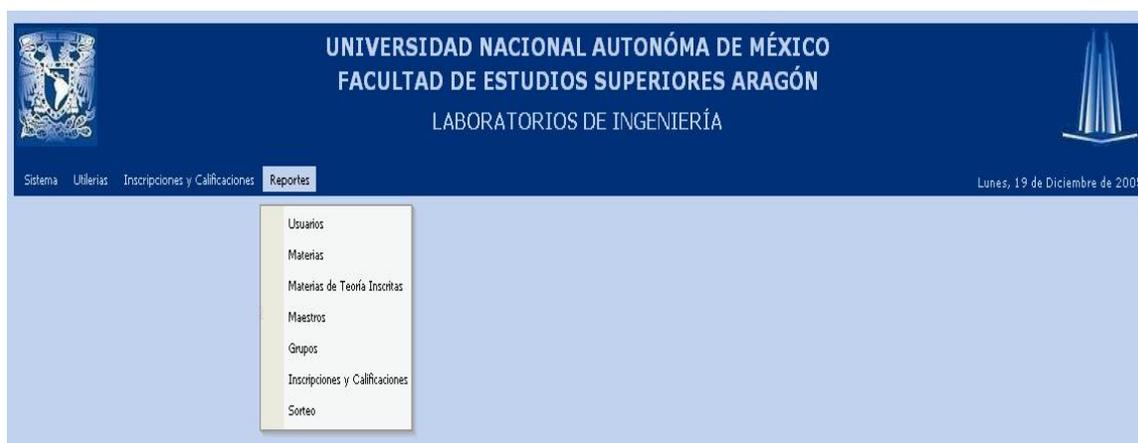


Figura 93. Menú Reportes.

4.6.1. Usuarios.

Mediante esta opción, el sistema genera un listado de los usuarios que se encuentran registrados dentro del mismo.

La figura 94 muestra el reporte de usuarios.

No.	Nombre	No. Cuenta	Telefono	Email	Usuario
2	Pedro Eduardo Rodríguez Alvarez	094115196	57635255	prz@servidor.unam.mx	prz
6	Nombre Paterno Materno	097543218	55555555	email@correo.com	user
7	Julio Cesar Valenzuela Maya	096778531	56575451	julio@correo.com	julio
8	Alejandro Guadarrama Serrano	094324579	55555555	myblackalex@hotmail.co	myblackalex
9	David González Maxinez		55555555	email@correo.com	david
10	Francisco Arista Patiño		55555555	email@correo.com	francisco

Figura 94. Reporte de usuarios.

4.6.2. Materias.

Mediante esta opción, el sistema genera un listado de las materias de las cuales se imparten los cursos de laboratorio. La figura 95 muestra el reporte de materias.

No.	Nombre	Clave
11	Análisis De Sistemas Y Señales	1400
12	Análisis De Circuitos Eléctricos	0024
13	Dispositivos Electrónicos	0138
14	Medición E Instrumentación	1715
15	Diseño Lógico	1720
16	Control Analógico	0112
17	Filtrado Y Modulación	1708
18	Comunicaciones Digtiales	0109

Figura 95. Reporte de materias.

4.6.3. Materias de Teoría Inscritas.

Con esta opción se puede obtener un listado de las materias de teoría que tienen inscritas todos los usuarios en el semestre que le sea especificado al sistema, o en su caso las materias de sólo un usuario en cuestión, para ello el sistema presenta primero el formulario de la figura 96 en caso de que se quiera consultar las materias inscritas de un alumno y, en caso de que se requiera la información de todos los usuarios, el formulario de la figura 97. Finalmente, en la figura 98 se muestra el reporte de materias de teoría inscritas.

Figura 96. Formulario para generar el reporte de materias inscritas por alumno.

Figura 97. Formulario para generar el reporte de materias inscritas de todos los alumnos.

No.	No. Cuenta	Nombre	Materia	Clave	Grupo
16	094115196	Pedro Eduardo Rodríguez Alvarez	Análisis De Circuitos Eléctricos	0024	1208
17	094115196	Pedro Eduardo Rodríguez Alvarez	Diseño Lógico	1720	1208
18	094324579	Alejandro Guadarrama Serrano	Diseño Lógico	1720	1208
19	094324579	Alejandro Guadarrama Serrano	Análisis De Circuitos Eléctricos	0024	1210

Figura 98. Reporte de materias inscritas.

4.6.4. Maestros.

Mediante este reporte se genera un listado con los datos de los maestros encargados de impartir los cursos de los laboratorios de ingeniería. En la figura 99 se muestra el reporte de maestros.

No.	Nombre	Clave
2	Francisco Arista Patiño	2
7	Fortunato Cerecedo Hernández	53
6	Ismael Díaz Rangel	52
1	Alberto González Ledesma	35
3	David González Maxinez	3
4	Pablo Luna Escorsa	50
5	Margarito Pineda Díaz	51

Figura 99. Reporte de maestros.

4.6.5. Grupos.

En este reporte se encuentra toda la información referente a los grupos de los diferentes laboratorios de ingeniería. En primera instancia hay que seleccionar el semestre del cual se quiere consultar la información, como se muestra en la figura 100. Posteriormente, en la figura 101 se muestra el reporte de grupos generado por el sistema.

Figura 100. Formulario para seleccionar el semestre a consultar.

No.	Materia	Maestro	Horario	Ubicación	Día	Cupo
1	Análisis De Circuitos Eléctricos	Francisco Arista Patiño	11:30-13:30	CE1	Lunes	0
2	Análisis De Circuitos Eléctricos	Francisco Arista Patiño	11:30-13:30	CE3	Miercoles	10
3	Diseño Lógico	David Gonzalez Maxinez	09:00-11:00	L3 Segundo Piso	Martes	10
4	Diseño Lógico	Fortunato Cerecedo Hernández	09:30-11:30	L3 Segundo Piso	Viernes	10

Figura 101. Reporte de grupos.

4.6.6. Inscripciones y Calificaciones.

Con este reporte se obtiene un listado de los alumnos inscritos en los laboratorios, así como la calificación obtenida por cada alumno durante el curso, es posible filtrar la información por Alumno, Grupo o Todo, como se muestra a continuación. En la figura 102 se muestra el formulario para generar el reporte de todos los alumnos; mientras que en la figura 103 se muestra para un alumno específico; en la figura 104 para un grupo y por último, en la figura 105 se muestra el reporte de inscripciones y calificaciones generado por el sistema.

Figura 102. Formulario para generar reporte de inscripciones y calificaciones de todos los alumnos.

Figura 103. Formulario para generar reporte de inscripciones y calificaciones por alumno.

Figura 104. Formulario para generar reporte de inscripciones y calificaciones por grupo.

No.	Materia	Día	Horario	nCuenta	Nombre	Calificación
15	Análisis De Circuitos Eléctricos	Miércoles	11:30-13:30	094115196	Pedro Eduardo Rodríguez Álvarez	10
16	Diseño Lógico	Martes	09:00-11:00	094115196	Pedro Eduardo Rodríguez Álvarez	
17	Diseño Lógico	Martes	09:00-11:00	094324579	Alejandro Guadarrama Serrano	
18	Análisis De Circuitos Eléctricos	Miércoles	11:30-13:30	094324579	Alejandro Guadarrama Serrano	10

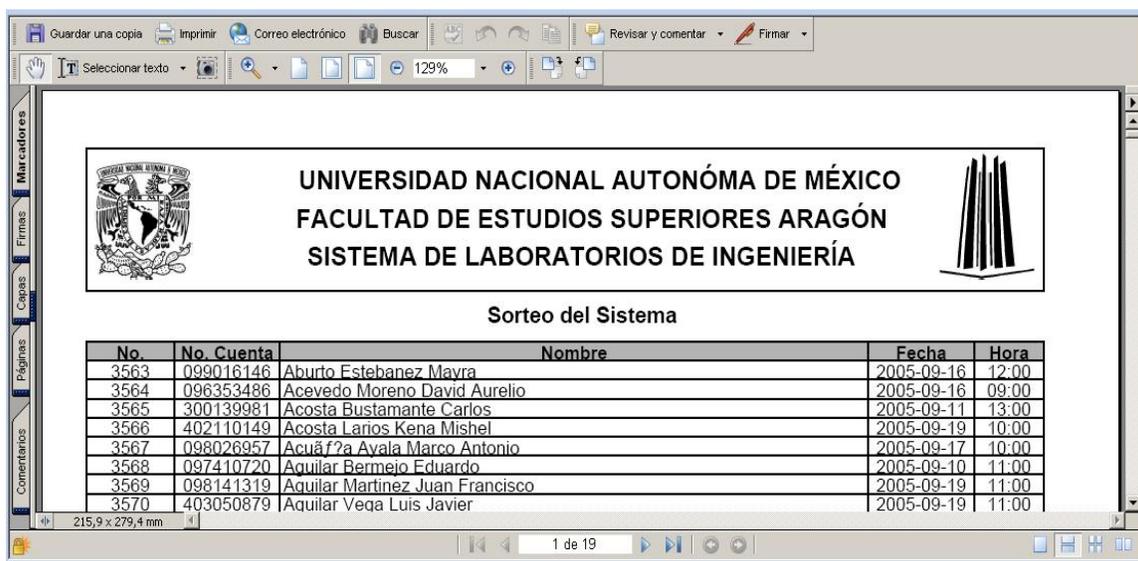
Figura 105. Reporte de inscripciones y calificaciones.

4.6.7. Sorteo.

Mediante este reporte, se puede consultar el sorteo de inscripciones para laboratorios de ingeniería, solo hay que especificar el semestre del cual se quiere obtener la información, como se muestra en la figura 106.

Figura 106. Formulario para generar el reporte del sorteo.

Finalmente, en la figura 107 se observa el reporte del sorteo generado por el sistema.



No.	No. Cuenta	Nombre	Fecha	Hora
3563	099016146	Aburto Estebanez Mayra	2005-09-16	12:00
3564	096353486	Acevedo Moreno David Aurelio	2005-09-16	09:00
3565	300139981	Acosta Bustamante Carlos	2005-09-11	13:00
3566	402110149	Acosta Larios Kena Mishel	2005-09-19	10:00
3567	098026957	Acuña?a Avala Marco Antonio	2005-09-17	10:00
3568	097410720	Aguilar Bermejo Eduardo	2005-09-10	11:00
3569	098141319	Aguilar Martínez Juan Francisco	2005-09-19	11:00
3570	403050879	Aguilar Veaa Luis Javier	2005-09-19	11:00

Figura 107. Reporte de sorteo del sistema.

4.7. Administración.

En esta última parte del presente documento, se describe la parte de administración del sistema, para lo cual es necesario que se tengan privilegios de Administrador, se recomienda que dicho usuario sea el encargado del sistema y la administración de los laboratorios de ingeniería, aunque es posible tener varias cuentas con dichos privilegios, según se requiera la operación del departamento.

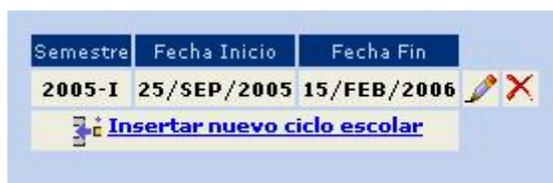
A continuación se da una breve descripción de las partes que se pueden administrar desde este módulo del sistema.

- **Administración de Catálogos.-** Altas, Bajas, Cambios de la información contenida en los catálogos de Ciclo Escolar, Usuarios, Maestros, Materias y Grupos.
- **Administración de Utilerías.-** Agregar o quitar documentos correspondientes a cada una de las prácticas y/o apuntes de cada una de las materias de los grupos de laboratorios.

- **Carga del Sorteo de Inscripción.-** Realiza la carga al sistema del sorteo de inscripción a laboratorios de ingeniería.
- **Carga de Materias de Teoría Inscritas.-** Realiza la carga al sistema de las materias de teoría que tiene inscritas cada uno de los alumnos registrados en la aplicación.
- **Verificar listado de Materias Inscritas.-** Realiza una verificación sobre las materias de teoría inscritas por los alumnos.
- **Cambiar Fecha/Hora de Inscripción.- Modifica** la fecha y/o hora de inscripción a laboratorios de los alumnos interesados.

4.7.1. Catálogo Ciclo Escolar.

En este punto se administran los datos del ciclo escolar sobre el cual se va a trabajar. Es de gran importancia ya que a partir de dichos datos, trabajan varios módulos del sistema, inicialmente, se muestra un listado de los ciclos escolares que tenga registrados el sistema, como se muestra en la figura 108.



Semestre	Fecha Inicio	Fecha Fin
2005-I	25/SEP/2005	15/FEB/2006

[Insertar nuevo ciclo escolar](#)

Figura 108. Listado de ciclo escolar.

Alta.

Para agregar un nuevo ciclo escolar, es necesario dar click sobre la liga “Insertar nuevo ciclo escolar” y el sistema muestra el formulario de la figura 109.

En éste se debe especificar el nombre del ciclo escolar (semestre), las fechas de inicio y término del ciclo escolar, la semana de cierre del semestre escolar y las horas y días en los cuales se realizará la inscripción a laboratorios de ingeniería,

esto último para llevar el control junto con el sorteo sobre el día y la hora en que se puede inscribir cada alumno.

Figura 109. Formulario de alta de ciclo escolar.

Baja.

Para eliminar un ciclo escolar, simplemente se da click en el icono de la tacha roja y se muestra el formulario de la figura 110.

Sólo para confirmar que se quiere eliminar este registro del catálogo de ciclo escolar.

El siguiente ciclo escolar será borrado de la base de datos

Semestre:	2005-I	
Fecha Inicio Ciclo Escolar:	25/SEP/2005	
Fecha Terminó Ciclo Escolar:	15/FEB/2006	
Semana de Cierre:	01/FEB/2006 - 15/FEB/2005	
Primer Hora Inscripción:	09:00	Segunda Hora Inscripción: 10:00
Tercer Hora Inscripción:	11:00	Cuarta Hora Inscripción: 12:00
Quinta Hora Inscripción:	13:00	Sexta Hora Inscripción: 15:00
Septima Hora Inscripción:	16:00	Octava Hora Inscripción: 17:00
Novena Hora Inscripción:	18:00	Decima Hora Inscripción: 19:00
Primer Día Inscripción:	10/SEP/2005	
Segundo Día Inscripción:	11/SEP/2005	
Tercer Día Inscripción:	12/SEP/2005	
Cuarto Día Inscripción:	13/SEP/2005	
Quinto Día Inscripción:	14/SEP/2005	
Sexto Día Inscripción:	15/SEP/2005	
Septimo Día Inscripción:	16/SEP/2005	
Octavo Día Inscripción:	17/SEP/2005	
Noveno Día Inscripción:	18/SEP/2005	
Decimo Día Inscripción:	19/SEP/2005	
Último día de Altas y Bajas:	25/FEB/2006	

Borrar Ciclo Escolar

Figura 110. Formulario de baja de ciclo escolar.

Editar.

Para realizar cambios sobre algún registro del catálogo, es necesario dar click en el icono del lápiz del lado derecho del registro que nos lleva al formulario de alta, para modificar las fechas de la semana del cierre, fechas y horas de inscripción.

4.7.2. Catálogo Usuarios.

En este punto se administran los datos de usuarios como los de la figura 111.

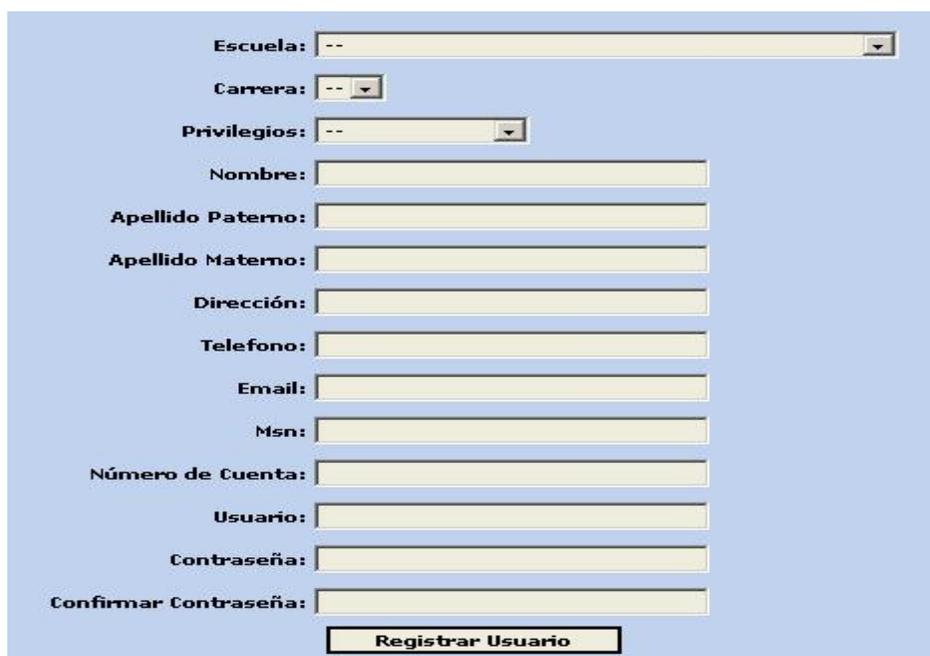
Buscar y Editar Usuario		Buscar y Borrar Usuario			
Nombre	Apellido Paterno	Apellido Materno	Número de Cuenta	Usuario	Privilegios
Pedro Eduardo	Rodríguez	Alvarez	094115196	prz	Administrador
nombre	paterno	materno	097543218	user	Maestro
Julio Cesar	Valenzuela	Maya	096778531	julio	Alumno
Alejandro	Guadarrama	Serrano	094324579	myblackalex	Alumno
David	González	Maxinez		david	Maestro
Francisco	Arista	Patño		francisco	Maestro

[Insertar nuevo usuario](#)

Figura 111. Listado de usuarios.

Alta.

Para agregar un nuevo usuario, es necesario dar click sobre la liga “Insertar nuevo usuario” y el sistema muestra el formulario de la figura 112.



El formulario de registro de usuarios se muestra en un recuadro azul claro. Incluye los siguientes campos:

- Escuela: menú desplegable con "--" y una flecha hacia abajo.
- Carrera: menú desplegable con "--" y una flecha hacia abajo.
- Privilegios: menú desplegable con "--" y una flecha hacia abajo.
- Nombre: campo de texto.
- Apellido Paterno: campo de texto.
- Apellido Materno: campo de texto.
- Dirección: campo de texto.
- Telefono: campo de texto.
- Email: campo de texto.
- Msn: campo de texto.
- Número de Cuenta: campo de texto.
- Usuario: campo de texto.
- Contraseña: campo de texto.
- Confirmar Contraseña: campo de texto.

Debajo de los campos hay un botón rectangular con el texto "Registrar Usuario".

Figura 112. Formulario de registro de usuarios.

Cabe mencionar que es el mismo formulario que se presenta en la opción “Alta en el Sistema” y sigue las mismas reglas en cuanto a la validación y funcionamiento. El usuario administrador, puede registrar usuarios de cualquier tipo (Alumno, Maestro y Administrador) desde esta misma opción.

Baja.

Para eliminar un alumno, simplemente se da click en el icono de la tacha roja y se muestra el formulario de la figura 113.

Sólo para confirmar que se quiere eliminar este registro del catálogo de usuarios.

Se va a eliminar un usuario con los siguientes datos:

Nombre:	Francisco
Apellido Paterno:	Arista
Apellido Materno:	Patiño
Número de Cuenta:	
Escuela:	Facultad de Estudios Profesionales Aragón
Carrera:	Ingeniería en Computación
Dirección:	Direccion
Telefono:	55555555
Email:	email@correo.com
Msn:	msn@correo.com
Usuario:	francisco
Privilegios:	Maestro

Borrar Usuario

Figura 113. Formulario de baja de usuarios.

Editar.

Para realizar cambios sobre algún registro del catálogo, es necesario dar click en el icono del lápiz del lado derecho del registro y nos lleva al formulario de edición que se muestra en la figura 114.

Dirección:	direccion
Telefono:	55555555
Email:	email@correo.com
Msn:	msn@correo.com
Contraseña:	
Confirmar Contraseña:	
Privilegios:	Maestro

Editar Usuarios

Figura 114. Formulario de edición de usuarios.

Buscar y Editar Usuarios.

El catálogo de usuarios contendrá muchos usuarios y se vuelve muy difícil el buscar un usuario dentro del listado original, por tal motivo, existe una liga arriba de dicho listado que dice “Buscar y Editar Usuarios”, al dar click sobre esta liga, se muestra el formulario de la figura 115 donde se busca el username del usuario en cuestión y el sistema muestra entonces el formulario para editar los datos del usuario.

El formulario muestra un campo de texto con el label 'Usuario:' y el valor 'User' dentro de un cuadro de selección. Debajo del campo hay un botón rectangular con el texto 'Buscar'.

Figura 115. Formulario para buscar usuarios a editar.

Buscar y Borrar Usuarios.

El catálogo de usuarios contendrá muchos usuarios y se vuelve muy difícil el buscar un usuario dentro del listado original, por tal motivo, existe una liga arriba de dicho listado que dice “Buscar y Borrar Usuarios”, al dar click sobre esta liga, se muestra el formulario de la figura 116, donde se busca el username del usuario en cuestión y el sistema muestra entonces el formulario para borrar el usuario.

El formulario muestra un campo de texto con el label 'Usuario:' y el valor 'user' dentro de un cuadro de selección. Debajo del campo hay un botón rectangular con el texto 'Buscar'.

Figura 116. Formulario para buscar usuarios a borrar.

4.7.3. Catálogo Maestros.

En este punto se administran los datos de los maestros registrados en el sistema. El sistema muestra un listado como el de la figura 117.

Nombre	Apellido Paterno	Apellido Materno	Clave		
Alberto	González	Ledesma	35		
Francisco	Arista	Patino	2		
David	Gonzalez	Maxinez	3		
Pablo	Luna	Escorsa	50		
Margarito	Pineda	Díaz	51		
Ismael	Díaz	Rangel	52		
Fortunato	Cerecedo	Hernández	53		
 Insertar nuevo maestro					

Figura 117. Listado de maestros.

Alta.

Para agregar un nuevo maestro, es necesario dar click sobre la liga “Insertar nuevo maestro” y el sistema muestra el formulario de la figura 118.

Nombre:

Apellido Paterno:

Apellido Materno:

Clave:

Figura 118. Formulario de alta de maestros.

Baja.

Para eliminar un maestro, simplemente se da click en el icono de la tacha roja y se muestra el formulario de la figura 119.

El siguiente maestro será borrado de la base de datos

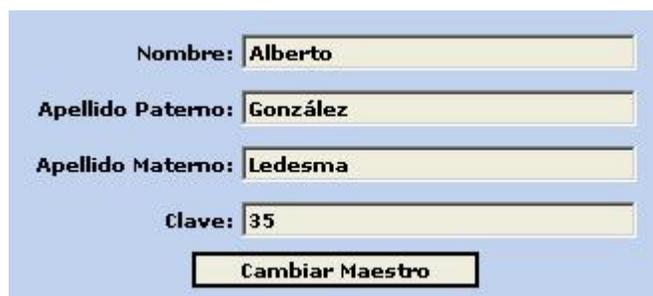
Nombre:	Alberto
Apellido Paterno:	González
Apellido Materno:	Ledesma
Clave:	35

Figura 119. Formulario de baja de maestros.

Solo para confirma que se quiere eliminar este registro del catálogo de maestros.

Editar.

Para realizar cambios sobre algún registro del catálogo, es necesario dar click en el icono del lápiz del lado derecho del registro que nos lleva al formulario de edición que se muestra en la figura 120.



Formulario de edición de maestros. El formulario contiene los siguientes campos de texto:

- Nombre: Alberto
- Apellido Paterno: González
- Apellido Materno: Ledesma
- Clave: 35

Debajo de los campos hay un botón que dice "Cambiar Maestro".

Figura 120. Formulario de edición de maestros.

4.7.4. Catálogo Materias.

En este punto se administran los datos de las materias registradas en el sistema. El sistema muestra un listado como el de la figura 121.



Nombre	Clave		
Análisis de Sistemas y Señales	1400		
Análisis de Circuitos Eléctricos	0024		
Dispositivos Electrónicos	0138		
Medición e Instrumentación	1715		
Diseño Lógico	1720		
Control Analógico	0112		
Filtrado y Modulación	1708		
Comunicaciones Digitales	0109		
Control Digital	0114		
Insertar nueva materia			

Figura 121. Listado de materias.

Alta.

Para agregar una nueva materia, es necesario dar click sobre la liga “Insertar nueva materia” y el sistema muestra el formulario de la figura 122. En el campo clave, se debe incluir la clave de la materia y en caso de que la materia tenga diferentes claves por pertenecer a más de una carrera, se debe escribir todas las claves de dicha materia separadas por coma.

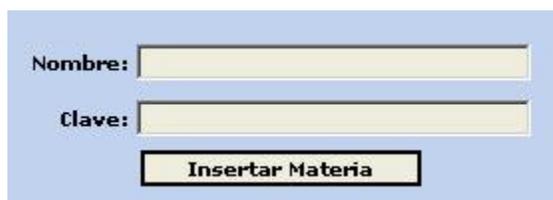
El formulario de alta de materias se muestra en un recuadro azul claro. Contiene dos campos de texto: "Nombre:" y "Clave:". Debajo de estos campos hay un botón rectangular con el texto "Insertar Materia".

Figura 122. Formulario de alta de materias.

Baja.

Para eliminar una materia, simplemente se da click en el icono de la tacha roja y se muestra el formulario de la figura 123. Sólo para confirmar que se quiere eliminar este registro del catálogo de materias.

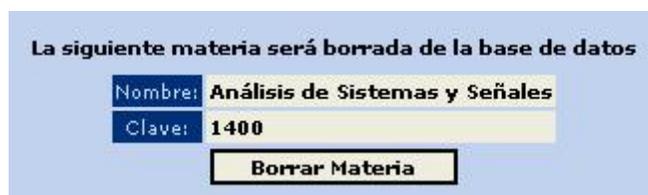
El formulario de baja de materias se muestra en un recuadro azul claro. Al inicio, hay un mensaje: "La siguiente materia será borrada de la base de datos". A continuación, se muestran dos campos de texto: "Nombre:" con el valor "Análisis de Sistemas y Señales" y "Clave:" con el valor "1400". Debajo de estos campos hay un botón rectangular con el texto "Borrar Materia".

Figura 123. Formulario de baja de materias.

Editar.

Para realizar cambios sobre algún registro del catálogo, es necesario dar click en el icono del lápiz del lado derecho del registro que nos lleva al formulario de edición que se muestra en la figura 124.

Nombre:

Clave:

Figura 124. Formulario de edición de materias.

4.7.5. Catálogo Grupos.

En este punto se administran los datos de los grupos de laboratorios registrados en el sistema.

Semestre:

Materia:

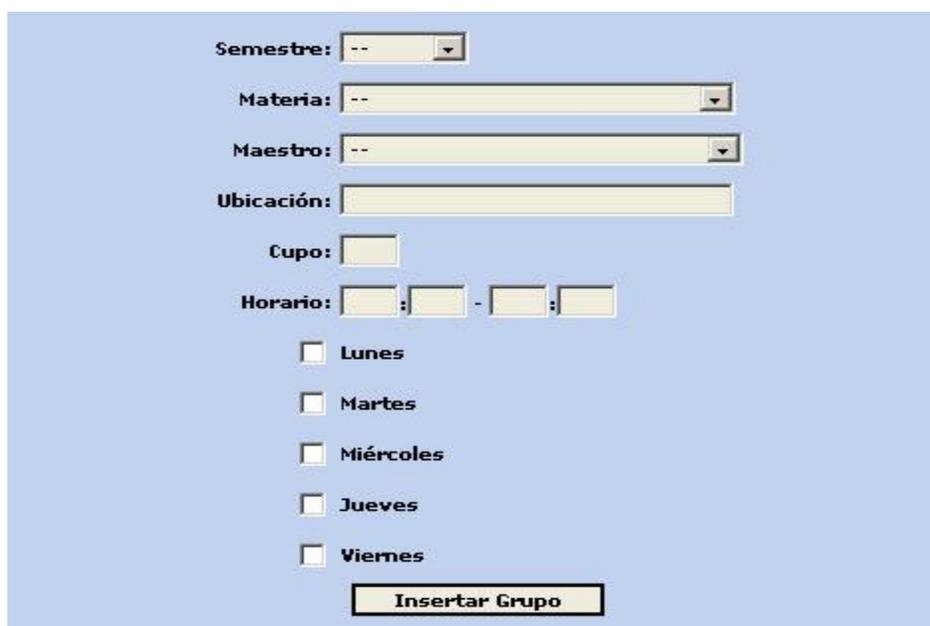
Materia	Horario	Maestro	Ubicacion	Dia	Cupo	Semestre	
Diseño Lógico	10:00- 12:00	Ing. Rafael Marquez Ramírez	L3 Segundo Piso	Lunes	12	2006-II	 
Diseño Lógico	13:30- 15:30	Ing. Esteban Arellano Rivera	L3 Segundo Piso	Lunes	12	2006-II	 
Diseño Lógico	15:30- 17:30	Oswaldo Ortíz Jiménez	L3 Segundo Piso	Lunes	12	2006-II	 

Figura 125. Listado de grupos.

Para ello se tiene que especificar el ciclo escolar sobre el cual se va a trabajar con los grupos y la materia en cuestión, entonces el sistema muestra un listado como el de la figura 125.

Alta.

Para agregar un nuevo grupo, es necesario dar click sobre la liga “Insertar nuevo grupo” y el sistema muestra el formulario de la figura 126.



Formulario de alta de grupos. El formulario está sobre un fondo azul claro y contiene los siguientes campos:

- Semestre: -- (menú desplegable)
- Materia: -- (menú desplegable)
- Maestro: -- (menú desplegable)
- Ubicación: (campo de texto)
- Cupo: (campo de texto)
- Horario: [] : [] - [] : [] (campos de hora)
- Una lista de días con casillas de verificación:
 - Lunes
 - Martes
 - Miércoles
 - Jueves
 - Viernes
- Botón "Insertar Grupo" (rectangular, con borde negro)

Figura 126. Formulario de alta de grupos.

Baja.

Para eliminar un grupo, simplemente se da click en el icono de la tacha roja y se muestra el formulario de la figura 127.

Sólo para confirmar que se quiere eliminar este registro del catálogo de grupos.



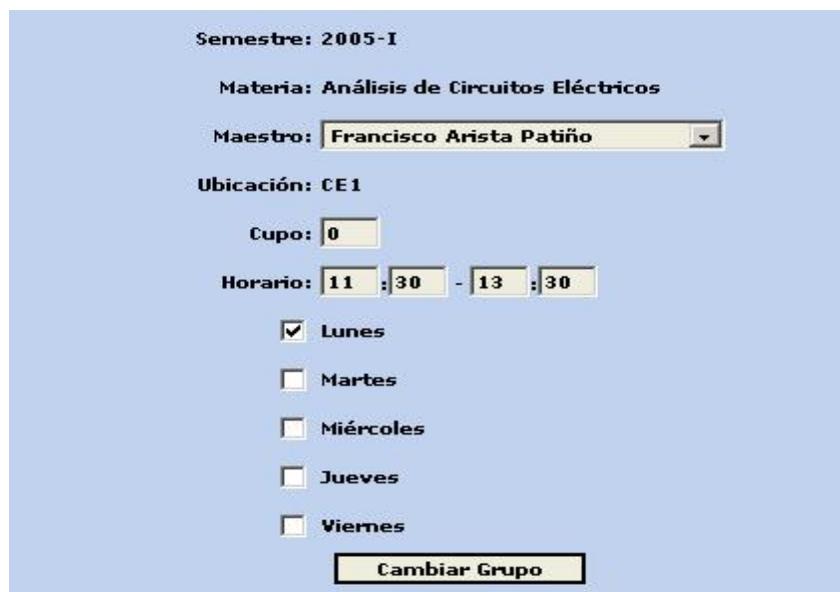
Formulario de baja de grupos. El formulario está sobre un fondo azul claro y contiene:

- Título: "El siguiente grupo será borrado de la base de datos"
- Campos de información del grupo:
 - Semestre: 2005-I
 - Materia: Análisis de Circuitos Eléctricos
 - Maestro: Francisco Arista Patiño
 - Ubicación: CE1
 - Cupo: 0
 - Horario: 11:30-13:30
 - Días: Lunes
- Botón "Borrar Grupo" (rectangular, con borde negro)

Figura 127. Formulario de baja de grupos.

Editar.

Para realizar cambios sobre algún registro del catálogo, es necesario dar click en el icono del lápiz del lado derecho del registro que nos lleva al formulario de edición que se muestra en la figura 128.



Semestre: 2005-I

Materia: Análisis de Circuitos Eléctricos

Maestro: Francisco Arista Patiño

Ubicación: CE1

Cupo: 0

Horario: 11 : 30 - 13 : 30

Lunes

Martes

Miércoles

Jueves

Viernes

Cambiar Grupo

Figura 128. Formulario de edición de grupos.

Copiar Grupos.

Dentro de la administración de laboratorios en el sistema, existe una opción para copiar uno o más grupos de un semestre a otro, ya que normalmente, los cursos mantienen los horarios y maestros sin cambio alguno y sólo cambian en dichos aspectos, por eventualidades que pocas veces se presentan. Para llevar a cabo esta tarea, en el listado de grupos hay que dar click en el vínculo, “Copiar grupos a otro semestre” y el sistema muestra un formulario como el de la figura 129.

Semestre Origen: 2006-I

Semestre Destino: 2006-II

Materia: Diseño Lógico

<input type="checkbox"/> Todo	Materia	Horario	Maestro	Ubicacion	Día	Cupo
<input type="checkbox"/>	Diseño Lógico	10:00-12:00	Ing. Rafael Marquez Ramírez	L3 Segundo Piso	Lunes	12
<input type="checkbox"/>	Diseño Lógico	13:30-15:30	Ing. Esteban Arellano Rivera	L3 Segundo Piso	Lunes	12
<input type="checkbox"/>	Diseño Lógico	15:30-17:30	Oswaldo Ortíz Jiménez	L3 Segundo Piso	Lunes	12
<input type="checkbox"/>	Diseño Lógico	17:00-19:00	Ing. Alejandro Villa fuerte .	L3 Segundo Piso	Lunes	12
<input type="checkbox"/>	Diseño Lógico	09:00-11:00	Ismael Díaz Rangel	L3 Segundo Piso	Martes	12
<input type="checkbox"/>	Diseño Lógico	11:00-13:00	Ing. Fortunato Cerecedo Hernández	L3 Segundo Piso	Martes	12
<input type="checkbox"/>	Diseño Lógico	13:00-15:00	Ing. Javier Alain Morones Camacho	L3 Segundo Piso	Martes	12
<input type="checkbox"/>	Diseño Lógico	09:30-11:30	Ing. Esteban Arellano Rivera	L3 Segundo Piso	Miercoles	12
<input type="checkbox"/>	Diseño Lógico	11:30-13:30	Ing. Benito Zuñiga Villegas	L3 Segundo Piso	Miercoles	12
<input type="checkbox"/>	Diseño Lógico	13:30-15:30	Ing. Benito Zuñiga Villegas	L3 Segundo Piso	Miercoles	12
<input type="checkbox"/>	Diseño Lógico	15:30-17:30	Ing. Benito Zuñiga Villegas	L3 Segundo Piso	Miercoles	12
<input type="checkbox"/>	Diseño Lógico	09:00-11:00	Ismael Díaz Rangel	L3 Segundo Piso	Jueves	12
<input type="checkbox"/>	Diseño Lógico	11:00-13:00	Ing. Fortunato Cerecedo Hernández	L3 Segundo Piso	Jueves	12
<input type="checkbox"/>	Diseño Lógico	13:30-15:30	Ing. Ricardo López Hernández	L3 Segundo Piso	Jueves	12
<input type="checkbox"/>	Diseño Lógico	09:30-11:30	Ing. Fortunato Cerecedo Hernández	L3 Segundo Piso	Viernes	12
<input type="checkbox"/>	Diseño Lógico	11:30-13:30	Ismael Díaz Rangel	L3 Segundo Piso	Viernes	12
<input type="checkbox"/>	Diseño Lógico	13:30-15:30	Raymundo Palacios Vazquez	L3 Segundo Piso	Viernes	12
<input type="checkbox"/>	Diseño Lógico	15:00-17:30	Ing. Martín Hernández .	L3 Segundo Piso	Viernes	12

Copiar

Figura 129. Listado de grupos a copiar.

Para generar el listado anterior, se debe seleccionar el semestre de origen y el semestre destino para realizar la copia de los grupos entre ellos, cabe señalar que ambos semestres deben ser diferentes entre sí para evitar duplicidad de los mismos en el sistema. Además de los semestres, se debe seleccionar también la materia de los grupos que se desea copiar. Hasta este punto, el sistema genera el listado y para realizar la copia, se debe seleccionar cada uno de los grupos que serán copiados, dando click en el recuadro que se encuentra del lado izquierdo referente a cada grupo y por último dar click en el botón copiar, puede verse que

en el lado izquierdo del encabezado de la tabla, hay otro recuadro con la leyenda “Todo”, dicho control, sirve para seleccionar todos los grupos listados en la tabla, en caso de así requerirse, y evitar con ello el dar click uno a uno.

4.7.6. Prácticas.

En este punto se administran las prácticas comprendidas en cada uno de los laboratorios de ingeniería, se muestra el formulario de la figura 130.



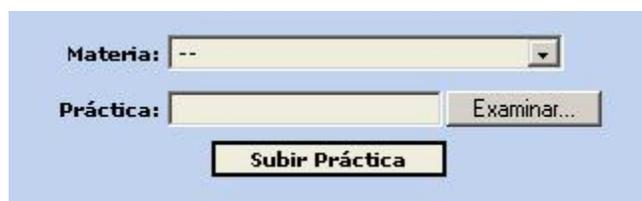
Materia	Práctica
Análisis de Sistemas y Señales	Práctica 1.doc 

 [Insertar nueva práctica](#)

Figura 130. Listado de prácticas.

Alta.

Para agregar una nueva práctica es necesario dar click sobre la liga “Insertar nueva práctica” y el sistema muestra el formulario de la figura 131.



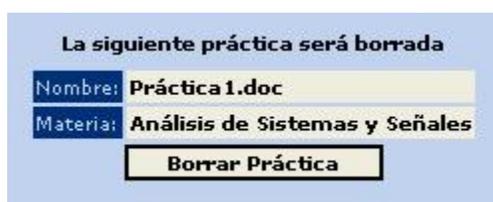
Materia:

Práctica:

Figura 131. Formulario para publicar prácticas.

Baja.

Para eliminar una práctica, simplemente se da click en el icono de la tacha roja y se muestra el formulario de la figura 132. Sólo para confirmar que se quiere eliminar la práctica en cuestión.



La siguiente práctica será borrada

Nombre:	Práctica 1.doc
Materia:	Análisis de Sistemas y Señales

Figura 132. Formulario para borrar prácticas.

4.7.7. Apuntes.

En este punto se administran los apuntes comprendidos en cada uno de los laboratorios de ingeniería, se muestra el formulario de la figura 133.



Materia	Apunte
Análisis de Sistemas y Señales	Apuntes.doc <input type="button" value="X"/>

Figura 133. Listado de apuntes.

Alta.

Para agregar un nuevo apunte es necesario dar click sobre la liga “Insertar nuevo apunte” y el sistema muestra el formulario de la figura 134.



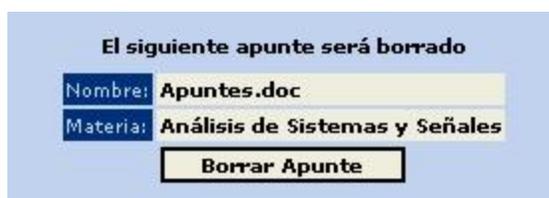
Materia:

Apuntes:

Figura 134. Formulario de alta de apuntes.

Baja.

Para eliminar un apunte, simplemente se da click en el icono de la tacha roja y se muestra el formulario de la figura 135.



El siguiente apunte será borrado

Nombre: Apuntes.doc

Materia: Análisis de Sistemas y Señales

Borrar Apunte

Figura 135. Formulario de baja de apuntes.

Sólo para confirmar que se quiere eliminar el apunte en cuestión.

4.7.8. Carga de Sorteo de Inscripción.

Mediante esta opción, el sistema realiza la carga del listado del sorteo enviado por servicios escolares, simplemente hay que seleccionar el semestre en cuestión, así como el archivo que contiene el listado, este archivo debe ser en texto plano y separado por comas, con el siguiente y estricto orden. En la figura 136 se muestra el formulario para realizar la carga de dicho sorteo.



Semestre: --

Archivo: Examinar...

Guardar

Figura 136. Formulario para realizar la carga del sorteo de inscripción.

- Contener como encabezado los siguientes datos Nombre, Cuenta, Día, Plantel, Carrera, Plan, Mdinsc, Hora Inscripción, Última Inscripción, Inscribió, en el mismo orden.
- Comenzar con los datos a cargar a partir del segundo renglón.

4.7.9. Carga de Materias de Teoría Inscritas.

Mediante esta opción, el sistema realiza la carga del listado de las materias de teoría inscritas enviado por servicios escolares, simplemente hay que seleccionar el semestre en cuestión, así como el archivo que contiene el listado, este archivo

debe ser en texto plano y separado por comas, con el siguiente y estricto orden. En la figura 137 se muestra el formulario para realizar la carga de dicho listado.

El formulario tiene un fondo azul claro. En la parte superior izquierda, hay un campo de texto etiquetado 'Semestre:' con un menú desplegable que muestra '--'. Debajo de esto, hay un campo de texto etiquetado 'Archivo:' con un botón 'Examinar...' a su derecha. En la parte inferior del formulario, hay un botón 'Guardar'.

Figura 137. Formulario para realizar la carga del listado de materias de teoría inscritas.

- Contener como encabezado los siguientes datos Nombre, Cuenta, Clave, Grupo, en el mismo orden.
- Comenzar con los datos a cargar a partir del segundo renglón.

4.7.10. Validación de listado Materias de Teoría Inscritas.

Mediante esta opción, el sistema realiza la validación del listado de materias inscritas, para evitar que se ocupen lugares en los grupos de laboratorio sin sustento alguno. En la figura 138 se muestra el formulario para realizar dicha validación.

Este formulario es idéntico al de la Figura 137, con un fondo azul claro, un campo 'Semestre:' con un menú desplegable '--', un campo 'Archivo:' con un botón 'Examinar...', y un botón 'Guardar'.

Figura 138. Formulario para realizar la validación del listado de materias de teoría inscritas.

4.7.11. Cambiar Fecha/Hora de Inscripción de Alumnos.

Mediante esta opción, el administrador del sistema puede cambiar la fecha y/o la hora de inscripción de los alumnos, siempre bajo el criterio de dicho administrador. En la figura 139 se muestra el formulario para realizar dicho proceso.

El formulario se muestra sobre un fondo azul claro. En la parte superior, hay un campo de lista desplegable etiquetado como "Semestre:" con el contenido "--" y un icono de flecha hacia abajo. Debajo de esto, hay un campo de texto etiquetado como "Número de Cuenta:". En la parte inferior del formulario, hay un botón rectangular con el texto "Guardar" en un recuadro negro.

Figura 139. Formulario para buscar el sorteo del alumno en el semestre seleccionado.

Una vez que el administrador proporciona los datos en el formulario anterior, el sistema muestra una pantalla para ingresar el día y la hora de inscripción del alumno.

4.8. Pie de página.

El sistema cuenta con un pie de página con las siguientes opciones que son accesibles sin que el usuario tenga que estar autenticado en la aplicación.

- **Contacto.-** Con esta opción se muestra una ventana con un formulario para que cualquier persona se ponga en contacto con el administrador del sistema. Simplemente hay que llenar los campos que se piden y al dar click en el botón, el sistema envía un correo electrónico a dicho administrador.
- **Créditos.-** Esta opción muestra la página donde se encuentran los créditos del sistema.
- **Ayuda.-** En esta opción se abre una ventana donde se encuentra la documentación necesaria para poder operar el sistema.
- **FAQ.-** Por último en esta sección se presentan distintas cuestiones con respectiva respuesta referentes a la forma de operar el sistema.

Hasta este punto, se ha detallado toda la funcionalidad que existe dentro del sistema, así como la forma de instalar, operar y administrar la aplicación. Se recomienda mantener actualizada la documentación correspondiente al manual de usuario y administrador, en caso de que se realicen cambios en alguno de los

módulos que componen a la aplicación y poder tener una referencia adecuada de dicha información en el momento en que sea necesario.

Recomendaciones

Recomendaciones

Como se pudo observar, tanto en el análisis y diseño, así como en el manual de usuario y administrador descritos en páginas anteriores, cualquier persona registrada en el sistema es capaz de generar una serie de listados que van desde el comprobante de inscripción y calificaciones obtenidas, hasta el listado del sorteo y los reportes del sistema, dependiendo de los privilegios que el operador tenga en su cuenta de acceso para poder llevar a cabo estas tareas. Debido a esta característica del sistema, se recomienda para que cualquiera de estos documentos tenga validez oficial, que el administrador del sistema o en su caso, el maestro encargado del grupo en cuestión, aplique un sello en el listado impreso correspondiente. Un ejemplo claro de tal necesidad se presenta cuando un alumno ha aprobado ya el laboratorio de una asignatura, pero por alguna razón no pudo aprobar la materia de teoría correspondiente. En este caso y por la forma en que opera la UNAM, el alumno puede volver a cursar una y solo una vez más la materia de teoría que no ha sido aprobada. Dado lo anterior, en ocasiones la calificación aprobatoria en las materias de teoría esta condicionada al haber aprobado el laboratorio que le corresponde. Debido a que la inscripción a laboratorios de ingeniería a través del sistema en cuestión esta condicionada a la materia de teoría inscrita en el semestre correspondiente, el alumno también puede inscribirse una y solo una vez más a dicho laboratorio, pero puede optar por revalidar la calificación obtenida en tal laboratorio en semestres anteriores. Se puede observar claramente que es necesario que la persona encargada de aplicar el sello correspondiente debiera consultar en el sistema la información pertinente para poder aplicar tal característica de validez oficial. Se deberá seguir el mismo procedimiento para los alumnos que ya no tienen posibilidad alguna de cursar la materia de teoría y optan por lo tanto en la realización de un examen extraordinario para aprobar la asignatura, para ello, dicho alumno debiera tener acreditado el correspondiente laboratorio en semestres anteriores y debiera solicitar una revalidación con el administrador del sistema, debido a que el sistema no permite la inscripción a un alumno que no ha sido inscrito en la materia de teoría en el semestre en curso.

Recomendaciones

Una de las características importantes dentro del manejo de la información, es la delegación de responsabilidades en una sola persona que se encargue de administrar el equipo en general y/o en su caso solo de las aplicaciones que se encuentran ejecutando en la computadora. En este caso, se recomienda que exista una persona encargada de la administración del sistema aunque esto no impide que se tengan varias cuentas con privilegios de Administrador en cuanto al funcionamiento del sistema se refiere.

Otro de los aspectos a considerar, es la distribución del espacio en disco duro que tendrá el servidor de la aplicación; es decir, en el caso del sistema operativo Windows, se podrán tener las particiones en disco duro y del tamaño que el administrador del equipo crea pertinente sin afectar el rendimiento del sistema, siempre y cuando se tomen en cuenta los requerimientos mínimos para lograr el funcionamiento del sistema. En el caso del sistema operativo Linux ó Unix en general, el administrador tiene la libre elección sobre los filesystems y el tamaño que tendrá cada uno de estos, tomando en cuenta que deberá asignar uno de ellos con el espacio mínimo requerido para la instalación del sistema.

Dentro de este mismo contexto, cabe señalar que la forma en que se describe la instalación del sistema tanto en sistema operativo Windows como sistemas Unix, no es la única, pero sí la más recomendable para que el sistema logre su funcionamiento óptimo, en este sentido, se deja a consideración del administrador, el seguir dicha instalación o diseñar alguna otra forma de hacerlo, pero no se asegura que el sistema funcione al 100% en todas sus características. Para una instalación diferente a la que se planteó anteriormente, será necesario consultar la documentación correspondiente en <http://www.php.net>, <http://www.apache.org>, <http://www.mysql.com> y <http://www.pear.php.net>.

El respaldo de la información del sistema es de vital importancia dentro de cualquier aplicación, motivo por el cual se recomienda que antes de comenzar el

Recomendaciones

ciclo de inscripciones a laboratorios y al final de cada ciclo escolar, se haga un respaldo del directorio donde reside el sistema, así como de la base de datos con el siguiente comando desde el sistema operativo:

```
mysqldump -databases laboratorio >> laboratorio.sql
```

Para mayor información acerca del comando anterior y sus parámetros, se puede consultar la página <http://www.mysql.com>

Por último, en caso de que se desee agregar nuevos aspectos funcionales al sistema, es necesario consultar la documentación API que se encuentra en el directorio docs del cd de instalación que acompaña al presente trabajo, en el cual se describen los archivos utilizados en la programación de la aplicación.

Conclusiones

Conclusiones

Los problemas que existen en cualquier proceso manejado de forma manual pueden tener grandes impactos en muchos aspectos afectando directamente a la productividad de la entidad que lleva a cabo dicho proceso. En este caso, la unidad encargada de las inscripciones a los laboratorios de ingeniería de la Facultad de Estudios Superiores Aragón actualmente presenta, como se ha comentado, una serie de problemas en dicho proceso y es en este punto donde se puede ver los beneficios que se pueden alcanzar al automatizar un determinado proceso mediante el uso de sistemas computacionales.

En el desarrollo del presente trabajo, se han aplicado los conocimientos obtenidos por los autores, durante el curso de la carrera de Ingeniería en Computación en la Facultad antes mencionada, para lograr los objetivos planteados al inicio de este escrito, aunado a ello, la aplicación de las diferentes reglas que se siguen actualmente en el respectivo proceso, así como la aportación de nuevos aspectos que ofrecen al sistema y al proceso mismo, aportaron la funcionalidad adicional que ayudarán a la gestión de la entidad correspondiente.

Cabe señalar que los beneficios que trae consigo el automatizar los procesos que existen dentro de cualquier institución, son siempre de alta importancia en cuanto a la reducción de tiempo y costos de operación, aunque para lograr alcanzar las metas establecidas, se deben contemplar ciertos factores en el diseño y desarrollo de sistemas, como son: la total comprensión de los requerimientos técnicos y funcionales del usuario, la adecuada selección de metodologías de desarrollo (patrones de diseño, técnicas de programación, implementación de las herramientas, etc.) y el costo de las herramientas que, en conjunto, formarán en su totalidad el sistema planteado.

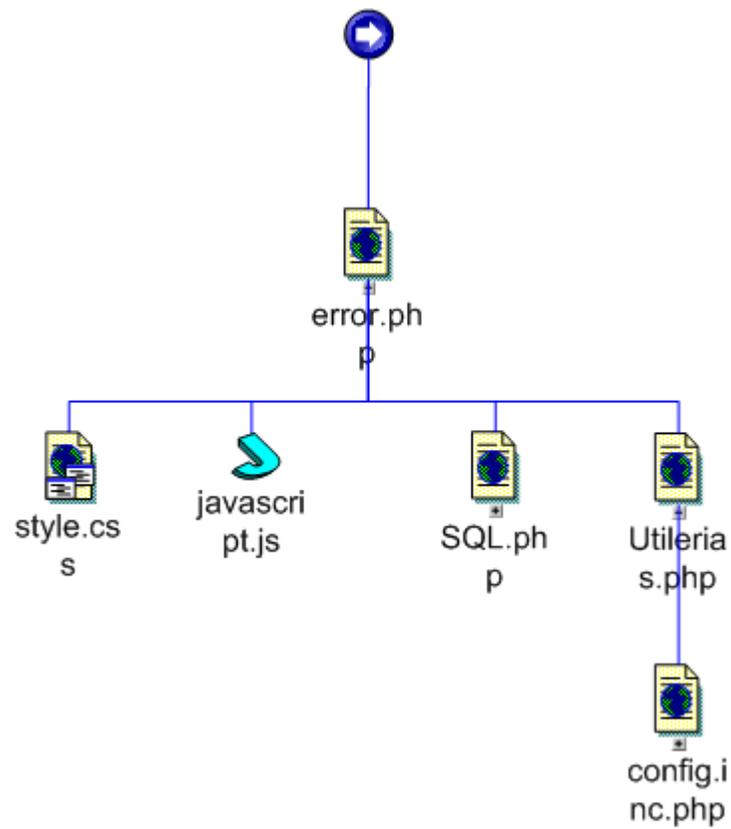
Tomando en cuenta el factor costo, se ha optado para el diseño, desarrollo e implementación del sistema comprendido en este trabajo, herramientas de tipo “software libre”, como su nombre define, son aplicaciones las cuales no causan

Conclusiones

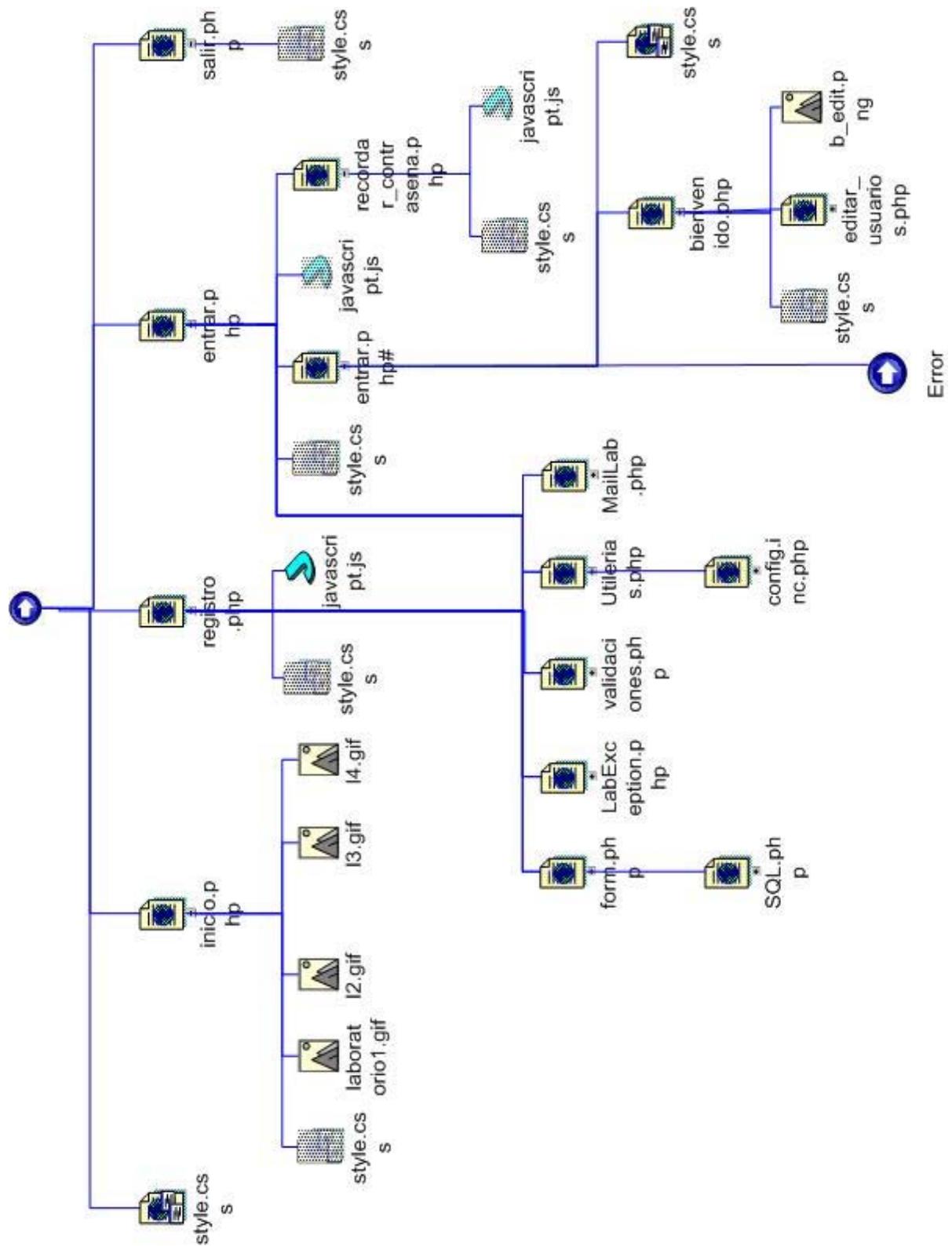
costo alguno en cuanto al licenciamiento se refiere, incluso muchas de ellas, ponen a disposición el código fuente de las mismas para que la comunidad interesada, pueda desarrollar nuevos módulos, parches y hasta nuevas versiones completas.

Bajo este contexto, se ha llevado a cabo el desarrollo de la aplicación, utilizando herramientas “software libre” sin preocuparse por obtener un presupuesto que seguramente nunca se destinaría para tal efecto en una institución pública como la FES Aragón, donde se ha llevado a cabo la aplicación.

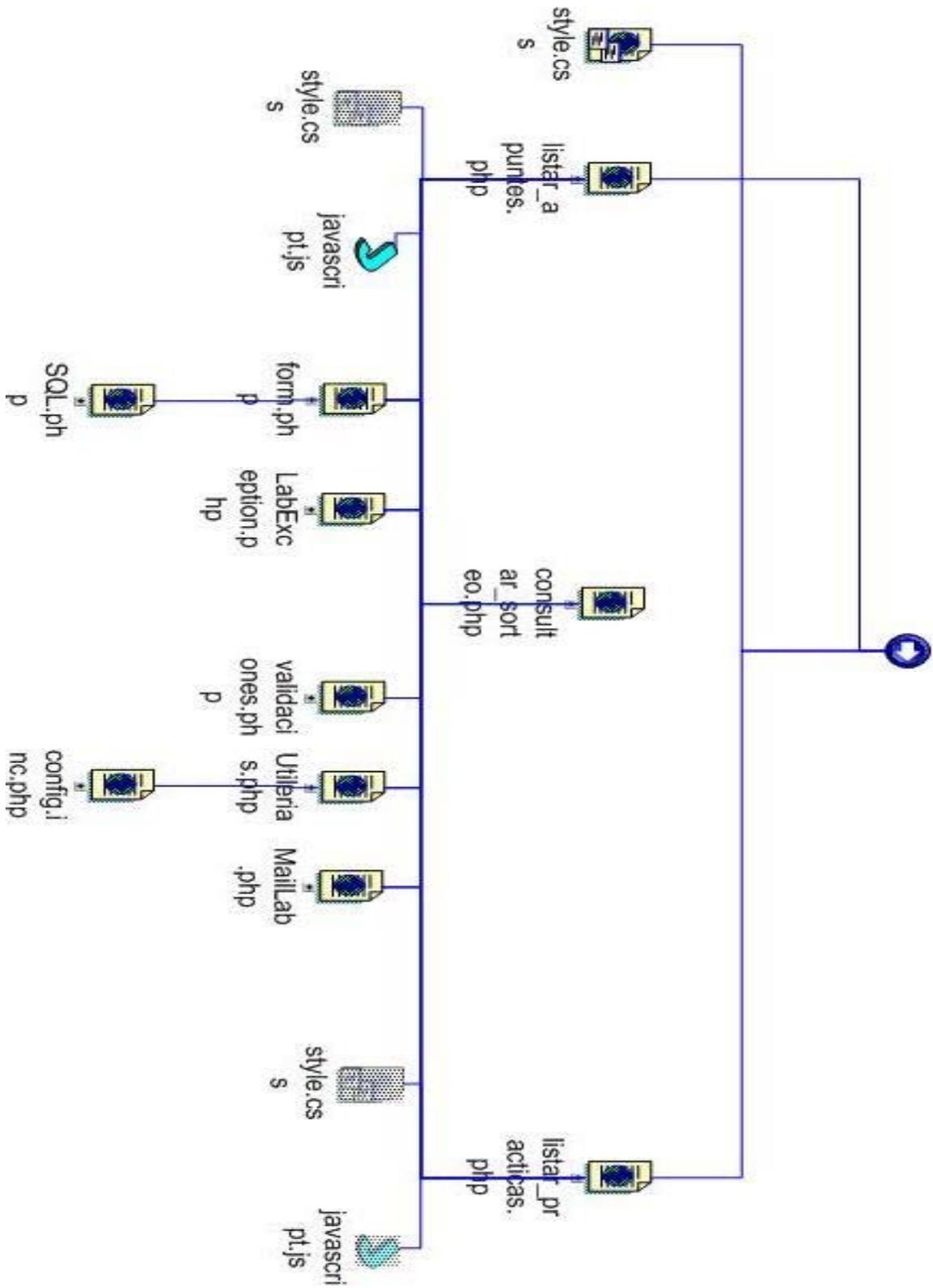
Anexos



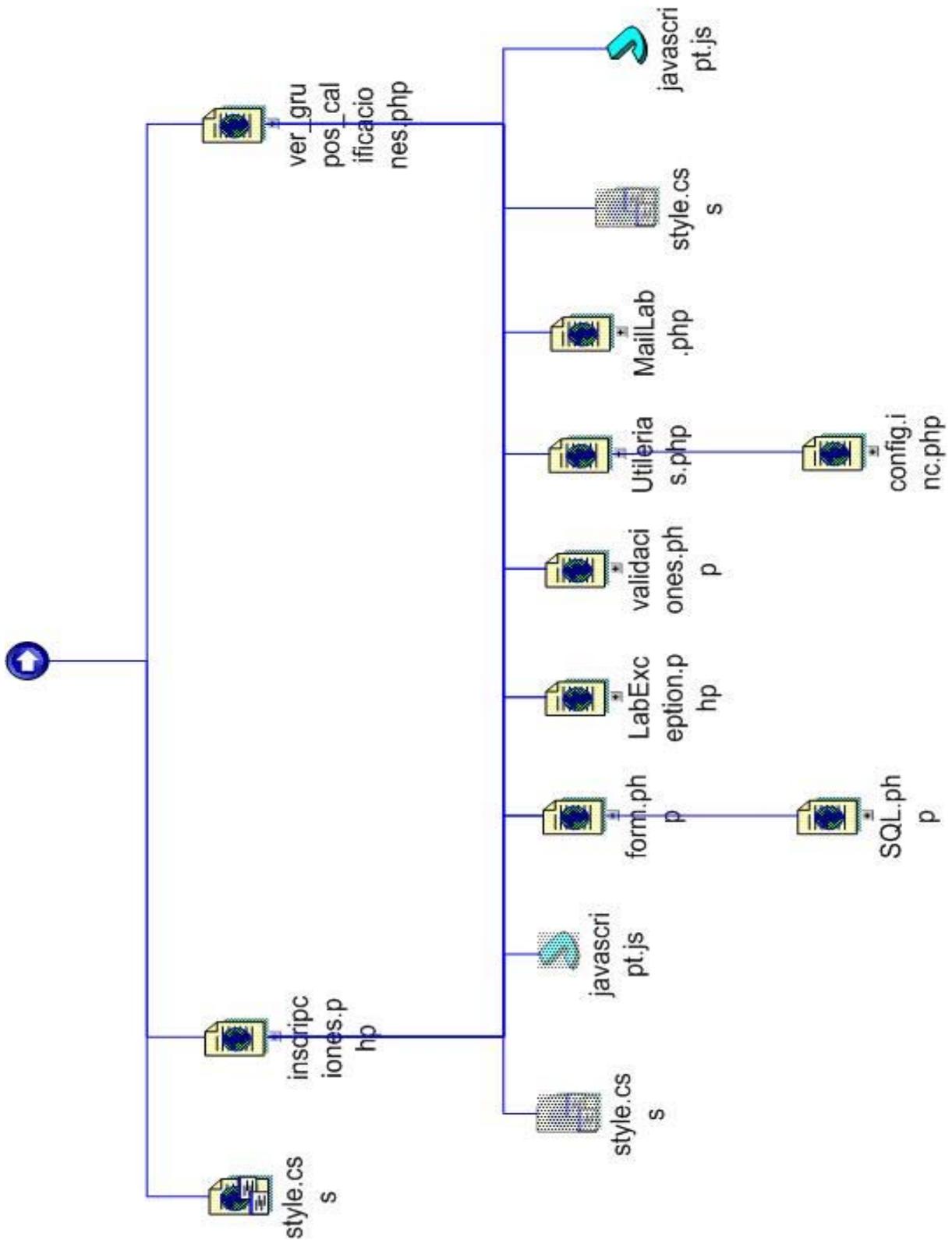
Error.



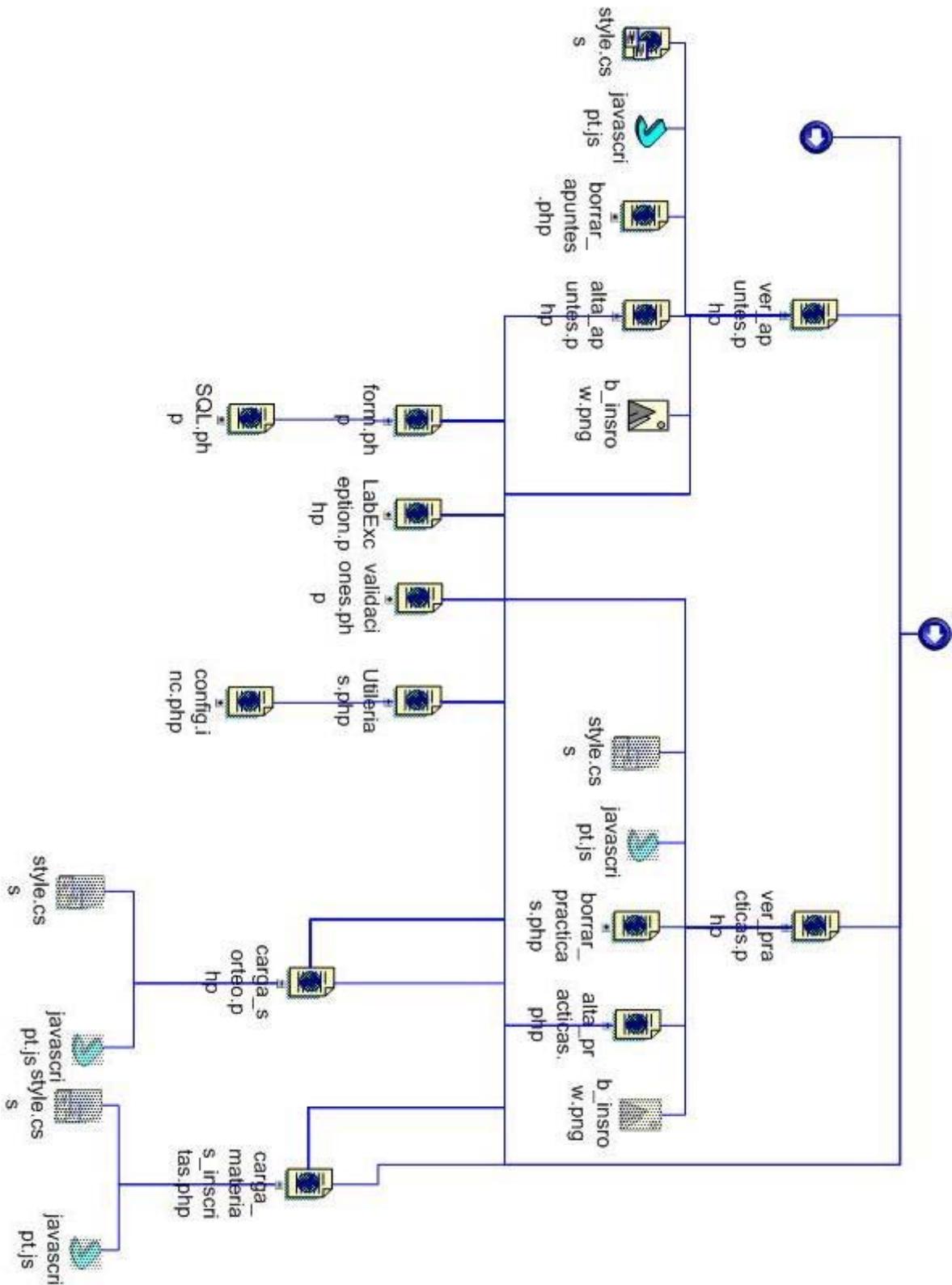
Sistema.



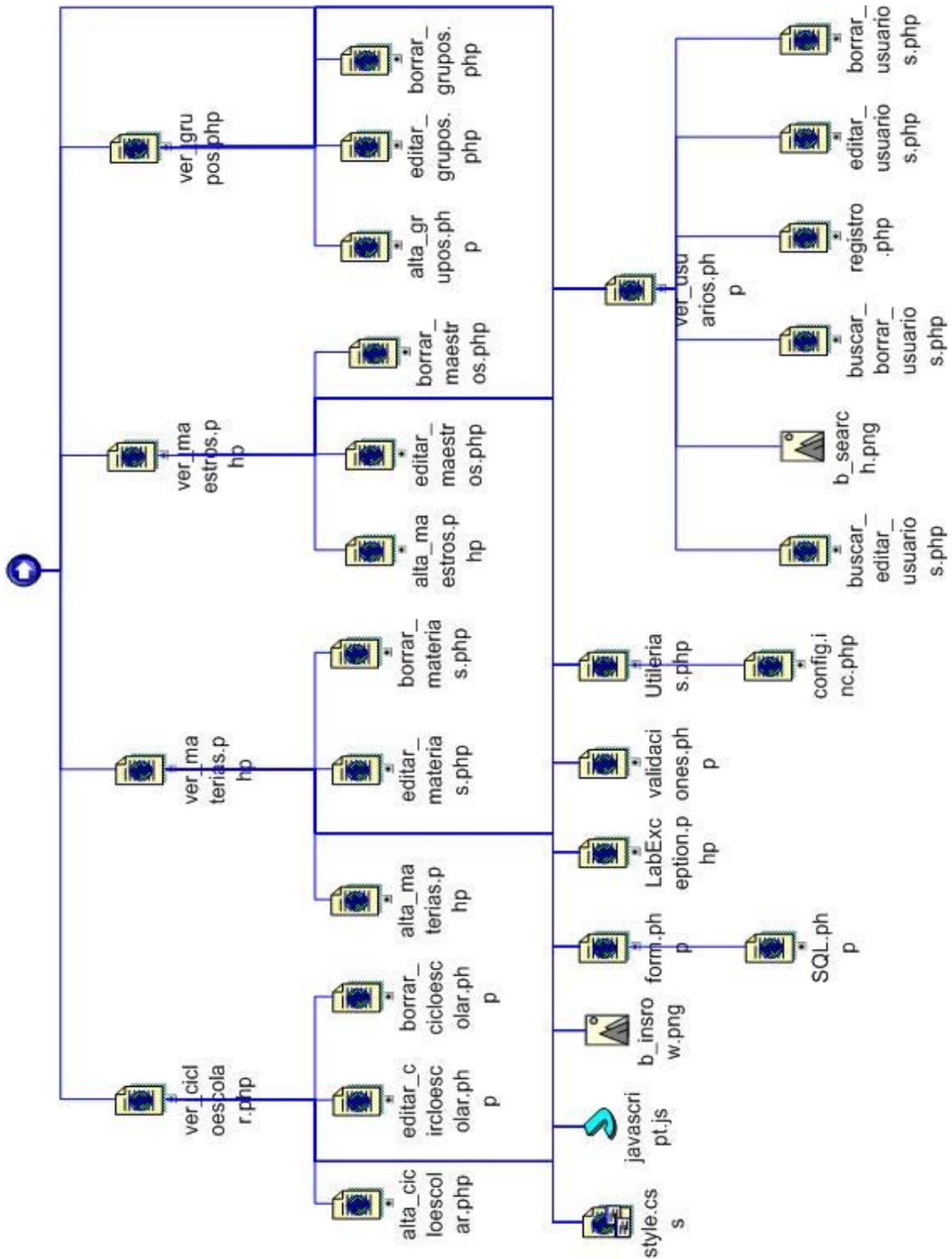
Utilerías.



Inscripciones y Calificaciones.



Administración.



Administración (Continuación).

Anexo 2.

Script de la Base de Datos.

```
drop database `laboratorio`;
```

```
create database `laboratorio`;
```

```
use laboratorio;
```

```
drop table if exists materia;
```

```
create table materia (
```

```
    materiaid int not null auto_increment,
```

```
    nombre varchar(70) null,
```

```
    clave varchar(50) null,
```

```
    fechacreacion date null default '0000-00-00' ,
```

```
    fechamodificacion date null default '0000-00-00' ,
```

```
    materia2usmodificoid int null,
```

```
    primary key(materiaid),
```

```
    unique index clave(clave)
```

```
);
```

```
insert into `materia` values (1, 'Diseño Lógico', '1720,0142', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (2, 'Dispositivos Electrónicos', '0138', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (3, 'Electrónica de Potencia', '0145', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (4, 'Microprocesadores', '0586', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (5, 'Amplificación de Señales', '0023', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (6, 'Electrónica Industrial', '1602', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (7, 'Comunicaciones Digitales', '0109', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (8, 'Filtrado y Modulación', '1708', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (9, 'Electricidad y Magnetismo', '0082,0085', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (10, 'Análisis de Circuitos Electricos', '1546', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (11, 'Medición e Instrumentación', '1715', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (12, 'Máquinas Eléctricas', '0549', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (13, 'Sistemas Electricos de Potencia II', '1064', '0000-00-00', '0000-00-00', 0);
```

```
insert into `materia` values (14, 'Control Digital', '0114', '0000-00-00', '0000-00-00', 0);
```

Anexos

```
insert into `materia` values (15, 'Control Analógico', '0112', '0000-00-00', '0000-00-00', 0);
```

```
drop table if exists `privilegio`;
```

```
create table privilegio (  
    privilegioid int not null auto_increment,  
    nombre varchar(20) null,  
    fechacreacion date null default '0000-00-00' ,  
    fechamodificacion date null default '0000-00-00' ,  
    privilegio2usmodificoid int null,  
    primary key(privilegioid)  
);
```

```
insert into `privilegio` values (1, 'Administrador', '0000-00-00', '0000-00-00', 0);
```

```
insert into `privilegio` values (2, 'Maestro', '0000-00-00', '0000-00-00', 0);
```

```
insert into `privilegio` values (3, 'Alumno', '0000-00-00', '0000-00-00', 0);
```

```
drop table if exists `maestro`;
```

```
create table maestro (  
    maestroid int not null auto_increment,  
    nombre varchar(50) null,  
    apepaterno varchar(50) null,  
    apematerno varchar(50) null,  
    clave varchar(4) null,  
    fechacreacion date null default '0000-00-00' ,  
    fechamodificacion date null default '0000-00-00' ,  
    maestro2usmodificoid int null,  
    primary key(maestroid),  
    unique index clave(clave)  
);
```

```
drop table if exists `escuela`;
```

```
create table escuela (  
    escuelaaid int not null auto_increment,  
    nombre varchar(70) null,
```

Anexos

```
clave varchar(4) null,  
fechacreacion date null default '0000-00-00' ,  
fechamodificacion date null default '0000-00-00' ,  
escuela2usmodificoid int null,  
primary key(escuelaid),  
unique index clave(clave)  
);
```

```
insert into `escuela` values (1, 'Facultad de Estudios Superiores Aragón', '411', '0000-00-00', '0000-00-00', 0);
```

```
drop table if exists `materiainscrita` ;  
create table materiainscrita (  
materiainscritaid int not null auto_increment,  
grupo int null default 0,  
semestre varchar(8) null,  
fechacreacion date null default '0000-00-00' ,  
fechamodificacion date null default '0000-00-00' ,  
ncuenta varchar(9) null,  
clave varchar(4) null,  
materiainscrita2usmodificoid int null,  
primary key(materiainscritaid),  
index semestre(semestre),  
index ncuenta(ncuenta),  
index clave(clave)  
);
```

```
drop table if exists `cicloescolar` ;  
create table cicloescolar (  
cicloescolarid int not null auto_increment,  
semestre varchar(8) null,  
fechainicio date null default '0000-00-00' ,  
fechafin date null default '0000-00-00' ,  
semanacierre varchar(21) null default '0000-00-00-0000-00-00',
```

Anexos

```
hora0 varchar(5) null default '00:00',
hora1 varchar(5) null default '00:00',
hora2 varchar(5) null default '00:00',
hora3 varchar(5) null default '00:00',
hora4 varchar(5) null default '00:00',
hora5 varchar(5) null default '00:00',
hora6 varchar(5) null default '00:00',
hora7 varchar(5) null default '00:00',
hora8 varchar(5) null default '00:00',
hora9 varchar(5) null default '00:00',
diainscripcion1 date null default '0000-00-00' ,
diainscripcion2 date null default '0000-00-00' ,
diainscripcion3 date null default '0000-00-00' ,
diainscripcion4 date null default '0000-00-00' ,
diainscripcion5 date null default '0000-00-00' ,
diainscripcion6 date null default '0000-00-00' ,
diainscripcion7 date null default '0000-00-00' ,
diainscripcion8 date null default '0000-00-00' ,
diainscripcion9 date null default '0000-00-00' ,
diainscripcion10 date null default '0000-00-00' ,
altasbajas date null default '0000-00-00' ,
fechacreacion date null default '0000-00-00' ,
fechamodificacion date null default '0000-00-00' ,
cicloescolar2usmodificoid int null,
primary key(cicloescolarid),
index semestre(semestre)
);
```

```
drop table if exists `error` ;
create table error (
errorid int not null auto_increment,
codigo int null default 0,
mensaje varchar(200) null,
```

Anexos

```
sugerencia varchar(200) null,  
tipo int null default 0,  
primary key(errorid),  
unique index codigo(codigo)  
);
```

```
insert into `error` values (1, 1, 'Los parámetros deben ser contenidos en un arreglo.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (2, 2, 'El tipo de elemento del formulario no existe.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (3, 3, 'El parámetro especificado no existe para el elemento a crear.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (4, 4, 'No existe la variable global.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (5, 100, 'Eo se puede conectar a la base de datos.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (6, 101, 'Error al realizar la consulta.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (7, 102, 'Error al actualizar el registro.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (8, 103, 'Error al insertar los datos a la base de datos.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (9, 104, 'Error al borrar el registro de la base de datos.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (10, 200, 'Se ha denegado el acceso al sistema.', 'Verificar el nombre de usuario y/o contraseña, consulte al administrador del sistema.', 1);
```

```
insert into `error` values (11, 201, 'No tienes acceso a esta página.', 'Debes entrar al sistema y dependiendo de tus privilegios, tendrás acceso o no a este módulo, consulte al administrador del sistema.', 1);
```

```
insert into `error` values (12, 202, 'No se puede guardar la contraseña del usuario.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (13, 203, 'No se puede obtener la contraseña del usuario.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (14, 204, 'El parámetro especificado no esta contenido dentro de la sesion actual.', 'Consulte al administrador del sistema.', 1);
```

```
insert into `error` values (15, 205, 'El atributo requerido es un arreglo.', 'Consulte al administrador del sistema.', 1);
```

Anexos

```
insert into `error` values (16, 206, 'No has confirmado tu registro al sistema.', 'Confirma tu registro siguiendo el mensaje del correo recibido o contacta al administrador del sistema.', 1);
```

```
insert into `error` values (17, 207, 'El tipo del archivo que quieres subir como listado no es el correcto.', 'Verifica que el listado haya sido guardado con el formato correcto o contacta al administrador del sistema.', 1);
```

```
insert into `error` values (18, 208, 'El número de columnas del archivo, no coincide con el número de columnas esperadas.', 'Verifica que el archivo contenga el número de columnas esperadas o contacta al administrador del sistema.', 1);
```

```
insert into `error` values (19, 209, 'El orden de las columnas contenidas en el archivo, no es el correcto.', 'Verifica que el orden de las columnas del archivo sea el correcto o contacta al administrador del sistema.', 1);
```

```
drop table if exists `carrera`;
```

```
create table carrera (  
  carreraid int not null auto_increment,  
  nombre varchar(50) null,  
  clave varchar(4) null,  
  fechacreacion date null default '0000-00-00' ,  
  fechamodificacion date null default '0000-00-00' ,  
  carrera2escuelaid int null default 0,  
  carrera2usmodificoid int null,  
  primary key(carreraid),  
  index carrera_fkindex2(carrera2escuelaid),  
  unique index clave(clave),  
  foreign key(carrera2escuelaid)  
    references escuela(escuelaid)  
    on delete restrict  
    on update cascade  
);
```

```
insert into `carrera` values (1, 'Ingeniería en Computación', '0110', '0000-00-00', '0000-00-00', 1, 0);
```

```
insert into `carrera` values (2, 'Ingeniería Mecánica Eléctrica', '0116', '0000-00-00', '0000-00-00', 1, 0);
```

```
insert into `carrera` values (3, 'Ingeniería Civil', '0107', '0000-00-00', '0000-00-00', 1, 0);
```

Anexos

```
drop table if exists `usuario`;  
create table usuario (  
  usuarioid int not null auto_increment,  
  usuario2carreraid int not null,  
  nombre varchar(50) null,  
  apepaterno varchar(50) null,  
  apematerno varchar(50) null,  
  ncuenta varchar(9) null,  
  telefono int null,  
  direccion varchar(80) null,  
  email varchar(50) null,  
  msn varchar(50) null,  
  usuario varchar(50) null,  
  contrasena varchar(32) null,  
  usuarionuevo int null default 0,  
  confirmacion char(1) null default 'N',  
  fechacreacion date null default '0000-00-00' ,  
  fechamodificacion date null default '0000-00-00' ,  
  usuario2privilegioid int null default 3,  
  usuario2usmodificoid int null,  
  primary key(usuarioid),  
  index usuario_fkindex2(usuario2privilegioid),  
  unique index ncuenta(ncuenta),  
  unique index usuario(usuario),  
  index contrasena(contrasena),  
  foreign key(usuario2privilegioid)  
    references privilegio(privilegioid)  
    on delete restrict  
    on update cascade,  
  foreign key(usuario2carreraid)  
    references carrera(carreraid)  
    on delete restrict  
    on update cascade
```

Anexos

);

```
insert into `usuario` values (1, 1, 'Administrador', "", "", "", 0, "", "", "", 'adminsli',  
'1d0258c2440a8d19e716292b231e3190', 0, 'Y', '0000-00-00', '0000-00-00', 1, 0);
```

```
drop table if exists `sorteo`;
```

```
create table sorteo (
```

```
    sorteoid int not null auto_increment,
```

```
    ncuenta varchar(9) null,
```

```
    nombre varchar(200) null,
```

```
    horainscripcion varchar(5) null,
```

```
    diainscripcion date null default '0000-00-00',
```

```
    fechacreacion date null default '0000-00-00' ,
```

```
    fechamodificacion date null default '0000-00-00',
```

```
    sorteo2semestreid int null,
```

```
    sorteo2usmodificoid int null,
```

```
    primary key(sorteoid),
```

```
    index sorteo_fkindex2(sorteo2semestreid),
```

```
    index ncuenta(ncuenta),
```

```
    foreign key(sorteo2semestreid)
```

```
        references cicloescolar(cicloescolarid)
```

```
        on delete restrict
```

```
        on update cascade
```

```
);
```

```
drop table if exists `laboratorio`;
```

```
create table laboratorio (
```

```
    laboratoriod int not null auto_increment,
```

```
    horario varchar(11) null,
```

```
    ubicacion varchar(50) null,
```

```
    diasemana varchar(50) null,
```

```
    cupo int null default 0,
```

```
    fechacreacion date null default '0000-00-00' ,
```

Anexos

```
fechamodificacion date null default '0000-00-00' ,
laboratorio2materiaid int null default 0,
laboratorio2maestroid int null default 0,
laboratorio2semestreid int null default 0,
laboratorio2usmodificoid int null,
primary key(laboratorioid),
index laboratorio_fkindex2(laboratorio2maestroid),
index laboratorio_fkindex3(laboratorio2semestreid),
index laboratorio_fkindex4(laboratorio2materiaid),
foreign key(laboratorio2maestroid)
  references maestro(maestroid)
  on delete restrict
  on update cascade,
foreign key(laboratorio2semestreid)
  references cicloescolar(cicloescolarid)
  on delete restrict
  on update cascade,
foreign key(laboratorio2materiaid)
  references materia(materiaid)
  on delete restrict
  on update cascade
);
```

```
drop table if exists `inscripcion`;
create table inscripcion (
  inscripcionid int not null auto_increment,
  calificacion char(2) null,
  fechacreacion date null default '0000-00-00' ,
  fechamodificacion date null default '0000-00-00' ,
  inscripcion2laboratorioid int null default 0,
  inscripcion2materiainscritaid int null default 0,
  inscripcion2usuarioid int null default 0,
  semestre varchar(8) null,
```

Anexos

```
inscripcion2usmodificoid int null,  
primary key(inscripcionid),  
index inscripcion_fkindex2(inscripcion2laboratoriod),  
index inscripcion_fkindex4(inscripcion2usuarioid),  
index inscripcion_fkindex5(inscripcion2materiainscritaid),  
index semestre(semestre),  
foreign key(inscripcion2laboratoriod)  
  references laboratorio(laboratoriod)  
  on delete restrict  
  on update cascade,  
foreign key(inscripcion2usuarioid)  
  references usuario(usuarioid)  
  on delete restrict  
  on update cascade,  
foreign key(inscripcion2materiainscritaid)  
  references materiainscrita(materiainscritaid)  
  on delete restrict  
  on update cascade  
);
```


Glosario

- ACTIVEX** Un conjunto de tecnologías desarrolladas por Microsoft que permiten a los componentes de software interactuar entre sí en un ambiente conectado de red, como Internet, independientemente del lenguaje de desarrollo en que fueron creados.
- APACHE** Servidor Web de código libre cuya implementación se realiza de forma colaborativa con prestaciones y funcionalidades equivalentes a los servidores comerciales.
- APPLETS** Pequeñas aplicaciones escritas en Java que se incluyen en páginas Web (HTML) y que se puede ejecutar en cualquier navegador que disponga de un intérprete Java, sin que para su uso necesite intercambiar Información con el servidor ya que siempre se ejecuta en el “cliente”.
- BROWSER** Navegador Web o Web browser es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores Web de todo el mundo a través de Internet.
- C** C es un lenguaje de programación creado en 1969 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell basándose en los lenguajes BCPL y B.
- CGI** Common Gateway Interface. Literalmente Interfaz Común de Comunicación, tecnología de la World Wide Web que permite a un cliente (explorador Web) solicitar datos de un programa ejecutado en un servidor Web.

CSS	Cascade Style Sheets, Las hojas de estilo en cascada son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.
DHTML	Dynamic Hipertext Markup Language. Designa el conjunto de técnicas que permiten crear sitios Web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript) y el lenguaje de Hojas de estilo en cascada (CSS).
HTML	HyperText Markup Language. Lenguaje de etiquetas de hipertexto es un lenguaje de etiquetas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.
HTTP	Hyper Text Transfer Protocol. El hipertexto es el contenido de las páginas Web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página Web, y la respuesta de esa Web, remitiendo la información que se verá en pantalla.
JAVA	Java es una plataforma de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales
JAVASCRIPT	Lenguaje de Script para páginas Web, basado en la sintaxis de Java

JPEG	Joint Photographic Experts Group. Es un algoritmo diseñado para comprimir imágenes con 24 bits de profundidad o en escala de grises.
JSCRIPT	Implementación de JavaScript por parte de Microsoft, compatible con el JavaScript de Netscape Navigator 2.0.
JVM	Java Virtual Machina. Máquina Virtual de Java. Es un programa ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial, el Java bytecode, el cual es generado por el compilador del lenguaje Java.
LINUX	Versión Shareware del conocido sistema operativo Unix. Es un sistema multitarea multiusuario de 32 bits para PC.
MYSQL	MySQL Es un gestor de Bases de Datos multiusuario que gestiona bases de datos relacionales poniendo las tablas en archivos diferenciados.
PDF	Portable Document Format. Formato de Documento Portátil. Es una forma de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.
PERL	Practical Extraction and Report Language. Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell, para la administración de tareas propias de sistemas UNIX

PHP	Hypertext Preprocessor. Originado inicialmente del nombre PHP Tools, o Personal Home Page Tools es un lenguaje de programación interpretado tipo script para entornos Web.
PNG	Portable Network Graphics. Graficos Portables de Red. Formato gráfico muy completo especialmente pensado para redes.
SCRIPT	Del inglés, guión. Instrucciones internas de una aplicación.
SERVLETS	Los servlets son objetos que corren dentro del contexto de un servidor de aplicaciones por ejemplo Tomcat y extienden su funcionalidad.
SQL	Structured Query Language. Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.
UNIX	Sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.
URI	Uniform Resource Identifier, literalmente Identificador Uniforme de Recurso. Conjunto de caracteres que identifica un recurso de red, mediante un nombre o utilizando sus datos de localización.
URL	Universal Resource Locator. Localizador Universal de Recursos. Sistema unificado de identificación de recursos en

la red. Es el modo estándar de proporcionar la dirección de cualquier recurso en Internet.

VBSCRIPT Lenguaje de programación para WWW desarrollado por Microsoft. Se destaca que VBScript y JavaScript de Netscape son muy similares.

XML eXtensible Markup Language. Lenguaje de etiquetas ampliable o extensible desarrollado por el World Wide Web Consortium (W3C).

Clase Entidad dentro de la programación orientada a objetos, mediante la cual se modelan los diferentes actores identificados en algún proceso que será automatizado mediante un sistema informático, desarrollado con la técnica de programación también conocida como POO.

Herencia Característica que algunas técnicas de programación utilizan para evitar escribir el mismo código en dos o más lugares dentro del desarrollo de software, también conocida como *reutilización del software*, en la cual se crean nuevas entidades a partir de otras ya existentes.

Polimorfismo Característica utilizada en la programación orientada a objetos, mediante la cual, un objeto puede tomar diferentes formas y con esto, llevar a cabo un mismo proceso, aunque de diferente forma.

Glosario

POO Programación Orientada a Objetos. Técnica de programación utilizada ampliamente en el desarrollo de sistemas informáticos.

UML Unified Modeling Language. Es un lenguaje gráfico para visualizar todas o algunas de las características de un sistema de cualquier índole.

Bibliografía

Bibliografía

DEITEL, Harvey M., Paul J. Deitel. *Cómo Programar en Java*. 1ª ed. México: Pearson Educación, 1998. 1056p.

DEITEL, Harvey M., Paul J Deitel. *Cómo Programar en C/C++*. 2ª ed. México: Prentice Hall, 1994. 927p.

GIL, Rubio Francisco Javier. *Creación de Sitios Web con PHP 4*. 1ª ed. España: Mc Graw Hill, 2001. 547p

PETTERSEN, Richard. *Linux Manual de Referencia*. 2ª ed. España: Mc Graw Hill, 2001. 1306p.

POWELLL, Thomas, Fritz Schneider. *Manual de Referencia JavaScript*. 1ª ed. España: Mc Graw Hill, 2002.1078p.

SILLBERSCHATZ Abraham. *Fundamentos de Bases de Datos*. 3ª ed. España: Mc Graw Hill, 1998. 641p.

Masografía.

<http://dev.mysql.com/doc/refman/4.1/en> (01 mayo 2006 : 14:00 hr.)

<http://httpd.apache.org/docs/1.3> (01 mayo 2006 : 14:00 hr.)

<http://www.pear.php.net/manual/en> (01 mayo 2006 : 14:00 hr.)

<http://www.php.net/manual/es> (01 mayo 2006 : 14:00 hr.)