PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL TITULO DE INGENIERIA DE SISTEMAS

TEMA:

"LA UTILIZACIÓN DE BASES DE DATOS Y GENERACIÓN DE APLICACIONES WEB DINÁMICAS EN EL SERVIDOR WEB DE LA PUCESA CON SISTEMA OPERATIVO SOLARIS".

WILSON FAVIAN TOASA ROMO

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
PROGRAMAS ACADEMICOS AMBATO
BIBLIOTECA

DIRECTOR DE LA DISERTACION: Ing. Victor Chuncha

AMBATO, 2005

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO
SECRETARIA
ESCUELA DE INGENIERIA
DE SISTEMAS

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL TITULO DE INGENIERIA DE SISTEMAS

TEMA:

"LA UTILIZACIÓN DE BASES DE DATOS Y GENERACIÓN DE APLICACIONES WEB DINÁMICAS EN EL SERVIDOR WEB DE LA PUCESA CON SISTEMA OPERATIVO SOLARIS".

DIRECTOR:



Ing. Victor Chuncha

AMBATO, 2005

DEDICATORIA

A mis padres, que con sacrificio, cariño y afecto supieron darme la educación y la oportunidad de alcanzar el éxito profesional en la vida.

W.F.T.R.

INDICE

INTRODUCCION

CAPITULO I

1.- Planteamiento de el Proyecto de Investigación 1

1.1 Planteamiento del Problema y Problematización 1

1.2 Delimitación 2

1.3 Importancia y Justificación 3

1.4 Objetivos 3

1.5 Hipótesis 4

1.6 Metodología de la Investigación 5

1.6.1 Población de la Investigación 5

CAPITULO II

2. Marco Teórico 6

2.1 Redes en Internet 6

2.1.1 Introducción 6

2.1.2 Aspectos Generales 9

2.1.2.1 Conceptos Básicos de Internet e Intranet 9

2.1.2.2 La Conveniencia de la Intranet 11

2.1.2.3 Las Computadoras en las Comunicaciones 12

2.1.3 La World Wide Web 13

2.1.3.1 Concepto de Dominio 13

2.1.3.2 Exploradores en Internet 16

2.1.3.3 Acceso al Web 17

2.1.3.4 El WWW ó telaraña a escala mundial 18

2.1.4 Bases de Datos en Internet 21

2.1.4.1 Creando las DB 21

2.1.4.2 Obtener un DNS 22

2.1.4.3 Instalación de PWS y SAMBAR SERVER 25

2.1.4.4 Instalación de Perl, Jsp y Php3 26

2.1.4.5 Tecnología Asp 29

2.1.4.6 Tecnología PHP 36

2.2 Lenguajes para el Diseño de Paginas Web 38

2.2.1 Lenguaje Html 38
2.2.2 Lenguaje Javascript 41
2.3 Diseño de Contenido Multimedia 43
2.3.1 Procesamiento de Imágenes con Fireworks 43
2.3.2 Macromedia Flash 43
2.3.2 Xara Web Style 43

CAPITULO III

3. Realización del Proyecto 44
3.1 Análisis de Diseño 44
3.1.1 Software existente en los Servidores Web 44
3.1.2 Hardware existente en los Servidores Web 45
3.2 Bases de Datos MySQL en Servidores Web UNIX 46
3.2.1 Instalación 46
3.3 Programación PHP 49
3.3.1 Conceptos 49
3.3.2 PHP Básico 49
3.3.3 Variables 50
3.3.4 Operadores 51
3.3.4.1 Operadores Aritméticos 51
3.3.4.2 Comparación 52
3.3.4.3 Operadores Lógicos 53
3.3.5 Sentencias 54
3.3.5.1 Sentencias Condicionales 54
3.3.6 Bucles 56
3.3.7 Impresión 58
3.3.8 Manejo de Cadenas 59
3.3.9 Funciones 60
3.3.10 Programación PHP Avanzado 61
3.4. Bases de Datos MySQL 64
3.4.1 Normalización 64
3.4.2 Reglas de la Normalización 64
3.4.3 Modelo Lógico Físico de la Base de Datos 65

3.4..3.1 Diseño Conceptual 65
3.4..3.2 Diseño Lógico 66
3.4.3.3 Diseño Físico 67
3.5 Administración de la Base de Datos 67
3.5.1 Conexión a la Base de Datos 68
3.5.2.Consulta a la Base de Datos 69
3.5.3 Adición de registros 70
3.5.4 Eliminación de registros 71
3.5.5 Actualización de registros 74

CAPITULO IV

4. Validación y Verificación de los Resultados 75
4.1 Comprobación de Hipótesis 75
4.1.1 Método Lógico 75
4.1.1.1 Certificado del Administrador del Centro de Computo 77
4.1.1.2 Operacionalización de las Variables 78
4.1.1.3 Análisis del Informe presentado por el Administrador 79
4.1.2 Comprobación Estadística 80
4.1.2.1 Formato de preguntas de la encuesta 80
4.1.2.2 Análisis estadístico de los resultados obtenidos en la encuesta 81
4.1.2.3 Gráficos estadísticos de los resultados 82
4.2. Validación 83
4.3 Conclusiones 87
4.4 Recomendaciones 88

BIBLIOGRAFIA

ANEXOS

A. MANUAL DE USUARIO

B. DIAGRAMA DE FLUJO DE EJECUCIÓN DE LOS SUBPROGRAMAS

C. ASIGNACIÓN DE NOMBRES, FUNCIONES Y PARAMETROS DE LOS PROGRAMAS

D. DOCUMENTACION DE PROGRAMAS

E. ENCUESTAS REALIZADAS

GLOSARIO

INTRODUCCIÓN

Como una parte característica de este fin de siglo, los desarrollos científicos experimentados en las dos ultimas décadas constituyen sin lugar a dudas una autentica revolución, cuyas consecuencias sociales, económicas y culturales cambiaran nuestra vida y a la sociedad que hoy conocemos.

Pero aunque este impulso tuvo mas auge en los países desarrollados, la masificación del fenómeno ha repercutido en todos los continentes, incluso en aquellos cuya situación económica los había marginado del mismo.

El aparecimiento del Internet en la década de los 90 supone un avance a pasos agigantados para la distribución libre de Información por todo el mundo. En el inicio de la navegación en Internet cuando aparecio el Navegador Mosaic, solo se podía apreciar paginas web básicas, pero esta realidad, poco a poco ha ido cambiando con el desarrollo de una nueva programación de aplicaciones para manipular información de Bases de Datos de sitios en Internet, desde los primeros Cgi, hasta los actuales modelos de programación **PHP-MySQL, JAVA-XML-Oracle, Perl–PostgreSQL, ASP-SqlServer-Access,** Siendo **PostgreSQL** y **MySQL**, bases de datos libres, haciendo entre Apache, PHP y una de ambas la solución "libre" para hacer sitios web dinámicos de alta calidad, sin dependencia tecnológica de ninguna empresa en particular.

El desarrollo de Sitios Web Comerciales con aplicaciones basadas en Bases de Datos y Paginas Web Dinámicas en las empresas seria un avance importante.

Es por esta razón el deseo de realizar el estudio y la implantación de Bases de Datos, al Website de la Pontificia Universidad Católica, el cual permitirá un mejor aprovechamiento de los recursos en Software y Hardware, además brindará mejores búsquedas de información. Estas herramientas trabajaran bajo el Sistema Operativo **SUN SOLARIS**, utilizando la Base de Datos MySql y el lenguaje de programación PHP.

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
PROGRAMAS ACADEMICOS AMBATO
BIBLIOTECA

1.2 Delimitación

El estudio, análisis e implantación de Bases de Datos y desarrollo de páginas Web Dinámicas (Aplicaciones que generan código web con el contenido del resultado de una búsqueda en las Bases de Datos), se realizaran en el Servidor de la Universidad Católica sede Ambato, manteniendo el actual diseño y fortaleciendo el sitio con herramientas de búsqueda.

Para efectuar esta disertación se realizara el análisis previo del Software y Hardware del Servidor, la investigación del lenguaje PHP, el estudio de la Base de Datos MySql, sus herramientas de administración, sus ventajas y desventajas, así como también el Sistema Operativo Sun Solaris y el Servidor Web. Este estudio se realizará durante el período académico 2003 – 2004 y la implantación de Software se realizará en el Servidor Web de la PUCESA.

El tema de la Investigación será:

"LA UTILIZACIÓN DE BASES DE DATOS Y GENERACIÓN DE APLICACIONES WEB DINÁMICAS EN EL SERVIDOR WEB DE LA PUCESA CON SISTEMA OPERATIVO SOLARIS".

1.3 Importancia y justificación

La importancia de este proyecto consiste básicamente, en el fortalecimiento del Web Site de la PUCESA existente con nueva funcionalidad lo que aumentara la representatividad de la Universidad en el Universo Internet, y estará al nivel de otros sitios de Universidades Importantes de América.

Se justifica por:

- La Actualización del Sitio Web de la PUCESA, con herramientas actuales, para el manejo de la Información y sus respectivas consultas.

- El estudio de software con licencia libre, como el PHP y MySQL que brindan facilidades para construir tanto sitios libres como comerciales, con aplicaciones que generen Páginas Dinámicas.

- Las consultas serán más rápidas y estarán manejadas por los criterios de búsqueda del usuario.

1.4 Objetivos

Objetivos Generales

Implementar una Base de Datos y sus aplicaciones de gestión en el Web Site de la PUCESA, acorde al Software existente del Servidor Web.

Objetivos Específicos

- Realizar el estudio previo del Software y Hardware existente en el Servidor Web del PUCESA.

- Desarrollar un buscador en el Sitio Web de la PUCESA que permita la consulta de la información de una forma rápida y ágil, por parte de todos sus usuarios.

- Generar aplicaciones orientadas a Objetos, para manipular Bases de Datos y que éstas a la vez produzcan páginas HTML DINAMICAS como resultado de consultas, actualizaciones y adición de información.

- Desarrollar el Software de Administración de Bases de Datos para el manejo de la información del sitio Web.

- Implementar aplicaciones de procesos de actualización de datos que permitan la actualización ágil de la Base de Datos de Sesiones del Consejo.

1.5 Hipótesis (Hipótesis dirigidas a la Base de Datos de Resoluciones de Consejo)

La implantación de Bases de Datos y la utilización de software libre como PHP y MySQL agilitaran la actualización de datos y el proceso de consulta en el Servidor Web de la PUCESA.

El análisis del software y hardware del servidor web de la PUCESA ayudaran a implementar y mejorar eficientemente su contenido.

El desarrollo de un buscador en el sitio web de la PUCESA permitirá la consulta ágil y rápida de los usuarios

La creación de Aplicaciones de Gestión de Base de Datos permitirán la actualización de la información de forma automática

La instalación de software compatible con el Servidor Web de la PUCESA mejorara su rendimiento y funcionalidad.

1.6 Metodología de la Investigación

Los Paradigmas que utilizaremos en esta disertación son RACIONALISTA Y PRAGMATISTA, Racionalista por cuanto todo lo pensado puede ser transformado al mundo real y Pragmatista debido a que las teorías científicas son acciones a realizarse para solucionar problemas.

El Tipo de Investigación será de Labor de Campo y Básica Aplicada, por cuanto toda la investigación realizada en el Campo se pondrá en Acción.

El nivel de la investigación utilizará los métodos: Diagnóstico y Descriptivo, ya que primero se realizará el Diagnóstico y luego se describirán las posibles soluciones.

Las técnicas a utilizarse en esta investigación serán: Entrevista y Test, por que se realizará un Foda a los usuarios del sitio para describir las Fortalezas, Debilidades, Amenazas y Oportunidades del Objeto de estudio y las Soluciones se realizarán a través de test de comprobación de las aplicaciones.

1.6.1 Población de la Investigación

Se tomará una muestra de usuarios de entre el personal administrativo y profesores de la PUCESA, para lo cual se trabajara con la fórmula:

$$n = \frac{m}{e^2 (m - 1) + 1} \qquad m = 8 \quad e = 5\% \qquad n = \frac{8}{(0.5)^2 (8-1)+1} \qquad n = 3$$

Personas	Número	Porcentaje	Muestra
Administrativo	2	25%	1
Directores	6	75%	2
Total	8	100%	3

CAPITULO II

2. MARCO TEORICO

2.1. Redes de Internet

2.1.1 Introducción

En el año 1957 surgió una red de servidores para intercomunicar los equipos de cómputo de mayor capacidad del ejército de los Estados Unidos y las principales universidades del país para permitir el intercambio de información entre ellas.

Esto implicó que cada dependencia norteamericana y cada universidad tuviera su propia red local de cómputo (con computadoras conectadas a un servidor local), pero que los servidores de cada una de dichas dependencias estuviesen interconectados con los de las demás a modo de intercambiar información entre ellas formando así una red de mayor tamaño, o sea, una red de redes.

La clave de estas redes fue el uso de un protocolo o idioma común, el llamado TCP:

TCP: Transmision control protocol - Protocolo de comunicaciones utilizado como norma para el intercambio de información entre computadoras.

Dicho protocolo fue adoptado como norma para la red que ahora se conoce como Internet. (el conjunto de redes integrado por las redes de las universidades y por las redes de las oficinas gubernamentales que forman el llamado BACKBONE y que hoy en día puede ser accesado por todo público).

BACKBONE: Conjunto de equipos de cómputo y de redes que forman la parte original y medular del INTERNET.

Existen varios Backbone, a traves del Internet, y los cuales son:

AGIS	ANS	ATMnet
BBNplanet	Compuserve	CRL
CWIX	DataXchange	DIGEX
Epoch	GetNet	GlobalCenter
GoodNet	GridNet	IBM
Interconnect	InternetMCI	iSTAR
MCIWorldcom 2000	NapNet	Netrail
NFS	PsiNet	Savvis
Sprint	UUNET	

Mapa del Backbone de IBM

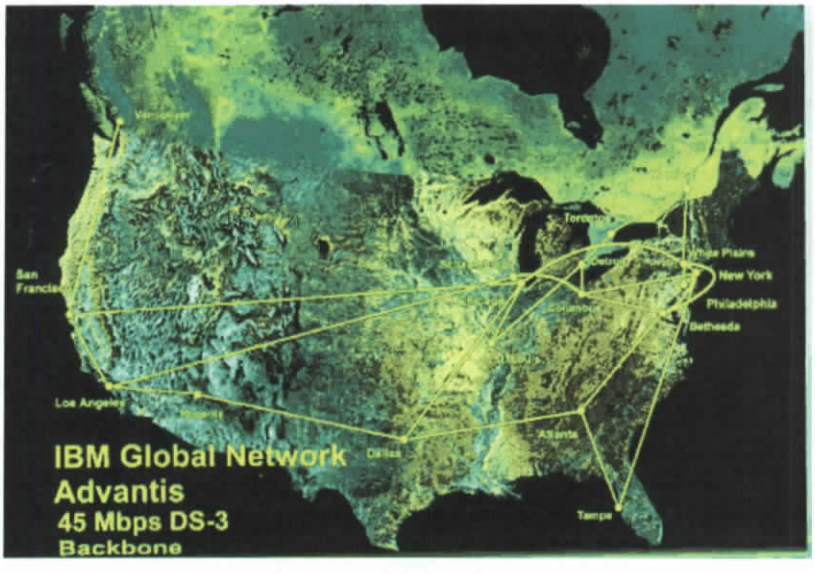


Fig. 1.1

Diagrama de Estructura BACKBONE Internet I

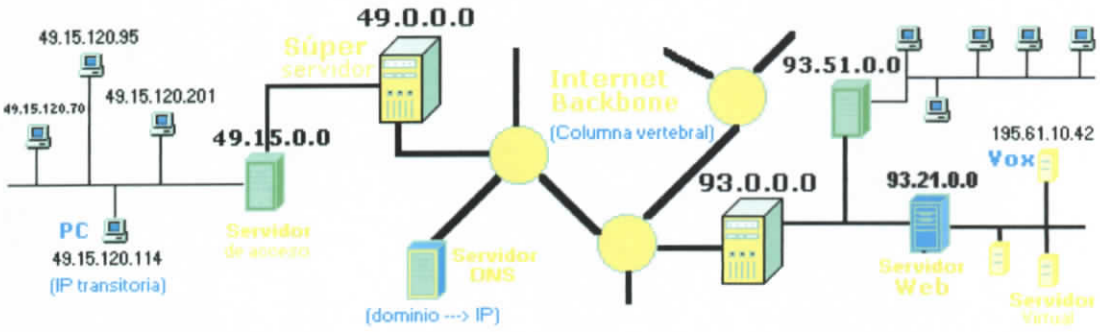
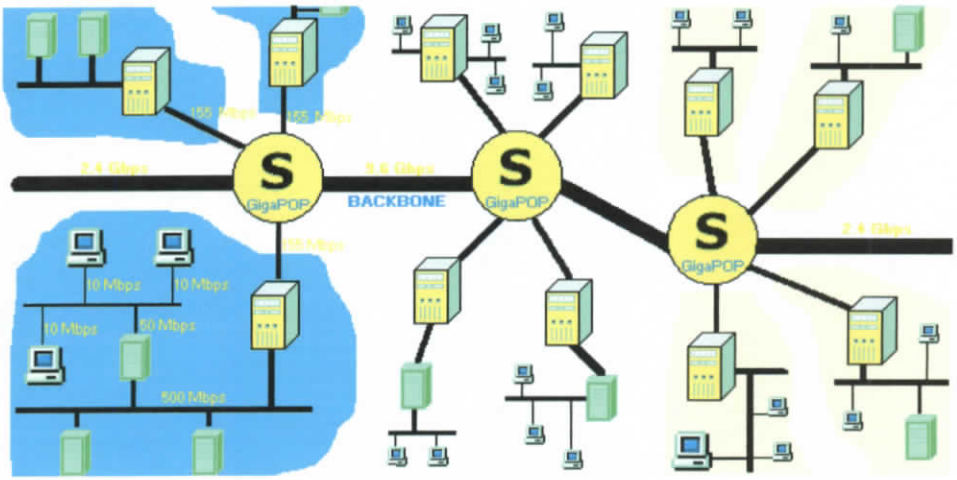


Diagrama de Estructura BACKBONE Internet II



Cuando se adoptó un mismo idioma de comunicaciones, o sea, un protocolo común, fue entonces cuando nació realmente el Internet. Fue también cuando al protocolo TCP le fueron agregadas las siglas IP (correspondientes a Internet Protocol o protocolo de Internet).

El hecho de que la conexión a la red ya no estuviera limitada a las dependencias gubernamentales y a las universidades norteamericanas permitió que equipos de cómputo de otros países se conectaran a la red original, dando lugar a lo que hoy se conoce como el Internet:

INTERNET: Red de equipos de cómputo denominados servidores, hosts o ruteadores que utilizan un protocolo común y que están intercomunicados a modo de permitir el intercambio de información entre ellos y entre los equipos a ellos conectados.

Al hablar de INTERNET, es necesario aclarar que:

Mucho antes del surgimiento de Internet existían diversas redes de información con datos consultables por sus usuarios. Sin embargo, dichas redes no formaban parte del llamado Internet por ser de índole exclusiva de una empresa o por no pertenecer a las universidades o a las dependencias gubernamentales norteamericanas.

Muchas de las redes privadas existentes antes de Internet subsisten hoy en día, pero les han sido agregadas las conexiones pertinentes a modo de poder tener acceso a la red de Internet (por ejemplo, la red Dow Jones, la red America On Line, la red Compuserve, etc.). Sin embargo, los usuarios de Internet, o sea, aquellos cuyos equipos están conectados a Internet, no necesariamente tienen acceso a las redes privadas de antes. Se puede decir que en estos casos la comunicación sólo es factible en un sentido, o sea, de la red privada a la pública, o sea hacia Internet.

•Finalmente, cabe señalar que el número de servidores conectados a la red Internet en el mundo asciende ya a más de 4 millones. Por ello puede decirse que Internet está en todo el globo, salvo en ciertas regiones muy aisladas (por ejemplo, en el Tíbet o en alguno de los países del Africa Central, más por razones políticas que técnicas).

2.1.2 Aspectos Generales

2.1.2.1 Conceptos básicos de Internet y de intranet

En el mundo moderno existe una palabra que denota claramente la tendencia de las operaciones comerciales de fines del siglo XX: globalización

Globalización: Básicamente, implica la realización de operaciones comerciales más allá de nuestra ciudad, de nuestro estado o incluso de nuestro país.

Significa buscar mercados que no estén limitados por fronteras físicas o políticas; significa también la comunicación con posibles proveedores o clientes de lugares tan cercanos o lejanos como las necesidades comerciales los requieran. Representa, en pocas palabras, entrar al siglo XXI aprovechando la tecnología existente para la realización de operaciones comerciales.

Elemento principal de la globalización: Definitivamente, la globalización sería imposible si no existieran los medios de comunicaciones actuales, tanto de transporte de mercancías como de intercambio de información escrita y hablada.

Al analizar los primeros medios electrónicos de comunicaciones hay que considerar las redes de teléfonos y las de telégrafos. Sin embargo, en la actualidad es necesario pensar en redes mucho más sofisticadas en las cuales intervienen toda una gama de dispositivos tales como satélites, computadoras, etc. que operan en forma global y transparente, o sea, en el mundo entero y sin que los usuarios de las mismas perciban su existencia. Ya no son dispositivos que operan en forma aislada en cada país y que están interconectados en algún punto con otros dispositivos similares. Se trata ahora de un mundo de redes en el cual los elementos involucrados son esencialmente computadoras en vez los clásicos conmutadores de antaño. Se trata de la red mundial de comunicaciones conocida como Internet.

Ventajas de la red mundial de comunicaciones: El uso de la red mundial de comunicaciones denominada Internet para fines particulares, o sea, para el intercambio de información entre computadoras de una sóla organización o de un conjunto de empresas en particular es lo que se conoce como intranet. Parte de este concepto de intranet estriba en utilizar también las nuevas técnicas de localización de información, derivadas del Internet.

Intranet: Es la integración de las tecnologias LAN/WAN y las tecnologias de Internet, es decir es una Red de Computadoras de una organización la cual utiliza el protocolo de Comunicación TCP/IP, para fines privados, o sea, para mejorar las comunicaciones entre empleados, independientemente de la ubicación física de éstos.

2.1.2.2 La Conveniencia del intranet

Las empresas que justifican un servicio de intranet, o sea, de una red de comunicaciones que utiliza los servicios públicos de Internet para intercomunicar empleados a través de computadoras, pero permitiendo el acceso a los datos privados de la empresa sólo al personal autorizado, son esencialmente aquellas cuyo volumen de operaciones justifican la existencia de un sistema privado de comunicaciones para enlazar diversas oficinas, múltiples sucursales, etc.

Ventajas del Intranet para las empresas

El concepto de redes de computadoras para el intercambio de información entre oficinas alejadas de una misma empresa no es una novedad. Sin embargo, el costo de dichas redes se ha visto drásticamente reducido merced al uso de la red de computadoras más grande del mundo (conocida como Internet) para fines privados, o sea, para mejorar las comunicaciones entre empleados, independientemente de la ubicación física de éstos.

El objetivo de las redes privadas de comunicaciones es el de optimizar las operaciones dentro de las organizaciones mediante el uso de computadoras, lográndose a la larga una reducción en costos y una mejor difusión o intercambio de información. Sin embargo, el uso de dichas redes privadas muchas veces da lugar a costos extremadamente elevados que no todas las empresas pueden sufragar. La novedad a partir de 1996 estriba en que existe ahora una red pública de computadoras conocida como Internet, que permite una drástica reducción en costos de comunicaciones para las empresas, y que, utilizada en forma protegida o privada (por medio esencialmente de claves de acceso restringidas), puede ampliar y mejorar notablemente las operaciones de las empresas a un costo mucho más accesible de lo requerido hace unos diez años. La tendencia de las operaciones en el siglo XX es hacia la globalización, o sea, hacia el manejo de operaciones comerciales en múltiples lugares intercomunicados por medio de redes de computadoras seguras, eficientes y confiables, pero a la vez, no muy costosas y de uso exclusivo para la organización considerada.

A este conjunto de redes es a lo que se le llama intranet. Representa un gran ahorro y una herramienta fundamental para las operaciones de las empresas. Facilita la consulta de información y mejora drásticamente la calidad y la velocidad de las comunicaciones privadas de las empresas. Además, les permite ser mucho más competitivas.

2.1.2.3 Las computadoras en las comunicaciones

Hasta cierto punto, los elementos tradicionales de comunicaciones para el intercambio de información, dígase teléfonos o faxes, ya han sido rebasados. Actualmente están siendo sustituidos por computadoras y por redes de comunicaciones que las enlazan. Las redes tradicionales de teléfonos están siendo sustituidas por redes de computación, en particular por una red que está abarcando el mundo entero denominada Internet.

Cuando hablamos por teléfono, aparentemente estamos conectando dos aparatos, pero en realidad se están entrelazando una infinidad de aparatos (teléfonos-subcentrales-centrales). ¿Qué pasa si sustituimos los aparatos telefónicos por computadoras?. El proceso resulta bastante similar al de los teléfonos. Lo importante es que el aparato que conteste sea de la misma naturaleza que el aparato que transmita información. Asi, un aparato telefónico sólo puede comunicarse con otro aparato telefónico; una máquina de Fax con otra máquina de Fax, una computadora con otra computadora, etc.

Si tratamos de comunicar un teléfono con un Fax, ¿qué pasa? - no se entienden porque no hablan el mismo idioma, o mejor dicho, porque no utilizan el mismo Protocolo.

Un **PROTOCOLO** es un conjunto de reglas para entablar una comunicación entre dispositivos electrónicos. Para intercomunicar dos computadoras es necesario contar con algo similar a una central. Es lo que en computación se llama un host, un ruteador o un servidor. Estas palabras denotan básicamente un equipo que controla las comunicaciones dentro de un área determinada.

NETWORK INFORMATION CENTER.- Centro de control de operaciones relativas al Internet que asigna la dirección de cada uno de los equipos conectados a la red.

En cada país existe una oficina dependiente del Network Information Center que controla las direcciones de los equipos en dicho país. Por ejemplo, en el caso de Ecuador, tenemos al Banco del Pacifico, el cual administra los dominios de tipo **.ec**

Como el recordar el número que le corresponde a cada equipo resulta algo tedioso, el sistema de control de usuarios ha sido modificado a modo de operar con una dirección más sencilla conocida como DOMINIO:

DOMINIO: Conjunto de datos que permite identificar cada equipo de cómputo conectado a la red de Internet en forma única.

En forma análoga a la de la Dirección Internet, el dominio de cada equipo consta de cuatro partes escritas en minúsculas y separadas por puntos, a saber:

máquina.empresa.tipo.país

Generalmente estas cuatro partes se leen de derecha a izquierda.

En ocasiones el orden de las mismas varía de un país a otro.

País - Es el país donde se encuentra el servidor. Generalmente lo conforman dos letras del nombre del país. Como se puede observar en la Tabla 1.1.

DOMINIO	PAIS
.at	Australia
.ca	Canada
.cl	Chile
.ec	Ecuador
.co	Colombia
.br	Brasil
.pe	Peru
.mx	México
(si se omite)	Corresponde a los USA

Tabla. 1.1

Tipo - Es el tipo de la empresa propietaria del servidor. Los tipos existentes son esencialmente los que se observan en la Tabla 1.2.

TIPO	DOMINIO	DESCRIPCION
Comercial	.com	Sitios de Negocios
Educación	.edu	Sitios educativos
Gobierno	.gov	Sitios de Gobiernos
Militar	.mil	Sitios Militares
Red	.net	Redes de Datos
Organicaciones	.org	ONGS

Tabla. 1.2

Empresa - Es el nombre del proveedor de servicios al cual está conectado el servidor.

Máquina - Es la máquina del proveedor del servicio de conexión (en caso de tener varias). Muchas veces se omite.

Ejemplos:

syr.edu corresponde a la Universidad de Syracuse en USA.

phys.uvic.ca corresponde a la facultad de física de la Universidad de Victoria en Canadá.

infosel.net.mx corresponde a la red de Infosel en México.

pucesa.edu.ec corresponde a la red de Pucesa que es de Educación en nuestro pais.

Como complemento a estos datos del servidor, es factible asociar un dato adicional del lado izquierdo del dominio (separado por arroba o @) para denotar una persona o un usuario conectado a la red en cuestión. Así pues,

pedro@dxx3.mx.inf.com

Permite identificar al usuario Pedro del equipo dxx3 de la empresa inf que es de comunicaciones y que está en México.

2.1.3.2 Exploradores de Internet

Los exploradores son simplemente herramientas que permiten el uso de direcciones específicas (conocidas como direcciones HTTP) y que facilitan la localización de imágenes. La ventaja de los exploradores modernos es que permiten buscar una imagen inicial y de allí obtener otras imágenes con textos, con gráficas, con animaciones, con videos o con sonidos.

El explorador más utilizado hoy en día es básicamente el Explorer de Microsoft. Sin embargo, cabe mencionar que existen otros exploradores tales como el Netscape y el Mosaic.

Características de un explorador Web .- Básicamente son las siguientes:

1. Una vez instalado, se arranca casi automáticamente.

2. Permiten el uso "bookmarks" o indicadores de aquellas direcciones de mayor interés para el usuario.

3. En efecto, una vez conocida una dirección de interés (por decir algo, la de la Microsoft - www.microsoft.com) para poder ir a ella otro día, es factible agregarla a una tabla de direcciones interesantes o bookmarks. Para ir a ella, basta con pedirle al explorador que muestre las bookmarks existentes, y seleccionar dentro de la lista la deseada en un momento dado.

4. Permiten el uso de herramientas adicionales asociadas con los exploradores para la localización de información cuando se desconoce la dirección del servidor dónde está almacenada (los buscadores del WEB cuya descripción aparece en la sección acerca del WEB).

2.1.3.3 Acceso al WEB

Los exploradores de Internet, son ayudas que permiten buscar en un gran número de servidores (desde uno hasta miles) un dato específico basándose en una dirección o dominio predefinido. Sin embargo, cuando el dominio es desconocido, es factible utilizar un buscador, o sea una herramienta para la localización de información en el conjunto de servidores interconectados al Internet y que constituyen lo que se conoce como la telaraña o WEB.

De los múltiples buscadores que hay en el mercado, cabe mencionar tres:

Lycos (cuya dirección es http://lycos.cs.cmu.edu)
Yahoo (cuya dirección es http://www.yahoo.com)
Open Text (cuya dirección es http://www.opentext.com:8080)

Sin embargo, existen muchos más.

Estos buscadores tienen las siguientes características:

1. Cuentan con un campo en el que es factible escribir algún dato o palabra a buscar en una infinidad de servidores en cuestión de segundos.

2. Cuentan también con una serie de páginas en las que aparecen diversos índices de temas (los cuales permiten entrar a su vez a otras páginas y así sucesivamente en función del interés de cada cual.

Por ejemplo, en Yahoo aparecen varios índices tales como el siguiente:

Literatura	Historia	Computación
Medicina	Juegos	etc.

Al seleccionar historia, por ejemplo, aparece otro índice en el que es factible escoger:

Universal	Por continentes	De personajes

etc.

Al seleccionar por continentes, Yahoo pregunta:

América	Africa	Europa

etc.

Al seleccionar América, aparece una lista de los países de América... y así sucesivamente hasta llegar al dato deseado.

Al hecho de estar pasando de un menú a otro, y de allí a otro, tantas veces como se desee, se le denomina navegar por Internet. Cuando la navegación es por páginas cuyo acceso es público, se dice que se navega en la telaraña o WEB.

2.1.3.4 EL WWW o telaraña a escala mundial

Evidentemente, al hablar de búsqueda en el WEB o conjunto de servidores de acceso público, es necesario definir el número de equipos en los cuales ha de ser hecha la búsqueda:

Lo ideal sería una búsqueda en la cual no importara la dirección del servidor. Tal búsqueda ya existe. Es lo más avanzado por ahora en el campo del Internet y se le conoce como dirección http, de (Hipertext Transfer Protocol).

El hipertexto: Las páginas del http son definidas por medio de un lenguaje especial denominado lenguaje HTML (Hypertext Markup Language). Dicho lenguaje permite el establecimiento de una relación entre un elemento de una imagen con otra imagen, independientemente de que estén ubicadas en el mismo servidor o en otro muy alejado.

En una imagen mostrada es factible brincar a otra imagen dando un clic en un punto específico diseñado para tal fin. Dicho punto es constituye un punto de enlace con otra imagen.

En una misma imagen pueden existir diversos puntos de enlace, cada uno de ellos a diferentes imágenes adicionales, por lo que al texto con una o varias palabras que permiten brincar a otras imágenes se le conoce como Hipertexto.

Por ejemplo, si busca la página de una empresa internacional, posiblemente le aparezca una imagen con el logotipo de la empresa y con un texto en el cual le sugiera la consulta de otras imágenes con información acerca de las ventas, de los productos, del valor de las acciones, etc. Al dar un clic en la palabra ventas, por decir algo, de inmediato le aparecerá otra imagen con un texto en el cual el autor pregunte si las ventas a consultar son las locales o a nivel mundial. Al dar un clic en la palabra locales, posiblemente aparezca otra imagen con datos estadísticos acerca de las ventas de la empresa. En este caso, las palabras ventas y locales habrán servido de enlace o de hipertexto.

Existen dos formas de distinguir los enlaces dentro de un hipertexto:

1. Ya sea porque aparecen en un color diferente del contexto (por ejemplo en azul siendo el resto del texto negro), o bien

2. Porque al pasar el apuntador del ratón o mouse cerca de ellas, dicho indicador o señalador cambia de forma, de una simple flechita a una mano con el dedo índice señalando la palabra en cuestión.

Por lo regular, el color del enlace cambia en caso de que sea accesado, o sea, de que sea vista la imagen ligada a él. Así, si en un principio aparece en azul, al visualizar la imagen ligada y volver al texto original, la palabra de enlace original cambia de color (por decir algo, de azul a violeta).

Sin entrar en detalles, cabe aclarar que existen diversos programas para hacer textos con hipertextos y con imágenes, entre los cuales destacan JAVA y HTML.

Independientemente de la imagen de que se trate, lo importante es que todas las imágenes o páginas ligadas entre si estén debidamente escritas en hipertexto y almacenadas en un servidor (puede ser en un mismo servidor, o en otro mucho muy lejano).

Para estar almacenada, cada imagen ha de tener una dirección HTTP (Hipertext Transfer Protocol). En este contexto es importante hacer un análisis de cómo se forma una dirección http. Dicha dirección es muy similar a la de una dirección Internet. Por lo regular consta de:

1. Un encabezado o indicador inicial, generalmente WWW.
2. Un nombre particular de la imagen. Generalmente las imágenes principales de cada grupo de imágenes ligadas (conocidas como Home Pages) constan del nombre de la compañía a la que pertenecen, o del producto.
3. Una clave del tipo de servidor al que pertenece la página en cuestión.
4. Los tipos existentes son esencialmente los indicados en la Tabla 1.2
5. Una clave del país donde se encuentra el servidor correspondiente, se puede observar en la **Tabla 1.2**

Por ejemplo, ¿cuál es la dirección http de la PUCESA?

Inicia con www; sigue con el nombre del servidor, y termina con el tipo de empresa y el país, así la direccion de la PUCESA es: www.pucesa.edu.ec

La imagen principal o Home Page de la PUCESA tiene esta dirección, pero las imágenes ligadas a esta home page suelen tener otras direcciones que inician con la misma dirección, pero con algún indicador adicional separado por diagonal. Por ejemplo: en el caso de la PUCESA, las imágenes ligadas a la primera tienen una dirección del tipo: www.pucesa.edu.ec/logo.gif

Nótese que no importa en qué servidor esté la imagen principal o cualquiera de las demás. Con conocer la dirección http de la imagen, la localización de las demás es automática.

Siguiendo con el ejemplo anterior, al buscar la página de la PUCESA, www.pucesa.edu.ec, al usuario no le importa si dicha página está en el servidor de La PUCESA de Ambato o en algun servidor de otra extension de la Universidad dentro del pais o fuera de ella. (Aunque lo más probable es que se ubique en Ecuador). Lo importante es el nombre de la página.

De esta manera, conociendo la dirección http de una de las imágenes es factible accesar cualquiera de las demás ligadas a dicha imagen inicial o home page.

2.1.4 Bases de Datos en Internet

Introducción

En las siguientes páginas, se detallara como acceder desde una página Web a una Base de Datos utilizando diferentes tecnologías. Algunas de ellas para Servidores Windows 9x, y otras para Unix y Linux..

2.1.4.1 Creación de las DB

El primer paso es crear las bases de datos que se utilizaran.
Se probara los ejemplos usando dos bases de datos relacionales en distinto formato para demostrar que no influye el tipo de servidor de bases de datos que se use, si se las conecta mediante el ODBC de Windows 9x. Las dos base de datos tendrán las mismas tablas que son las siguientes:

Tabla de clientes:

Nombre de la fila	Tipo de la fila
Nombre de la fila	Tipo de la fila
Dni	Texto (8), Primary Key
Nombre	Texto (20)
Apellido1	Texto (20)
Apellido2	Texto (20)

Tabla de proveedores:

Nombre de la fila	Tipo de la fila
Nombre de la fila	Tipo de la fila
Cif	Texto (9), Primary Key
Empresa	Texto (20)
NombreContacto	Texto (20)
Teléfono	Numérico

2.1.4.2 Obtención de un DSN

(DSN) Data Source Name. Es un identificador único de la base de datos en el sistema. Al definir un DSN hay que especificar tanto la ruta completa del archivo de base de datos como el controlador adecuado a la misma (MSAccess, FoxPro, DBase etc). Una vez creado, es todo lo que se necesita saber acerca de la base de datos para poder abrirla, consultarla, modificarla, etc.

Una vez que este preparado el archivo de bases de datos (por ejemplo ejemplo1.mdb) sé lo coloca en el directorio Data del dominio. Después en el panel de control del dominio sé obtendrá un DSN para dicho archivo. El DSN será el identificador que permitirá la conexión con la base de datos.

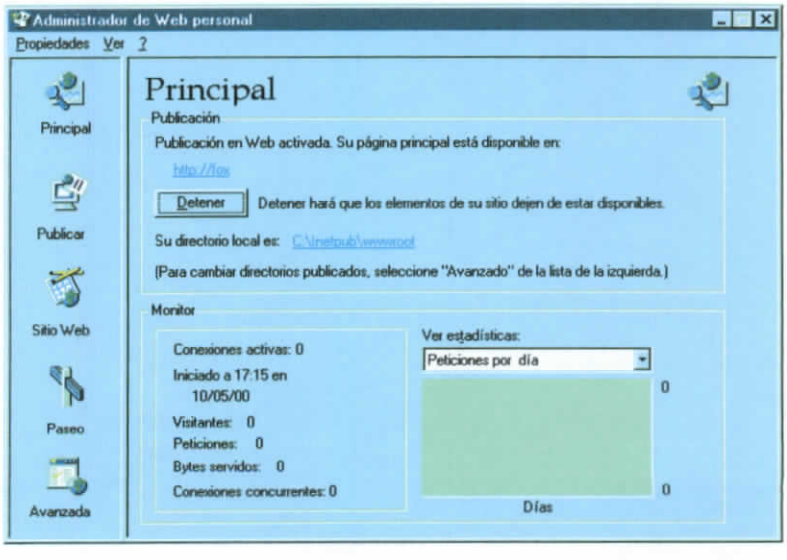
En la etiqueta **Data Source Name**, se introduce el nombre con el cual se llamara a la base de datos. Para el ejemplo se introducirá Ejemplo1. Finalmente, se pulsa el botón **Select** y se busca el nombre y path completo de la base de datos que se había creado (Data/Ejemplo1.mdb). De esta manera ya esta accesible la base de datos mediante ODBC.

2.1.43 Instalando Personal Web Server y Sambar Server

Personal Web Server es uno de los servidores WEB de Microsoft. Viene en el CD de Windows 98 aunque es necesario instalarlo después de la instalación de Windows. Para su instalación se siguen los siguientes pasos:

- Insertar el disco compacto de Windows 98 en la unidad de CD
- Escoger el botón Inicio y después Ejecutar.
- En Abrir, escribir: x:\add-ons\pws\setup.exe (o instalar.exe) donde x es la letra de la unidad de CD-ROM.
- Escoger Aceptar.
- Seguir las instrucciones de Personal Web Server Setup.

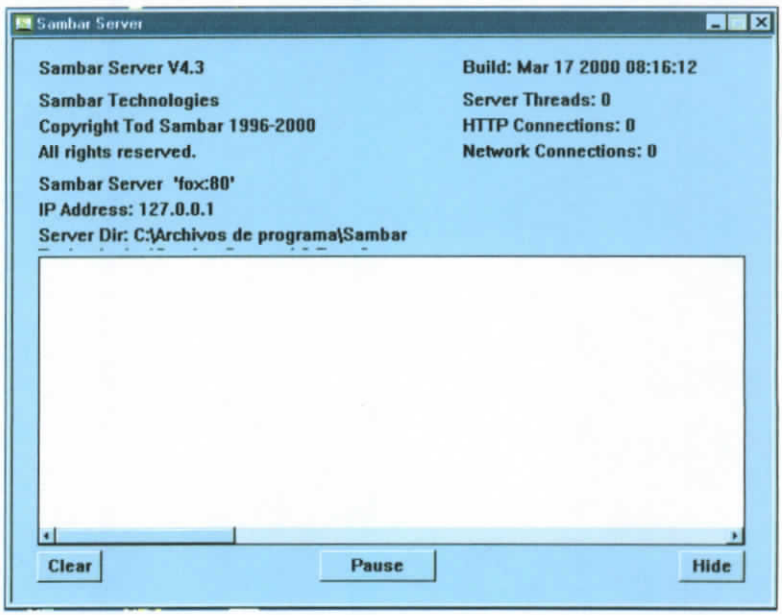
Al reiniciar el equipo aparecerá en la barra de tareas el icono de PWS. Si se lo abre aparecerá una pantalla parecida a esta:



Con la opción Avanzada se puede crear directorios y darles los permisos de ejecución para poder guardar en él los scripts o bien utilizar el directorio que él crea por defecto.

Para ver las diferencias de las bases de datos se procederá a instalar otro servidor este es el Sambar web Server que se lo puede bajar gratuitamente desde http://www.winfiles.com/.

Una vez que grabado el fichero .zip de Sambar server se descomprime en un directorio tmp y se ejecuta el setup.exe, Al ejecutar el instalador se debe observar una pantalla como la siguiente:



2.1.4.4 Instalando Perl, JSP y PHP3

Ahora se indicara como instalar los lenguajes que va a utilizar y configurar el servidor para su uso.

Para ASP como es de Microsoft y el servidor también, no tiene que hacer nada. Para C tampoco porque el CGI en C va a ser un ejecutable compilado y no se necesitara el compilador en el servidor.

PHP3

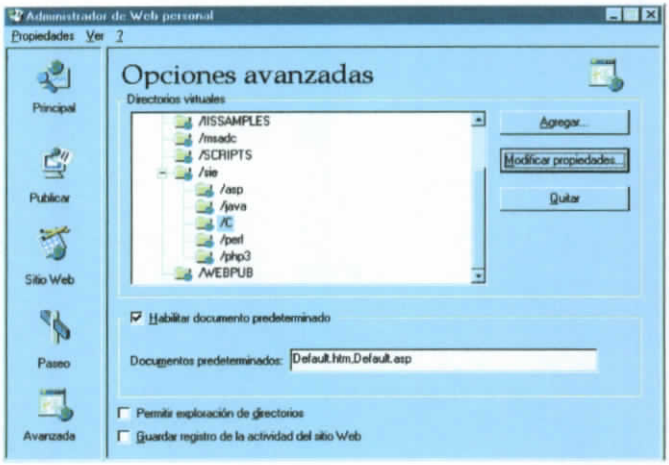
- **PASO1**: se descomprime el fichero **php-3_x_x-win32.zip** bajado de http://www.php.net/ en un directorio. Por ejemplo "C:\PHP3\".
- **PASO 2**: Se copia el fichero **'php3.ini.dist'** al directorio **%WINDOWS%** y sé renombra por el nombre de **'php3.ini'**

- **PASO 3**: Se edita el fichero **php3.ini** y se busca 'extension_dir' que debe cambiarse para que apunte al directorio donde se ha descomprimido la instalación de php3. Se elige los módulos (dll) que se quiere cargar cuando php se ejecute. Esto se hace quitando el guión de comentario(') que tienen las líneas del estilo "extension=php3_*.dll". Para el ejemplo basta quitar el comentario de la línea "extension=php3_odbc.dll". Si se desea más información se la puede encontrar en la FAQ de http://www.php.net/ A continuación se busca la linea browscap y debe ser igual a "browscap = c:\WINDOWS\SYSTEM\inetsrv\browscap.ini".
- **PASO 4**: Edita el registro de sistema (Es recomendable realizar una copia de seguridad), Luego se busca HKEY_LOCAL_MACHINE/System /CurrentControlSet/Services/W3Svc/Parameters/ScriptMap

1. En el menú Edición, pulsar Nuevo ->Valor de la cadena.
2. En Nombre se escribe las extensiones asociadas a los scripts php. Por ejemplo '.php3'
3. Ingresar en el nuevo valor el path del ejecutable php.exe, por ejemplo: 'c:\php3\php.exe %s %s'. Los '%s %s son muy importantes, sin ellos php no funcionaria.
4. Repetir 1) - 4) para cada extensión que se quiera asociar con PHP. Los más comunes son .php3 y .phtml
5. En HKEY_CLASSES_ROOT, pulsar Edición y luego Nuevo ->Clave
6. El nombre de la clave es la extensión que se quiere asociar con php ('.php3')
7. Digitar la nueva clave y en el panel derecho dar doble click en el valor por defecto y escribir 'phpfile'
8. Repetir 5) - 8) para cada extensión que se quieras asociar con php.
9. En Edición, pulsar Nuevo ->Clave. El nombre de la clave debe ser 'phpfile'
10. Digitar la nueva clave y en el panel derecho dar doble click en el valor por defecto y escribir 'PHP Script'
11. Crear una nueva clave hija de 'phpfile' y llámarla 'Shell'
12. Crear una nueva clave hija de 'Shell' y llámarla 'open'
13. Crear una nueva clave hija de 'open' y llámarla 'command'
14. Digitar la clave command y en el panel derecho dar doble click en el valor por defecto y escribir: 'c:\php3\php.exe -q %1'
15. Salir del Regedit

El directorio cgi-bin

Sin depender del servidor se necesita que los script y los cgi estén en un directorio que tenga permiso de ejecución. Lo más frecuente es que este directorio se llame cgi-bin. (otros como Apache o Sambar no permiten otro directorio que no sea cgi-bin). PWS crea el directorio /SCRIPTS. Particularmente, se puede crear en el servidor varios directorios (uno para cada tecnología) y se les da permiso de ejecución. Esto se realiza con el PWS en la opción AVANZADAS. Y se observara la ventana siguiente:



Ahora se observara cómo se conecta con la base de datos y como se trabaja con ella desde las distintas tecnologías que se han mencionando.

2.1.4.5 Tecnología ASP

Introducción

El sistema de acceso a bases datos a través de la web utilizando la tecnología Microsoft, se denomina ADO (ActiveX Data Objects). Aquí se expone solo una introducción. Para ver una completa referencia acerca de todo el sistema de bases de datos de Microsoft, se puede consultar en www.microsoft.com/data.. La referencia completa de métodos y propiedades de los objetos está también en la biblioteca online de MSDN (msdn.microsoft.com). Ahí se encontrara múltiples ejemplos en los que comprobará la flexibilidad del sistema ADO (al mismo resultado se puede llegar por vías distintas). Por motivos pedagógicos, aquí se expone una utilización más rígida.

Terminología.- ADO: ActiveX Data Objects. Es una familia de objetos componentes dedicados a facilitar el acceso a bases de datos. El ProgID de cada uno de ellos se forma combinando ADODB. con el nombre del objeto (por ejemplo ADODB.Recordset, ADODB.Connection, ADODB.Command, etc). Por tanto, en VBScript los objetos se crean con sentencias tipo Set mirst = Server.CreateObject("ADODB.Recordset"), etc

Los objetos componentes de ADO

Todo el sistema ADO se basa en una serie de objetos cuyas propiedades y métodos hay que utilizar. Estos objetos están registrados en el sistema, sin embargo no están predefinidos. Es decir, hay que crearlos utilizando Server.CreateObject.

Como valores de muchas de las propiedades de los objetos ADO se utilizan constantes, como adOpenForwardOnly, adOpenKeySet, etc. Todas estas constantes empiezan por ad y están definidas en un archivo llamado adovbs.inc para Visual Basic script y adojavas.inc para JavaScript. Estos archivos se los puede encontrar en su instalación local de ADO. Para incluirlos en las páginas ASP es necesario utilizar un código de este tipo al comienzo de cada página ASP que use las constantes: <!--#INCLUDE FILE="adovbs.inc"-->

Para que esta inclusión funcione el archivo adovbs.inc debe estar en el mismo directorio que la página ASP que lo incluye.

Connection.- Representa una conexión a una base de datos. Este es el primer objeto que se debe crear para poder conectar con la base de datos. Tanto el objeto ADO que permite acceder a los datos (Recordset) como el que permite realizar consultas (Command) disponen de una propiedad llamada ActiveConnection que es una referencia al objeto connection que enlaza con la base de datos a la que queremos acceder.

Propiedades y métodos más relevantes

ConnectionString.- Es una cadena de caracteres con la información necesaria para establecer una conexión con la fuente de datos. Existen 7 argumentos distintos que se pueden suministrar en esta cadena, los básicos son el DSN que identifica al archivo de base de datos y el login y password si existen. Los argumentos se separan con punto y coma.

Ejemplo: "DSN=midsn; UID=milogin; PWD=micontraseña"

Open.- Abre la conexión con la base de datos. Si antes se ha asignado la propiedad ConnectionString, este método no necesita parámetros.

Close.- Cierra la conexión con la base de datos.

Ejemplo de uso

```
<%
    Set miconexion = Server.CreateObject("ADODB.Connection")
    miconexion.ConnectionString = "DSN=midsn"
    miconexion.Open
    ' .......
    '........
    miconexion.Close
%>
```

Recordset.- Este es el objeto ADO más importante, ya que permite acceder directamente a los datos de las tablas, tanto para leerlos como para modificarlos.

Un objeto Recordset representa una tabla, que puede ser una tabla física de la base de datos o bien una obtenida mediante una operación como un filtrado o sentencia SQL. En cualquier caso, el objeto representa a la tabla con todos sus registros, aunque sólo uno de ellos es el activo. El registro activo es en el que podemos leer o modificar los valores de los campos. También se le llama cursor.

Propiedades y métodos más importantes

Para una mejor compresión y puesto que son numerosos, estos se han dividido por categorías de utilidad.

Propiedades que hacen referencia al origen de los datos:

Estas dos propiedades deben ser asignadas a todo Recordset, pues le dicen la base de datos y la tabla de la que obtener sus datos.

ActiveConnection.- Como se ha comentado antes a esta propiedad se le debe asignar un objeto connection que se haya creado previamente. Indicará al Recordset la base de datos en la que buscar su tabla.

Source.- Indica al objeto Recordset la tabla a la que representará. A la propiedad Source se le asigna normalmente una cadena de caracteres con el nombre de la tabla. Sin embargo, también es posible asignarle una sentencia SQL y entonces el objeto Recordset referenciará al resultado de aplicar dicha sentencia.

LockType.- Indica el tipo de bloqueo que se realiza sobre la base de datos. Por defecto toma el valor adLockReadOnly que como su nombre indica sólo sirve para leer datos. Si deseamos editar o añadir registros, tendremos que cambiar esta propiedad por otro valor. El más usado es adLockOptimistic que permite editar la base de datos o añadir registros realizando un bloqueo optimista (sólo cuando sea estrictamente necesario).

CursorType.- El tipo de cursor que se utiliza para recorrer los registros de un recordset. Por defecto toma el valor adOpenForwardOnly que como su nombre indica sólo permite moverse hacia adelante. Por ello si queremos utilizar libremente todos los métodos de movimiento de un recordset (MoveFirst, MoveTo, MoveNext, MovePrevious, etc) tendremos que cambiar el cursor por uno más potente. Esto puede hacerse asignando a esta propiedad el valor adOpenKeyset

Ejemplo:

Se crea en primer lugar un objeto Connection y luego un Recordset al que se asigna dicho objeto. Se debe realizar algo parecido para abrir el Recordset.
(NOTA: Si hay que añadir registros es necesario cambiar el LockType antes de inicializar el recordset).

```
<%  Set miconexion = Server.CreateObject("ADODB.Connection")
    miconexion.ConnectionString = "DSN midsn"
    miconexion.Open
    Set mirecordset = Server.CreateObject("ADODB.Recordset")
    mirecordset.ActiveConnection = miconexion
    mirecordset.Source = "Clientes"
    mirecordset.Open
    ........ ' Operaciones con los datos
    ........' de la tabla Clientes.
    mirecordset.Close
    miconexion.Close
%>
```

Propiedades que hacen referencia al número de registros:

RecordCount.- Número de registros de la tabla a la que representa el objeto recordset.

Ejemplo:

```
<h3>
Tenemos <%= rstClientes.RecordCount%>
clientes registrados en nuestra base de datos
</h3>
```

Métodos para mover el cursor (registro activo):

Estos métodos pueden funcionar todos o no dependiendo del tipo de cursor creado. El tipo de cursor se asigna en la propiedad CursorType. Por ejemplo, para asignar un cursor adOpenKeySet que permita moverse hacia adelante o hacia atrás:

```
<%
    Set miconexion = Server.CreateObject("ADODB.Connection")
    miconexion.ConnectionString = "DSN=midsn"
    miconexion.Open
    Set mirecordset = Server.CreateObject("ADODB.Recordset")
    mirecordset.ActiveConnection = miconexion
    mirecordset.Source = "Clientes"
    mirecordset.CursorType = adOpenKeySet
    mirecordset.Open
    ........ ' Operaciones con los datos
    ........' de la tabla Clientes.
    mirecordset.Close
    miconexion.Close
%>
```

Nótese que la asignación de propiedades al recordset debe hacerse antes de invocar el método open que es el que lo inicializa con registros.

MoveFirst.- Mueve el cursor al primer registro de la tabla

MoveLast.- Mueve el cursor al último registro de la tabla

MoveNext.- Mueve el cursor al siguiente registro.

MovePrevious.- Mueve el cursor al registro anterior.

Ejemplo de un bucle que recorre todos los registros de una tabla

```
<%
    mirecordset.MoveFirst
    do while not mirecordset.EOF
    ..... ' Tratamiento
    ..... ' de datos
     mirecordset.MoveNext
    loop
%>
```

Lectura y modificación de los campos del registro activo

La sintaxis para acceder a los datos de un campo del registro activo de un recordset es: mirecordset("Domicilio"). Esto se usa tanto para leer como asignar valores. En nuestro ejemplo, el objeto mirecordset representa a una tabla uno de cuyos campos tiene el nombre "Domicilio". Así, con la expresión dom = mirecordset("Domicilio")

Leemos el valor del campo domicilio del registro activo y se lo asigna a la variable dom. Con la expresión: mirecordset("Domicilio") = "Av. Cevallos y Mera No. 78" mirecordset.Update asignamos un valor al campo Domicilio del registro activo. Tras la edición del registro, es necesario llamar al método Update. El motivo es que los cambios en el registro activo se realizan sobre un buffer y no sobre el registro propiamente dicho.

Ejemplo:

El recordset rstClientes representa a la tabla de clientes, que entre otros, tiene los campos Provincia e IVA . El siguiente bucle recorre todos los clientes, comprueba su provincia y le asigna el IVA correspondiente: 0 para Tungurahua y 12 para los demás

```
<%
    rstClientes.MoveFirst
    do while not rstClientes.EOF
        if rstClientes("Provincia") =
          "Tungurahua"
        Then
            rstClientes("IVA") = 0
        else
            rstClientes("IVA") = 12
        end if
        rstClientes.Update
        rstClientes.MoveNext
    loop
%>
```

Métodos para agregar o eliminar registros de la tabla

Delete.- Eliminar el registro activo es muy fácil. Basta con invocar este método. Por ejemplo, este bucle elimina todos los clientes morosos de nuestra base de datos:

```
<%  rstClientes.MoveFirst
    do while not rstClientes.EOF
        if rstClientes("Deuda") > 0 Then
            rstClientes.Delete
        end if
    rstClientes.MoveNext
    loop
%>
```

AddNew y Update.- Crear un nuevo registro involucra dos métodos: Primero AddNew crea un nuevo registro en blanco. Después se asigna valores a los distintos campos del registro. Por último se invoca al método Update para que se actualize el nuevo registro con los valores asignados.

En este ejemplo se crea un nuevo cliente a la base de datos

```
<%
    rstClientes.AddNew
    rstClientes("Nombre") = "Favian Toasa"
    rstClientes("Direccion") = "AV. Cevallos y Mera No. 78"
    rstClientes("Localidad") = "Ambato"
    rstClientes("Profesion") = "Ingeniero Sistemas"
    rsClientes.Update
%>
```

Se asigna un tipo de bloqueo que permita la edición de la base de datos. Asignando a la propiedad LockType del recordset uno de estos valores: adLockOptimistic, adLockPessimistic o adLockBatchOptimistic.

Command.- Este objeto es la representación de un comando que se envía a la base de datos. Este comando puede realizar una de estas 3 caracteristicas;

- Llamada a un procedimiento guardado en la base de datos.
- El texto de una sentencia SQL.
- El nombre de una tabla.

ActiveConnection.- Es una referencia al objeto connection que enlaza con la base de datos. es imprescindible asignar esta propiedad antes de invocar el método command.execute para ejecutar el comando.

CommandText.- Este es el texto del comando. Si se trata de una consulta SQL (lo habitual), esta propiedad es simplemente una cadena con el texto de la consulta.

Execute.- El método que ejecuta la consulta. El resultado de la ejecución del comando normalmente será un recordset.

En el ejemplo, se obtiene un recorset con un filtrado de la tabla Clientes en el que se toma sólo aquellos clientes cuya provincia es Pichincha.

```
<%  Set miconexion = Server.CreateObject("ADODB.Connection")
    miconexion.ConnectionString = "DSN=midsn"
    miconexion.Open
    Set SqlCommand = Server.CreateObject("ADODB.Command")
    SqlCommand.ActiveConnection = miconexion
    SqlCommand.CommandText =
          "SELECT * FROM Clientes WHERE Provincia = 'Pichincha'"
    Set rstNavarros = SqlCommand.execute
    ' .......
    '........
    miconexion.Close
%>
```

2.1.4.6 Tecnología PHP

Introducción

El PHP3 es otro lenguaje de script que permite programar aplicaciones que se ejecutarán en el servidor. Es uno de los lenguajes más utilizados en el mundo Unix con el servidor Apache aunque también se puede ejecutar bajo Windows. Una vez instalado PHP3 como se trato anteriormente. Se utilizara la base de datos en formato Access con el servidor Personal Web Server.

El Lenguaje PHP3

Lo primero que se indica al browser es que se va a ejecutar código PHP3. Esto se hace con la directiva <?php para que sepa cómo interpretarlo. Se trabaja con objetos.

Primero creamos el objeto conexión:

```
$cnx = odbc_connect(  'Ejemplo1' ,  'Alejandro',  'pasword' );
    if (!$cnx) {
        Error_handler(  "Error al conectar odbc" , $cnx );
    }
```

Después se definira la sentencia sql:

```
$SQL_Exec_String =  "select * from Clientes";
```

Se ejecuta:

```
$cur= odbc_exec( $cnx, $SQL_Exec_String );
    if (!$cur) {
        Error_handler(  "Error en odbc_exec( ) " , $cnx );
    }
```

Se obtienen los resultados en un objeto cursor y se los envía al cliente en una tabla.

```
echo "<table border=1><tr><th>Dni</th><th>Nombre</th>".
     "<th>Apellido1</th><th>Apellido2</th></tr>\n";
    $nbrow=0;    //Variable local para contar el n° de filas
    while( odbc_fetch_row( $cur ) ) {
      $nbrow++;
      $Dni= odbc_result( $cur, 1 );
      $Nombre= odbc_result( $cur, 2 );
      $Apellido1= odbc_result( $cur, 3 );
      $Apellido2= odbc_result( $cur, 4 );
      echo  "<tr><td>$Dni</td><td>$Nombre</td>".
            "<td>$Apellido1</td><td>$Apellido2</td></tr>\n";
    }
   echo   "<tr><td   colspan=2>$nbrow   entradas   en   la   bd
</td></tr></table>";
```

Se cierra la conexión.

```
odbc_close( $cnx);
```

Insertar, borrar.- Para las demás opciones de la base de datos (insertar, Borrar, modificar ...) se tiene que modificar la sentencia sql y volver a realizar un select * para devolver la nueva tabla al usuario.

Ejemplo:

```
function Enter_New_Entry($Dni,$Nombre,$Apellido1,$Apellido2) {

    $cnx = odbc_connect(  'Ejemplo1' ,  'Alejandro',  'Password'
);
    if (!$cnx) {
        Error_handler(  "Error en odbc_connect" , $cnx );
    }
    $SQL_Exec_String =
           "Insert  Into  Clientes  (dni,  Nombre,  Apellido1,
Apellido2)
           Values     ('$Dni',     '$Nombre',     '$Apellido1',
'$Apellido2')";

    $cur= odbc_exec( $cnx, $SQL_Exec_String );
    if (!$cur) {
        Error_handler(  "Error en odbc_exec(  ) " , $cnx );
    }
    odbc_close( $cnx);
}
```

2.2 Lenguajes para el diseño de páginas WEB

2.2.1. Lenguaje HTML

2.2.2. Lenguaje JAVASCRIPT

2.2.1 Lenguaje HTML

A las instrucciones que forman el lenguaje HTML se les llama elementos. La notación de los elementos consiste en los símbolos < y > que encierran dentro una instrucción.

Los elementos pueden ser llenos o vacíos.

Elementos llenos.- Se forman escribiendo la instrucción correspondiente seguida del texto al que se quiere aplicar la instrucción y se termina repitiendo la instrucción pero con una barra inclinada inmediatamente antes de la misma. Por ejemplo, el elemento H1 que sirve para dar el máximo tamaño al texto, se escribirá:

<H1> Texto de prueba </H1>

Si se omite </H1> todo el resto de la página tendrá el mismo tamaño grande.

Elementos vacíos.- Los elementos vacíos se escriben como los llenos, pero no es necesario poner la instrucción repetida al final con una barra. Esto se debe a que estos elementos no producen un efecto sobre el texto. Generalmente se utilizan para separar bloques de texto, y por tanto no es necesario indicar su fin. Empiezan y terminan en el mismo punto.

Por ejemplo, el elemento <HR> que sirve para dibujar una línea horizontal en la pantalla, se escribirá:

<HR>

Elementos con argumento.- Algunos elementos se escriben con argumento. Es como pasarle parámetros a la instrucción, y se llaman atributos del elemento.

Por ejemplo, el elemento <A> que sirve para hacer un link (enlace) con otro documento, se escribirá:

<A HREF="http://www.miservidor.com/mifichero.htm"> Link de prueba</A>

Este es un elemento lleno donde al atributo HREF se le asigna el valor que aparece entre comillas "http://www.miservidor.com/mifichero.htm".

Si el fichero procede de un servidor http remoto, (es decir un servidor WWW; http es el nombre del servicio invocado), éste se encargará de decirle al visualizador cuanto necesite saber, pero si lo que hay que procesar es un fichero local, la única forma de saberlo es por la extensión del fichero.

La extensión de un fichero son las tres o cuatro letras (depende del sistema operativo con el que se trabaje) que hay después del nombre del fichero y que están separadas del mismo por un punto. Por ejemplo, esta página se llama estruc.htm. Las tres letras que hay después del punto (htm) son la extensión del fichero. Las otras, las que están delante del punto, son el nombre del fichero.

Hay que tener presente que las extensiones de 4 o más letras son propias de sistemas operativos como Unix, Macintosh o Windows 95 y las de 3 o menos del DOS.

Estas son alguna de las extensiones standard más comunes y sus contenidos:

.html o .htm, Documento HTML Contiene texto e instrucciones HTML que serán interpretadas.

.text o .txt, Contiene texto plano. El visualizador presentará todo el fichero como si fuera un único bloque de texto y no interpretara ninguna instrucción HTML que pudiera llevar. Lo mismo hará con cualquier documento que lleve una extensión desconocida.

.gif, Contiene un fichero de imagen de formato GIF

.xbm, Contiene un fichero de imagen en formato (blanco y negro)

.xpm, Contiene un fichero de imagen en formato X-Bitmap (color)

.jpeg o jpg, Contiene un fichero de imagen en formato codificado jpeg.

.mpeg o mpg, Contiene un fichero de imagen o vídeo o cine (en movimiento)

.au, Contiene un fichero de audio (sonido) codificado en aiff-encoded.

.mid, Fichero de audio (sonido) generado con secuenciadores midi

.avi, Contiene un fichero con vídeo y sonido típico de Windows.

.exe, Contiene un fichero binario ejecutable en DOS

.hqx, Contiene un fichero ejecutable Macintosh.

.Z o .zip, Contiene uno o varios fichero comprimidos de cualquier clase.

2.2.2 Lenguaje JAVASCRIPT

El JavaScript ha evolucionado en conjunto con las versiones de navegadores. El lenguaje ha ido avanzando durante sus años de vida e incrementando sus capacidades. En un principio podía realizar muchas cosas en la página Web, pero tenía pocas instrucciones para crear efectos especiales. Las distintas versiones de JavaScript son:

JavaScript 1.- Nació con el Netscape 2.0 y soportaba gran cantidad de instrucciones y funciones, casi todas las que existen ahora ya se introdujeron en la primera versión.

JavaScript 1.1.- Versión que se diseño con la llegada de los navegadores 3.0. Implementaba poco más que su anterior versión, como por ejemplo el tratamiento de imágenes dinámicamente y la creación de arrays.

JavaScript 1.2.- La versión de los navegadores 4.0. Esta tiene como desventaja que es un poco distinta en plataformas Microsoft y Netscape, ya que ambos navegadores crecieron de distinto modo y estaban en plena lucha por el mercado.

JavaScript 1.3.- Los navegadores 5.0 en adelante la poseen. En esta versión se han limitado alguna diferencias y asperezas entre los dos navegadores. Las diferentes versiones de JavaScript han sido finalmente integradas en un estándar denominado ECMAScript-262. Dicho estándar ha sido realizado por la organización ECMA dedicada a la estandarización de información y sistemas de comunicación.

Para programar con JavaScript necesitamos básicamente lo mismo que para programar páginas Web con HTML. Un editor de textos y un navegador compatible con JavaScript.

Sintaxis.- La sintaxis de JavaScript es prácticamente idéntica a la de los lenguajes orientados a objeto scomo el Java y el Lenguaje C.

JavaScript, es Case Sensitive, termino inglés utilizado para expresar que un lenguaje de programación es capaz de distinguir entre letras mayúsculas y minúsculas, y que ni solo es capaz de distinguirlas, sino que se comporta de manera estricta a la hora de ejecutarlas.

Componentes de JavaScript.- Todas las páginas tienen los siguientes objetos:

Navigator.- Tiene propiedades para el nombre, version del navegador usado, etc.

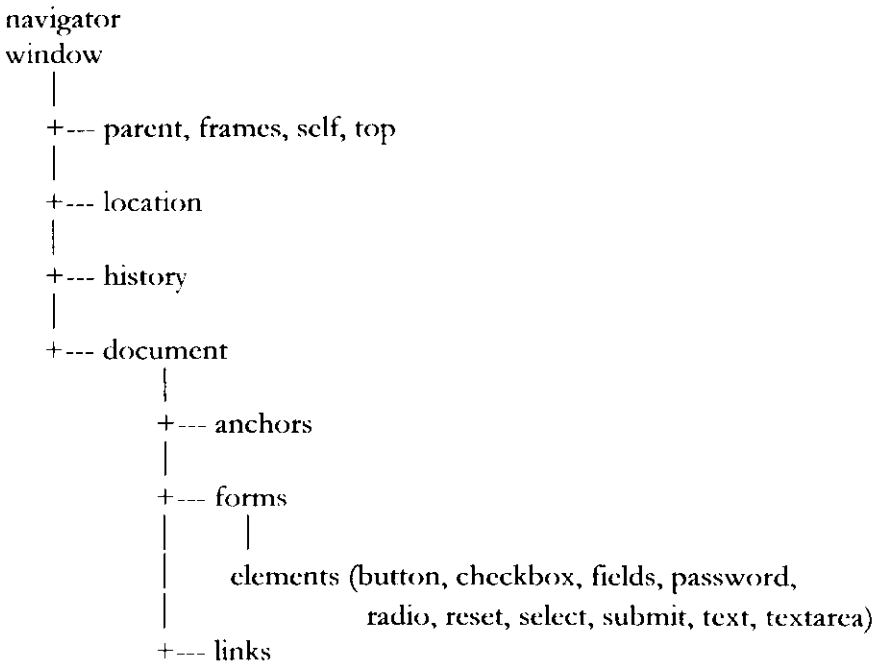
Window.- El objeto de nivel más alto, tiene propiedades que se aplican a la ventana entera. También existe un objeto window por cada "ventana hijo" en los frames de un documento.

Document.- Contiene propiedades sobre el contenido del documento, (title, background, color, links y forms).

Location.- Contiene propiedades de la URL

History.- Contiene propiedades de las URLs referenciadas anteriormente.

Jerarquía de los objetos:



Trabajo con funciones en JavaScript.- Las funciones suelen introducirse entre los Tags <HEAD> </HEAD> de HTML y tienen la siguiente sintaxis:

```
//Definición de una función
function nombre_función()
{ CODIGO DE FUNCION
  return (valor de retorno);
}
```

Para realizar el llamado a una función se procede de la siguiente manera:

```
<SCRIPT>
    document.write("El valor retornado por la función es:",función(5),".")
</SCRIPT>
```

2.3 Diseño de contenido multimedia

2.3.1 Procesamiento de Imágenes con Fireworks

2.3.2 Desarrollo de Animaciones con Flash

2.3.3 Creaciones de elementos Web con Xara WebStyle

2.3.1 Procesamiento de imágenes con Fireworks

Fireworks, es una herramienta que nos permite editar las imágenes de sitios web, y nos permite importarlas en varios formatos como pueden ser: GIF, JPEG, GIF ANIMADOS, así como también la creación de Rollovers, Menús desplegables, definir Mapas y Zonas hipervinculadas.

2.3.2 Desarrollo de animaciones con Flash

Flash es el software de creación más avanzado para crear animación interactiva escalable para el Web Tanto si se desea crear logotipos animados, controles de navegación de sitios Web, animaciones de gran formato o sitios Web completos de Flash, la capacidad y flexibilidad de Flash es el medio ideal para desarrollar sitios Web interactivos con gran contenido multimedia.

.

2.3.3 Creación de elementos con Xara Web Style

Xara Web Style, es un programa para la creación de contenido Web, que contiene extensas librerías de Botones, Separadores de líneas horizontales y verticales, Barras de Navegación, Fuentes de Texto artísticas, Texturas optimizadas para sitios Web, elementos 3D. Todos estos elementos son configurables de acuerdo a las necesidades del sitios Web y de sus diseñadores.

CAPITULO III

3. REALIZACION DEL PROYECTO

3.1 Análisis de Diseño

3.1.1 Software existente en los Servidores WEB

La Pontificia Universidad Católica Sede-Ambato, cuenta de acuerdo a la ley, con las licencias de los siguientes productos informáticos mostrados en la **Tabla 1.3**

SOFTWARE EXISTENTE EN LOS SERVIDORES WEB DE LA PUCESA

CATEGORÍA	PAQUETE	Ver.	FABRICANTE	CANT.
Sistemas Operativos	Windows NT	4.0	Microsoft	1
	Linux Red Hat	7.0	GNU	1
	Sun Solaris	2.6	Sun Micro Systems	1
Herramientas de Desarrollo	Oracle	8.1	Oracle Corp.	1
	Power Builder	5.0	Power Soft	1
	Visual Studio	6.0	Microsoft	1
	Visual C++	6.0	Microsoft	1
	Visual J++			1
	PHP	4.0	GNU	
Bases de Datos	SQL Server	7.0	Microsoft	10
	DB2	8.0	IBM	1
	MYSQL	2.0	GNU	1
Diseño Gráfico	Flash	4.0	Macromedia	Share
	Fireworks	5.0	Macromedia	Share
Diseño de Páginas WEB	Front Page	98	Microsoft	1
	DreamWeaver	MMX	Macromedia	Share
	Ultradev			

Tabla 1.3

3.1.2 Hardware existente en los Servidores WEB

Los equipos que realizan la función de Servidores Web de la PUCESA son los detallados en la Tabla **1.4**

HARDWARE EXISTENTE EN LOS SERVIDORES WEB DE LA PUCESA

EQUIPO	MARCA	CARACTERISTICAS
Computadora	SUN	SUN Microsystem 250 e Procesador Ultra Spark 400 Mhz, 256 RAM, CD-ROM, 180 Gb de Disco Duro, Tape Backup SCSI.
Computadora	CLON	AMD 1.26 Mhz, 256 RAM, D.D. de 30 GB, CD-ROM, Multimedia, Lan, Video, USB.
SWITCH	CNET	Base 100 T, Fast Ethernet Port.
HUB	CNET	Base 100 T, 24 Ports.

Tabla 1.4

En la Fig.# 1.2 se detalla la distribución del hardware del Centro de Cómputo de la Universidad.

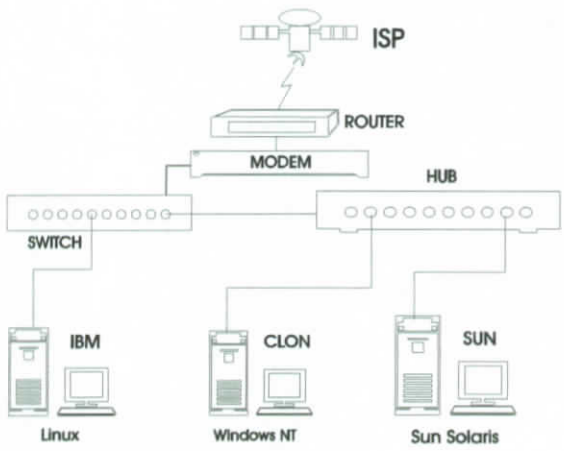


Fig. # 1.2

3.2 Bases de Datos MySQL en servidores Web UNIX

Para la utilización de Bases de Datos en Internet, existe actualmente el modelo **PHP-MySQL,** que junto con **Apache** se constituyen en la solución "libre" para desarrollar sitios Web Dinámicos de alta calidad, actualmente es una de las opciones mas utilizadas.

Instalación.- Para la obtención del Software se accede a las siguientes direcciones:

http://www.apache.org/ para obtener la versión de apache para Unix, en este caso el archivo es: **apache-1.3.x.tar.gz**

http://www.mysql.org/ para grabar el software; para la Base de Datos, que se encuentra empaquetado en el fichero **mysql-3_22_22_tar.gz**.

http://www.php.net/ y adquirir el lenguaje de programación PHP, que estara comprimido en el archivo php-3.0.x.tar-gz.

Para la instalación se debe ingresar como **root** en la máquina donde se va ha instalar.

Crear un directorio de instalación, en este caso podria ser /usr/local.

Una vez grabado el software de instalación en el directorio /root/install, se lo descomprime:

```
cd /usr/local
tar zxvf /root/install/apache-1.3.x.tar.gz
tar zxvf /root/instal/mysql-3.22.x.tar.gz
toot/instal/mysql-3.22.x.tar.gz
tar zxvf /root/instal/php-3.0.x.tar-gz
```

Crear los enlaces con el comando ln:

```
ln -s /usr/local/apache-1.3.x /usr/local/apache
ln -s /usr/local/mysql-3.22.x /usr/local/mysql
ln -s /usr/local/php-3.0.x /usr/local/php
```

Preparar las fuentes para la compilación de Apache:

```
cd /usr/loca/apache
./configure --prefix=/usr/local/apache
```

Compilar e instalar MySQL:

```
cd /usr/local/mysql
./configure --witout-debug --prefix=/usr/local/mysql
make
make install
cp /usr/local/support-files/mysql.server /etc/rc.d/init.d/mysql
chmod 755 /etc/rc.d/init.d/mysql
```

Crear la base de datos del sistema MySQL

```
/usr/local/mysql/bin/mysql_install_db
```

Arrancar el servidor MySQL:

```
/etc/rc.d/init.d/mysql     start</CODIGO"/etc/rc.d/init.d/mysql
start
```

Asignar el password del administrador (root) de MySQL

```
/usr/local/mysql/bin/mysqladmin -u root password "clave"
```

Y compilar PHP como módulo de Apache.

```
cd /usr/local/php
./configure --with-mysql=/usr/local/mysql \
--with-apache=/usr/local/apache \
--enable-track-vars
make
make install
#cp php3.ini-dist /usr/local/lib/php3.ini
```

Compilar Apache

```
cd /usr/local/apache
./configure --prefix=/usr/local/apache \
--activate-module=src/modules/php3/libphp3.a
# si hemos compilado PHP4 utilizaremos
#--activate-module=src/modules/php4/libphp4.a
# quitar los comentarios para habilitar el módulo de proxy
#--activate-module=src/modules/proxy/libproxy.a
make
make install
```

Definir las extensiones de los scripts PHP, añadiendo las siguientes líneas en el fichero de configuración de apache (httpd.conf):

```
AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3 .php
AddType application/x-httpd-php3 .phtml
```

Arrancar el servidor, copiando el script de arranque en /etc/rc.d/init.d

```
cp /usr/local/apache/bin/apachectc /etc/rc.d/init.d/apache
/etc/rc.d/init.d/apache start
```

Para comprobar la instalación se crea un fichero llamado test.php3 con la siguiente línea:

```
<?php phpinfo() ?>
```

Luego se guarda el fichero creado en el directorio de documentos de Apache, y se abre con el navegador, luego aparecerá una página con todas las variables de PHP.

3.3 Programación PHP

3.3.1. Conceptos

El lenguaje PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C. Pero a diferencia de Java o JavaScript que se ejecutan en el servidor, esta característica permite acceder a los recursos que tenga el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podía ser una página WML.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que la páginas PHP funcionen, el servidor donde están alojadas deben soportar PHP.

3.3.2. PHP Básico

La ventaja que tiene PHP sobre otros lenguajes de programación que se ejecutan en el servidor (como podrían ser los script CGI Perl), es que permite intercalar las sentencias PHP en las páginas PHP.
En azul está el código HTML y en negro el código PHP. Se seguirá este criterio durante todo el proyecto.

```
<html>
<head>
  <title> Resoluciones de la PUCESA </title>
</head>
<body>
<BR><BR>
<?php
   echo "</center> Menú de Resoluciones <br>";
   echo "</center> de la PUCESA <br>";
?>
</body>
</html>
```

El código PHP ejecutado, imprime en primera instancia "Menú de Resoluciones", luego produce un salto de línea y finalmente imprime "de la PUCESA".

3.3.3 Variables

Una variable es una asignación de memoria, en la que se puede almacenar números enteros, números decimales o caracteres. El contenido de las variables se puede leer y se puede cambiar durante la ejecución de una página PHP.

En PHP todas las variables comienzan con el símbolo del dólar $ y no es necesario definir una variable antes de usarla. Tampoco tienen tipos, es decir que una misma variable puede contener un número y luego puede contener caracteres.

```
<html>
<head>
    <title> Variables PHP </title>
</head>
<body>
<?php
   $a = 1;
   $b = 3,34;
   $c = "PONTIFICIA UNIVERSIDAD CATOLICA SEDE AMBATO";
   echo $a, "<br>",$b,"<br>",$c;
 ?>
</body>
</html>
```

En este ejemplo se ha definido tres variables, $a, $b y $c y con la instrucción echo se imprime el valor que contienen, insertando un salto de línea entre ellas.

Existen 2 tipos de variables; variables locales que solo pueden ser usadas dentro de funciones y las variables globales que tienen su ámbito de uso fuera de las funciones, se puede acceder a una variable global desde una función con la instrucción global nombre_variable.

3.3.4. Operadores

3.3.4.1. Operadores Aritméticos

Los operadores de PHP son muy parecidos a los de C y JavaScript.

Estos son los operadores que se pueden aplicar a las variables y constantes numéricas.

Operador	Nombre	Ejemplo	Descripción
+	Suma	5+6	Suma de dos números
-	Resta	7-9	Resta de dos números
*	Multiplicación	6*3	Multiplica dos números
/	División	4/8	Divide dos números
%	Módulo	7%2	Devuelve el resto de dividir ambos números, en este ejemplo el resultado es 1
++	Suma 1	$a++	Suma 1 al contenido de una variable
--	Resta 1	$a--	Resta 1 al contenido de una variable

```
<html>
<head>
   <title> Operadores Aritméticos en PHP </title>
</head>
<body>
<?php
    $a= 8;
    $b= 3;
    echo $a + $b,"<br>";
    echo $a - $b,"<br>";
    echo $a * $b,"<br>";
    $a++;
    echo $a,"<br>";
    $b--;
    echo $b,"<br>";
?>
</body>
</html>
```

3.3.4.2. Comparación

Los operadores de comparación son usados para comparar valores y así poder tomar decisiones.

Operador	Nombre	Ejemplo	Devuelve cierto cuanto:
==	Igual	$a == $b	$a es igual $b
!=	Distinto	$a != #b	$a es distinto $b
<	Menor que	$a < $b	$a es menor que $b
>	Mayor que	$b > $b	$a es mayor que $b
<=	Menor o igual	$a <= $b	$a es menor o igual que $b
>=	Mayor o igual	$a >= $b	$a es mayor o igual que $b

Ejemplo del uso de Operadores de Comparación

```
<html>
<head>
   <title> Operadores de Comparación en PHP </title>
</head>
<body>
<?php
   $a= 8;
   $b= 3;
   $c= 3;
   echo $a == $b, "<br>";
   echo $a != $b, "<br>";
   echo $a < $b,  "<br>";
   echo $a >$b,  "<br>";
   echo $a >= $c, "<br>";
   echo $b <=$c, "<br>";
?>
</body>
</html>
```

3.3.4.3. Operadores Lógicos

Los operadores lógicos son usados para evaluar varias comparaciones, combinando los posibles valores de estas.

Operador	Nombre	Ejemplo	Devuelve cierto cuando:
&&	T	(7>2) && (2<4)	Devuelve verdadero cuando ambas condiciones son verdaderas.
And	Y	(7>2) and (2<4)	Devuelve verdadero cuando ambas condiciones son verdaderas.
II	O	(7>2) II (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera
Or	O	(7>2) or (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera
!	No	! 97>2)	Niega el valor de la expresión

Ejemplo del uso de operadores lógicos:

```
<html>
<head>
   <title> Operadores Lógicos en  PHP </title>
</head>
<body>
<? Php
    $a=  8;
    $b= 3;
    $c= 3;
    echo ($a == $b) && ($c > $b),"<br>";
    echo ($a == $b) || ($b == $c),"<br>";
    echo !($b <= $c),"<br>";
?>
</body>
</html>
```

3.3.5. Sentencias

3.3.5.1. Sentencias Condicionales

Las sentencias condicionales nos permiten ejecutar una o mas instrucciones dependiendo del resultado de evaluar una condición. Las más frecuentes es la instrucción if y la instrucción switch.

Sentencia if ... else

La sentencia if ejecuta una serie de instrucciones u otras dependiendo de la condición. Esta instrucción es la más importante en cualquier lenguaje de programación.

Formato:

```
If (condición)
{
  sentencias por verdadero
}
else
{
  sentencias por falso
}
```

Ejemplo del uso de la sentencia condicional **If ... else**

```
<html>
<head>
  <title> Ingreso al Sistema de Resoluciones </title>
</head>
<body>
<? Php
if ($password == $pas)
{     echo "Ingresando al Sistema"; }   else
{     echo "Clave no existe, consulte al Administrador"; }
?>
</body>
</html>
```

En este ejemplo si la condición es verdadera imprime el mensaje "Ingresando al sistema", caso contrario imprime el mensaje "El usuario no existe".

Sentencia Switch.... case

La sentencia switch ejecuta una u otras instrucciones dependiendo del valor de una variable, de comparación o de control.

Formato:

```
Switch (variable de control)
{
        case "valor de comparación 1": sentencias a ejecutarse; break;
        case "valor de comparación 2": sentencias a ejecutarse; break;
        case "valor de comparación 3": sentencias a ejecutarse; break;
        case "valor de comparación n": sentencias a ejecutarse; break;
    Default: // la variable de control contiene otro valor
        sentencias a ejecutarse;
}
```

Ejemplo del uso de la sentencia **Switch ... case**

```
<?php ob_start(); ?>
<html>
<head>
   <title> Control de Nivel de Acceso de Resoluciones </title>
</head>
<body>
<?php
switch($row[nivel]){
    case "1" : $url="http://localhost/admin.htm"; break;
    case "2" : $url="http://localhost/digitar.htm"; break;
    case "3" : $url="http://localhost/simple.htm"; break;
     };
 }
header("Location: " .$url);
?>
</body>
</html>
```

En el ejemplo anterior, dependiendo del valor de la variable $row[nivel] se ejecuta el bloque 1 cuando el valor es "1", el bloque 2 cuando el valor es "2" y el bloque 3 cuando el valor es "3", finalmente se abre automáticamente, la página de Nivel asignada en cada bloque.

Sentencia for

La sentencia for es la instrucción de bucles más completa. En una sola instrucción nos permite controlar todo el funcionamiento del bucle.

Formato:

```
<?php
 for (inicial; condición; ejecutar en iteración)
{
   instrucciones a ejecutar.
 }
?>
```

```
<html>
<head>
    <title>Ejemplo de PHP </title>
</head>
<body>
<?php
  $result = mysql_db_query("consejo",'select * from libro order by ');
  $total   = mysql_num_rows($result);
  $bit = 1; $cin = 1; $pag = '<a href="page.php?pag='.$bit.'">'.$bit.'<a/>';
   for ($c=1;$c<$total+1;$c++)
   {
      if ($cin > 5) {
        $bit++;
        $pag = $pag.' '.$pag = '<a href="page.php?pag='.$bit.'">'.$bit.'<a/>';
        $cin=1;  }
        $cin++;
    }
    echo $pag;
}
  mysql_free_result($result);
?>
</body>
</html>
```

En el ejemplo se muestra como generar el código HTML para poder revisar una consulta a través de páginas. En primera instancia se realiza una búsqueda de datos, luego se obtiene el total de registros producidos por la consulta y finalmente con un lazo for se controla que cada cinco filas de impresión se genere una página de consulta.

3.3.7. Impresión

Hasta ahora se ha usado la instrucción echo para realizar la salida por pantalla, esta instrucción es bastante limitada ya que no permite formatear los valores de salida, y no permite intercalar texto entre ellos.

Sentencia printf

```
<?php  printf(cadena formato, variable1, variable2...), ?>
```

La cadena de formato indica cómo se ha de representar los valores que posteriormente se indicaran. La principal ventaja es que además de poder formatear los valores de salida, también permite intercalar texto entre ellos.

```
<html>
<head>
   <title>Ejmeplo de Printf</title>
</head>
<body>
<?php
   printf("El numero dos con diferentes formatos: %d %f %.2f",2,2,2);
?></body>
</html>
```

La cadena de formato puede incluir una serie de caracteres especiales (**elementos**) que indican como formatear las variables que se incluyen en la instrucción

Elemento	Tipo de variable
%s	Cadena de caracteres
%d	Número sin decimales
%f	Número con decimales
%c	Carácter ASCII
Aunque existen otros tipos, estos son los más importantes	

3.3.8. Manejo de cadenas

Dado el uso del lenguaje PHP el tratamiento de cadenas es muy importante, existen bastantes funciones para el manejo de cadenas, a continuación se explica las más usadas.

1. strlen(cadena). Devuelve el número de caracteres de una cadena.
2. split(separador, cadena). Divide una cadena en varias usando un carácter separador.
3. sprintf(cadena de formato, var1, var2...). Formatea una cadena de texto al igual que printf pero el resultado es devuelto como una cadena.
4. substr(cadena, inicio, longitud). Devuelve una subcadena de otra, empezando por inicio y de longitud longitud.
5. chop(cadena). Elimina los saltos de línea y los espacios finales de una cadena.
6. strpos(cadena1, cadena2, texto). Reemplaza la cadena1 por la cadena2 en el texto.
7. ltrim(cadena) Elimina los espacios al inicio de una cadena de caracteres.

En el siguiente ejemplo se recibe dos argumentos a través del método GET uno de ellos indica la frase que se buscara y el otro el orden en que aparecerá la consulta. Para obtener la frase de búsqueda se eliminar primero los espacios al inicio de la cadena con la instrucción ltrim y luego se elimina los espacios al fin de la cadena con la instrucción chop.

```
<html>
<head> <title>Resultados de la Búsqueda </title> </head>
<body>
<?php
  $tex = $HTTP_POST_VARS["busca"];
  $ord = $HTTP_POST_VARS["orden"];
  $bus = ltrim($tex);
  $bus = chop($bus);
  $condi = 'tema ='.$bus;
  $link = mysql_connect("localhost","","");
  $sSQL='select * from libro where '.$condi.' order by '.$ord;
  $result=mysql_db_query("consejo",$sSQL);
  while ($row=mysql_fetch_array($result))
  {  echo $row["id"] .'<br>';
    echo.$row["fecha"].'<br>';
  };
  mysql_free_result($result);
 ?>
</body>
</html>
```

3.3.9. Funciones

Funciones

El uso de funciones permite agrupar varias instrucciones **PHP** bajo un solo nombre y poder llamarlas a estas varias veces desde diferentes sitios, sin la necesidad de escribirlas de nuevo.

```
<?php
   function Nombre(parametro1, parametro2...)
{
  instrucción 1;
  instrucción 2;
  instrucción 3;
  instrucción 4;
  instrucción n;
  return valor_de_retorno;
}
?>
```

Opcionalmente se puede pasar parámetros a las funciones que se tratarán como variables locales y así mismo se puede devolver un resultado con la instrucción **Return valor**; Esto produce la terminación de la función retornando un valor.

```
<?php
function how($c)
{
    return strlen($c);
}
?>
```

En el ejemplo anterior, se utiliza una función para contar el numero de caracteres que tiene una cadena, en este caso, la cadena es pasada a través de un parámetro, se calcula su longitud con la instrucción **strlen** y se retorna el valor del mismo con la instrucción **Return**.

3.3.10. Programación PHP Avanzado

Procesador de formularios

Envío y recepción de datos

El lenguaje PHP proporciona una manera sencilla de manejar formularios, permitiendo de esta manera procesar la información que el usuario ha introducido.
Al diseñar un formulario se debe indicar la página PHP que procesará el formulario, así como el método por el que se le pasará la información a la página.

Start.htm

```
<html>
<head>
        <title>Ingreso al Sistema</title>
</head>
<body>
<FORM ACTION= "uing.php" METHOD= "POST">
<input type = "text" name = "id"><BR>
<input type  = "password" name = "pa"><BR>
<input type = "submit" value = "enviar">
</FORM>
</body>
</html>
```

Al pulsar el botón Enviar el contenido de los elementos del formulario es enviado a la página que se indica en el atributo ACTION de la etiqueta FORM, en este caso **uing.php**, para su procesamiento.

PHP crea una variable por cada elemento del FORM, esta variable creada tiene el mismo nombre que el cuadro de texto de la página anterior y el valor que se haya introducido. En este ejemplo se han creado dos variables una llamada **$id** y otra **$pa**.

Uing.php

```
<?php ob_start(); ?>
<html>
<title>Control de Acceso al Sistema</title>
<body>
<?php
 $ide  = $HTTP_POST_VARS["id"];
 $pas  = $HTTP_POST_VARS["pa"];
 mysql_connect("localhost","","");
 mysql_select_db("consejo");
 $sSql = "select * from usuario Where id='$ide' and clave='$pas'";
 $result=mysql_db_query("consejo",$sSql);
 $url="http://localhost/error.htm";
 while ($row=mysql_fetch_array($result))
 {
    switch($row[nivel]){
    case "1" : $url="http://localhost/admin.htm"; break;
    case "2" : $url="http://localhost/digitar.htm"; break;
    case "3" : $url="http://localhost/simple.htm"; break;
     };
 }
 mysql_free_result($result);
 header("Location: " .$url);
?>
</body>
</html>
```

En este ejemplo se pasa a través del método POST un **ID de Usuario** y una **Clave**, se busca en la Base de Datos. Si existe se verifica su nivel y de acuerdo a este se envía a una página especifica para cada nivel, caso contrario se abre una página de Error.

Métodos GET y POST

Anteriormente se comento que los datos de un formulario se envía mediante el método indicado en el atributo METHOD de la etiqueta FORM, los dos métodos posibles son GET y POST.

La diferencia entre estos dos métodos radica en la forma de enviar los datos a la página, mientras que el método GET envía los datos usando la URL, el método POST los envía por la entrada estándar STDIO.
El resultado final es el mismo, solo que con el método GET se puede observar los parámetros pasados ya que están codificados en el URL.

3.4. BASES DE DATOS MySQL

3.4.1 Normalización

La Normalización es un conjunto de reglas que sirven para ayudar a los diseñadores a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataba de manipular los datos. Existen básicamente tres niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización.

3.4.2 Reglas de la Normalización

Primera Forma Normal.- Establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

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

Tercera Forma Normal.- Señala que hay que eliminar y separar cualquier dato que no sea clave. El valor de esta columna debe depender de la clave. Todos los valores deben identificarse únicamente por la clave.

Existen seis niveles más de normalización. Ellos son: Forma Normal Boyce - Codd, Cuarta Forma Normal (4NF), Quinta Forma Normal (5NF) o Forma Normal de Proyección - Unión, Forma Normal de Proyección - Unión Fuerte, Forma Normal de Proyección - Unión Extra Fuerte y Forma Normal de Clave de Dominio.

USUARIOS RESOLUCIONES

Primera Forma Normal

Id_Número
Fecha
Remas
Resoluciones
Link
Id_Usuario
Clave
Nivel

Segunda Forma Normal

Id_Número Fecha Remas Resoluciones Link

Id_Usuario Clave Nivel

3.4.3 Modelo Lógico Físico de la Base de Datos

3.4.3.1 Diseño Conceptual

Identificación de usuarios y sus roles

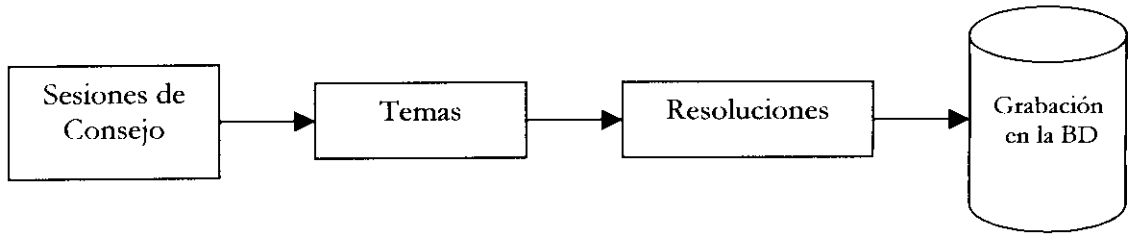
Administrador del Sistema.- Se encargara de realizar el mantenimiento de la Información relacionada con el Sistema, entre sus tareas tenemos:

- Administración de Usuarios
- Administración de la información de Sesiones del Consejo Uniersitario.

Digitador.- Se encargara del, ingreso, modificación y eliminación de las Sesiones del Consejo Universitario.

Usuarios.- Son las personas que tendrán acceso de solo lectura a la Información de las Sesiones de Consejo Universitario.

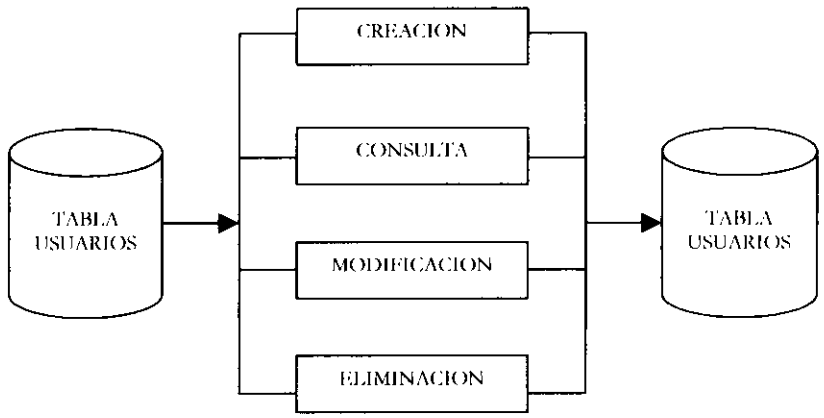
Proceso de Flujo de Trabajo



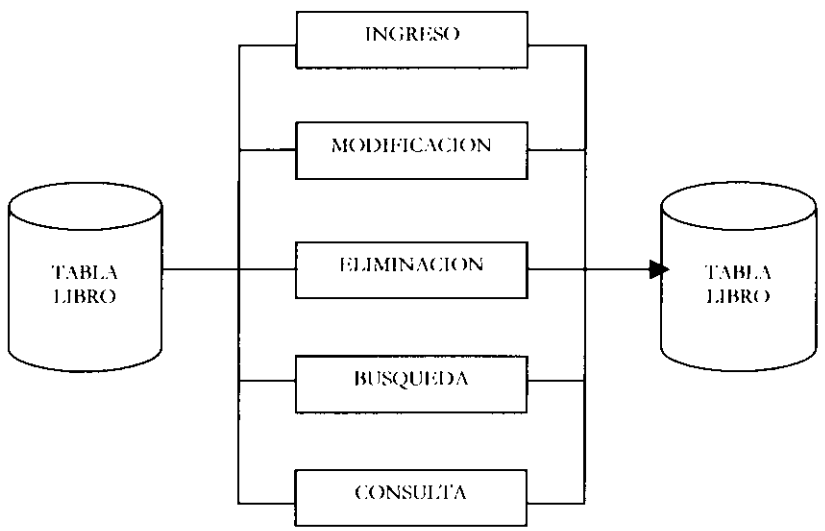
3.4.3.2 Diseño lógico

Diseño del flujo de la información

Diseño lógico de usuarios



Diseño lógico de sesiones de consejo



3.4.3.3 Diseño Físico

Diseño de la Base de Datos Consejo

TABLA LIBRO

NOMBRE COLUMNA	TIPO	LONGITUD	PRIMARY KEY
ID	SMALLINT	6	
FECHA	DATE	12	
TEMAS	TEXT	VAR	
RESOLUCIONES	TEXT	VAR	
LINK	TEXT	VAR	

TABLA USUARIOS

NOMBRE COLUMNA	TIPO	LONGITUD	PRIMARY KEY
ID	VARCHAR	10	X
CLAVE	VARCHAR	10	
NIVEL	SMALLINT	1	

3.5 Administración de la Base de Datos

Para la realización de este proyecto sobre PHP con acceso a base de datos se ha elegido la base de datos MySQL por ser gratuita y la más empleada en entornos UNIX, para lo cual el servidor donde tenemos alojadas las páginas tendrá que proporcionar herramientas para crearla o acceso Telnet para que se la cree por parte del usuario.

El comando para crear una base de datos MySQL es el siguiente:

Mysqladmin –u root create base_datos

Con este comando se crea la base de datos en el servidor de bases de datos de nuestro servidor.

Una vez realizado esto se debe crear las tablas en la base de datos, la descripción de las tablas contienen la estructura de la información que almacenaremos en ellas. Para lo cual se usara el lenguaje de consultas SQL común para todas las bases de datos relacionales.

Para crear las tablas también se puede usar la herramienta de MySQL del servidor web o se puede escribir un fichero de texto con el contenido de la sentencia SQL equivalente y luego decirle al motor de base de datos que la ejecute con la siguiente instrucción:

Mysql –u root base_datos <consejo.sql

```
# Creación de la Base de Datos #
CREATE DATABASE CONSEJO;
USE CONSEJO;
# Estructura de tabla para tabla `libro` #
CREATE TABLE libro (
  id smallint(6) NOT NULL default '0',
  fecha date default NULL, temas text, resolu text,
  link text) TYPE=MyISAM;
# Estructura de tabla para tabla `usuario` #
CREATE TABLE usuario (
  id varchar(10) default NULL,
  clave varchar(10) default NULL,
  nivel smallint(6) default NULL
) TYPE=MyISAM;
```

3.5.1 Conexión a la base de datos

Una vez creada la base de datos en nuestro servidor, el siguiente paso es conectarse a la misma desde una página PHP. PHP nos proporciona una serie de instrucciones para acceder a bases de datos MySQL.

```
<html>
<head> <title>Conexión a la Base de Datos<.title></head>
<body>
<?php function Conectarse() {
 if (!($link=mysql_connect("localhost","usuario","Password")))
   { echo "Error conectando a la base de datos.";
   exit(); }
 if (¡mysql_select_db("consejo",$link))
   { echo "Error seleccionando la base de datos.";
   exit(); }
return $link; }
$link = Conectarse();
echo "Conexión con la base de datos conseguida.<br>";
mysql_close($link); //cierra la conexión ?>
</body>
<html>
```

Al ejecutar la instrucción mysql_connect se crea un vínculo entre la base de datos y la página PHP, este vínculo será usado posteriormente en las consultas que se realicen a la base de datos.

Finalmente, una vez que se ha dejado usar el vínculo con la base de datos, se lo libera con la instrucción mysql_close para que la conexión no quede ocupada.

3.5.2 Consultas a la Base de Datos

Una vez que se ha conectado con el servidor de base de datos; ya se puede realizar consultas a las tablas de la base de datos.

Ucon.php

```
 <html>
<head> <title>CONSULTA DE USUARIOS</title></head>
<body>
<center><div align="center">USUARIOS DE RESOLUCIONES</div></center>
<?
//Conexion con la base
  mysql_connect("127.0.0.1","","");
//Ejecutamos la sentencia SQL
  $result = mysql_db_query("consejo","select * from usuario order by id");
  echo '<form >';
  echo '<table align="center" border=0>';
  echo '<tr><th>Id</th><th>Clave</th><th>Nivel</th></tr>';
//Mostramos los registros
while ($row = mysql_fetch_array($result))
{
echo '<tr>';
echo '<td><input type="text" readonly size="10" value="'.$row["id"].'"> </td>';
echo '<td><input type="text" readonly size="10" value="'.$row["clave"].'"> </td>';
echo '<td><input type="text" readonly size="10"value="'.$row["nivel"].'">
</td></tr>';
}
mysql_free_result($result)
?>
</form>
</table>
</body>
</html>
```

En este programa se utiliza las instrucciones nuevas: mysql_query, mysql_fetch_array y mysql_free_result. Con la instrucción mysql_query se realiza una consulta a la base de datos en el lenguaje de consultas SQL, con la instrucción mysql_fetch_array se extrae los datos de la consulta a un array y con mysql_free_result se libera la memoria usada en la consulta.

3.5.3 Adición de registros

Para la adición de registros se usa un formulario y en el ACTION del FORM <FORM ACTION="programaPHP"> se indica que debe ser procesado por una página PHP.
Esta página lo que hará será introducir los datos del formulario en la base de datos.

Uins.htm

```
<html>
<head>
<title>ingreso de nuevo usuario</title></head>
<body>
<center><font size="2" face="arial">crear nuevo usuario</center>
<form method="post" action="uins.php" name="datos"  onsubmit="validar(); return
document.retorna">
<table>
 <tr>
  <td>id: <br></td>
  <td><input type="text" name="id"><br></td>
 </tr>
 <tr>  <td>clave: <br></td>
  <td><input type="password" name="cl"><br></td> </tr>
 <tr> <td>tipo: <br></td>
 <td>
  <select name = "ni" size="1" style="width:150px;">
    <option value="1"> acceso de 1 nivel </option>
    <option value="2"> acceso de 2 nivel </option>
    <option value="3"> acceso de 3 nivel </option>
  </select></td> </tr>
<td><input type="submit" value="insertar"></td>
</table>
</form>
</body>
</html>
```

El programa uins.htm es un formulario que permite introducir un Id, un password y el nivel de acceso, para ser añadido a la base de datos.

Uins.php

```
<html>
<head><title>Insercion de Usuarios</title></head>
<body>
<?
//Conexion con la base
  $link = mysql_connect("localhost","","");

  $id  = $HTTP_POST_VARS["id"];
  $cl  = $HTTP_POST_VARS["cl"];
  $ni  = $HTTP_POST_VARS["ni"];

mysql_db_query("consejo","insert into usuario (id,clave,nivel) values ('$id','$cl','$ni')");
mysql_free_result($result);

?>
</body>
</html>
```

El formulario incluido en el programa Uins.htm, llama al programa uins.php que añadirá los datos a la tabla usuario.

El programa uins.php se conecta a la base de datos y añade un nuevo registro con la instrucción **insert** del lenguaje de base de datos SQL.

3.5.4 Eliminación de registros

Para indicar que elemento se eliminara, se utilizara el programa ueli.php, el cual realizara dos consultas a los datos de la tabla usuario y el contenido de la misma lo transportara a una sentencia **select** para que se cree una lista con los campos ID y PASSWORD de la tabla antes mencionada, y llamara al programa udel.php, para que realice la eliminación. Los datos son pasados a través del método POST.

Ueli.php

```
<html>
<head><title>Eliminacion de Usuarios</title></head>
<body>
<center><font size="2" face="Arial">ELIMINACION DE USUARIOS</font></center>
<?
 //Conexion con la base
mysql_connect("localhost","","");
 echo '<FORM METHOD="POST" ACTION="upre.php">';
 echo '<center><font size="1" face="Arial">Escojer ID y Clave</font></center>

 //Creamos la sentencia SQL y la ejecutamos
$sSQL="Select id From usuario Order by id";
 $result=mysql_db_query("consejo",$sSQL);

//Mostramos los registros en forma de menú desplegable

 echo '<select name="id">';
 {echo '<option>'.'id'; }
 while ($row=mysql_fetch_array($result))
  {echo '<option>'.$row["id"];}
 echo '</select>';

 //Creamos la sentencia SQL y la ejecutamos

 $sSQL="Select clave From usuario Order by id";
 $result=mysql_db_query("consejo",$sSQL);

 //Mostramos los registros en forma de menú desplegable

 echo '<select name="clave">';
 {echo '<option>'.'clave'; }
 while ($row=mysql_fetch_array($result))
 {echo '<option>'.$row["clave"];}
 mysql_free_result($result)
?>
 </select>
<INPUT TYPE="SUBMIT" value="Mostrar">
</form>
</body>
</html>
```

Udel.php

```
<html>
<head>
<title>eliminacion de usuarios</title>
</head>
<body>
<?
    //conexion con la base
    mysql_connect("localhost","","");

    $id    = $http_post_vars["id"];
    $clave  = $http_post_vars["clave"];
    $nivel  = $http_post_vars["nivel"];

    //creamos la sentencia sql y la ejecutamos
    $ssql="delete from usuario where id='$id' and clave='$clave'";
     mysql_db_query("consejo",$ssql);

?>
<center>
<p><font size="2" face="arial">usuario eliminado</font></p>
</center>
</body>
</html>
```

El programa udel.php se conecta a la base de datos y borra el registro indicando con las variables $id y $clave, que han sido pasados desde el programa ueli.php. La eliminación se realiza con la instrucción SQL **DELETE**.

Ambato, 12 de octubre de 2004

Yo, Ing. Diego Santacruz Abril, en mi calidad de Administrador del Centro de cómputo de la Pucesa, tengo a bien

CERTIFICAR

Que el señor, Wilson Fabián Toasa Romo, como parte de su Disertación de Grado titulada "La utilización de bases de datos y generación de aplicaciones web dinámicas en el servidor web de la Pucesa con sistema operativo Solaris", realizó las siguientes actividades:

- Análisis del hardware y software existente en el servidor web del Centro de Cómputo para la implementación de la aplicación de consulta de Resoluciones de Consejo Universitario.
- Implementación de la base de datos para la aplicación de consulta de Resoluciones de Consejo Universitario.
- Instalación de la aplicación para consultas de Resoluciones de Consejo Universitario, software desarrollado en base a software GNU (libre), concretamente PHP y MYSQL, que son los lenguajes y gestores de bases de datos que se utilizan regularmente en el servidor.
- La aplicación implementada se encuentra enlazada a la página web de la Pucesa, esto permitirá el acceso del personal Docente y Administrativo a las correspondientes consultas de Resoluciones de Consejo Universitario.

El interesado puede hacer uso de la presente como a bien tuviere.

Ing. Diego Santacruz Abril
Administrador Centro de Cómputo
Pucesa

4.1.1.2 ANALISIS Y OPERACIONALIZACION DE LAS VARIABLES

VARIABLES	DEFINICION CONCEPTUAL	DEFINICION INSTRUMENTAL	DEFINICION OPERACIONAL
IMPLANTACION	La instalación del motor de base de datos MySQL y el desarrollo de aplicaciones en php	El administrador del centro de computo emitirá un informe en donde certificara las siguientes actividades:	Para determinar el grado de satisfacción se sumaran los valores que corresponden a las actividades realizadas. Esto es:
TRANSACCIONES EN LA BASE DE DATOS	Es la creación, eliminación, actualización de información de la Base de Datos y el sistema de búsqueda y presentación de la información de la base de datos a través de aplicaciones en lenguaje PHP	1.- El análisis realizado al Software y Hardware existentes en el Servidor Web de la PUCESA	Para la primera actividad se asigna 25 puntos
		2.- Implementación de la Base de Datos, para la administración y consulta de las Sesiones de Consejo Universitario.	Para la segunda actividad se asigna 20 puntos
		3.- Desarrollo e instalación de aplicaciones para consultas y actualización, elaboradas con el leguaje PHP.	Para la tercera actividad se asigna 50 puntos
		4.- El acceso de las herramientas antes mencionadas a través de la Página Principal de la PUCESA.	y para la cuarta actividad se asigna 5 puntos
			Resultando así una escala de 0 a 100 puntos como grado de satisfacción de los Objetivos propuesta.

4.1.1.3 Análisis del informe presentado por el administrador del centro de computo.

Del informe presentado, se desglosa las siguientes actividades:

ACTIVIDADES	OBJETO	Puntos	PORCENTAJE
1. ANALISIS	Hardware		20%
	Software	5	
		15	
2. IMPLEMENTACION	Normalización	5	25%
	Creación de Base	5	
	Tipos de Datos	5	
	Creación Tablas	10	
3.- INSTALACIÓN	Desarrollo en PHP	40	50%
	Instalación	5	
	Pruebas	5	
4.- ENLACE	Link en el Website	5	5%
	Total	100	100%

De los resultados obtenidos del análisis realizado al informe presentado por el Administrador del Centro de Computo se concluye que el grado de satisfacción de los objetivos propuestos es del 100%.

4.1.2. Método Estadístico

Para la comprobación de las Hipótesis se realizo también una encuesta a los usuarios del Sistema de Resoluciones.

4.1.2.1 Formato de preguntas de la encuesta:

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	RESPUESTA	
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	**SI** ☐	**NO** ☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	**SI** ☐	**NO** ☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	**SI** ☐	**NO** ☐
CUALES:		
4.- ¿En la secretaria de su Escuela cuenta con Internet?	**SI** ☐	**NO** ☐

"GRACIAS POR SU COLABORACIÓN"

4.1.2.2. Análisis estadístico de los resultados obtenidos en la encuesta

Hipótesis 3. **El Desarrollo de un buscador en el Sitio Web de la PUCESA permitirá la consulta ágil y rápida de los usuarios**

Hipótesis 4. **La creación de Aplicaciones de Gestión de Base de Datos permitirán la actualización de forma automática**

S = Universo de la Encuesta
S = 8

1.- Pregunta El sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los usuarios?

2.- Pregunta El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?

4.- Pregunta En la secretaria de su Escuela cuenta con Internet?

PERSONAS	RESPUESTAS					
	PRIMERA PREGUNTA		SEGUNDA PREGUNTA		CUARTA PREGUNTA	
	Si	No	Si	No	Si	No
Rector	x		x		x	
Secretaria General	x		x			x
Asistente Secretaria	x		x			
Dir. Esc. Ing. Comercial	x		x		x	
Dir. Esc. Inglés y Lenguistica	x		x			x
Dir. Esc. Pastoral	x		x			x
Dir. Esc. Diseño Industrial	X		x		x	
Dir. Esc. Optometria	X		x			x
TOTAL	**8**		**8**		**3**	**4**

CUADRO DE FRECUENCIAS ESTADISTICAS PREGUNTA 1

Variable	Frecuencia	hi	Hi%
Si	8	1	1
No	0	0	0
Total	**8**	**1**	**100%**

Conclusión: El 100% de los encuestados responde Si

CUADRO DE FRECUENCIAS ESTADISTICAS PREGUNTA 2

Variable	Frecuencia	hi	Hi%
Si	8	1	1
No	0	0	0
Total	**8**	**1**	**100%**

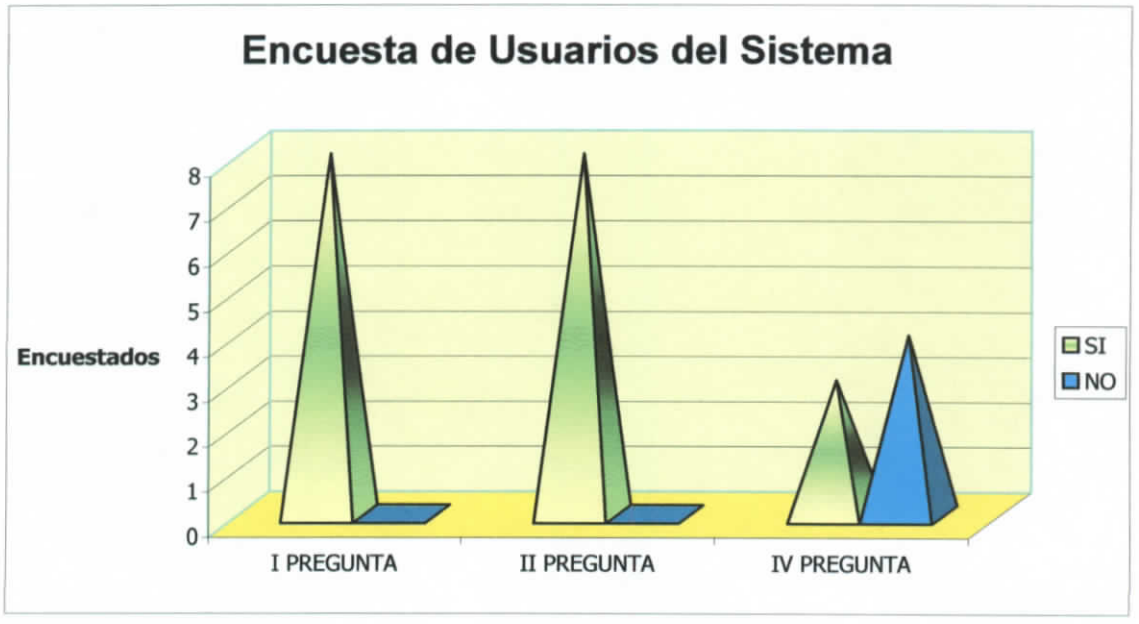
Conclusión: El 100% de los encuestados responde Si

CUADRO DE FRECUENCIAS ESTADISTICAS PREGUNTA 4

Variable	Frecuencia	hi	Hi %
Si	3	0.43	43%
No	4	0.57	57%
Total	7	1	100%

Conclusión: El 43% de los encuestado responde SI y el 57% No

4.1.2.3. Gráficos estadísticos de los resultados



4.2 VALIDACION

Ambato, Noviembre 26 de 2004
Of. 268 - EAE-PUCESA-04

Ingeniero
Telmo Viteri
DIRECTOR ESCUELA DE INGENIERIA EN SISTEMAS
Presente

De mi consideración:

Por la presente comunico que luego de haber examinado el trabajo realizado por el señor Wilson Toasa, alumno de la escuela de Ingeniería en Sistema, referente a la Utilización de bases de datos y generación de aplicaciones Web dinámicas en el servidor, considero que es aceptable y útil , por lo que se puede dar paso a la defensa de este proyecto.

Sin otro particular por el momento, suscribe.

Atentamente,

Ing. Vinicio Mejía Vayas
DIRECTOR

VMV/rr.

c.c. archivo

Ambato diciembre 16, 2004

Ingeniero
Telmo Viteri
DIRECTOR
ESCUELA INGENIERIA DE SISTEMAS
Presente.

De mi consideración:

Luego de saludarle respetuosamente, tengo a bien informar que una vez que he observado la parte práctica del trabajo "La utilización de base de datos y generación de aplicaciones Web dinámicas en el servidor Web de la PUCESA, con sistema operativo solaris", realizado por el señor Wilson Favián Toasa Romo, considero que el mismo es de mucha utilidad para la Universidad, en especial para la parte administrativa. Sin embargo, me permito recomendar que el sistema de seguridad para acceder al programa sea resguardado y vigilado continuamente con la finalidad de evitar que personas no autorizadas ingresen a él por medio de los diversos avances tecnológicos.

Atentamente,

Ab. Miriam Viteri Sánchez
SECRETARIA GENERAL PROCURADORA
PUCE SEDE AMBATO

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO
SECRETARIA GENERAL
PROCURADURIA

Ambato enero 6, 2005
PRO-001

Ingeniero
Telmo Viteri
DIRECTOR ESCUELA DE SISTEMAS
Pontificia Universidad Católica de Ambato
Presente

Ingeniero Viteri:

Por la presente me permito indicar que luego de haber examinado el trabajo realizado por el señor Wilson Favian Toasa Romo alumno de la Escuela de Sistemas referente a la "La utilización de bases de datos y generación de aplicaciones WEB dinámicas en el servidor WEB de la PUCESA con sistema operativo Solaris", en tal virtud considero que es aplicable y útil para la Universidad, por lo que se puede dar paso a la defensa de este proyecto.

Por la gentil atención, reciba mi agradecimiento.

Atentamente,

P. Dr. César González Loor
PRORRECTOR PUCE SEDE AMBATO

cc.: archivo

PONTIFICIA
UNIVERSIDAD
CATOLICA
DEL ECUADOR
SEDE AMBATO
PRO-RECTORADO

Av. Manuelita Sáenz s/n
Sector El Tropezón
Apartado Postal No.18-01-662
Telf: 593 3 411 868 ext. 101
Fax: 593 3 411 868 ext. 102
pucesedeambato@hotmail.com
Ambato - Ecuador
www.pucesa.edu.ec

4.3 CONCLUSIONES

- Las Hipótesis presentadas, han sido demostradas utilizando el Método Lógico Inductivo y el Método Estadístico basado en encuestas, lo que las transforma en Tesis.

- Todos los objetivos, propuestos en el presente proyecto, se han cumplido, ya que se ha implementado una Base de Datos y sus aplicaciones de gestión en el Web Site de la PUCESA, las cuales manejan las Sesiones del Consejo Universitario de la PUCESA.

- Se ha desarrollado un buscador en el Sitio Web de la PUCESA que permite la consulta de la información de una forma rápida y ágil, por parte de todos sus usuarios, así como la actualización de los datos.

- Se ha utilizado Software de tipo Libre, cuyo código esta disponible bajo licencia GPL.

- De la muestra tomada el 43% de las secretarias que trabajan en la PUCESA tienen conexión a Internet, y el resto el 57% no tiene conexión a Internet.

- Las secretarias de la PUCESA que no tienen conexión a Internet son: la Secretaria General, la Secretaria de Idiomas, la secretaria de Optometría, y la secretaria de Pastoral.

- Los Directores de las diferentes escuelas de la PUCESA, sugieren desarrollar proyectos parecidos al Sistema de Resoluciones, en las siguientes áreas: Integración de los Programas Académicos de todas las Escuelas y Carreras, Sistema Escolástico, Sistema de Auto evaluación de Docentes, Alumnos y Personal Administrativo, Sistema Administrativo Financiero, Sistema de Matriculación, Sistema de Administración de laboratorios, audiovisuales, auditorio.

- Los Recursos Hardware y Software del Servidor Web de la PUCESA, están siendo, manejados óptimamente por sus Administradores.

4.4 RECOMENDACIONES

- Adicionar contenidos orientados a comprobar las hipótesis, como pueden ser el Método del Chi Cuadrado y el método Modus Pones, en la asignatura de Disertación.

- Continuar con el desarrollo de nuevos sistemas con las herramientas utilizadas e implementar en las siguientes áreas: Integración de los Programas Académicos de todas las Escuelas y Carreras, Sistema Escolástico, Sistema de Auto evaluación de Docentes, Alumnos y Personal Administrativo, Sistema Administrativo Financiero, Sistema de Matriculación, Sistema de Administración de laboratorios, audiovisuales, auditorio, entre otros.

- Actualizar el pensum y los contenidos de las Materias e inducir al estudio y la investigación del lenguaje PHP y MySQL, por cuanto son las herramientas utilizadas en la actualidad, para la creación de Sitios Web Dinámicos.

- Implementar una conexión a Internet en la Secretaria General, ya que será el organismo encargado de la administración del Sistema de Resoluciones de la PUCESA

- Dar acceso a Internet a las siguientes secretarias: secretaria de Optometría, secretaria de Idiomas y Lenguas, secretaria de Pastoral.

- Desarrollar un sitio similar en la Sede principal, que maneje un sistema de Resoluciones al nivel de Pro - rectores y Rectorado.

- Adquirir de ser posibles nuevas versiones del Software libre existente en el mercado. (PHP, MySQL)

- Continuar con las mismas políticas de Administración del Servidor Web.

BIBLIOGRAFIA

LIBROS

CHERRE, **Rafael Juan,** Biblioteca del Programador WEB, **Lima Perú: Editorial MACRO, Enero 2002.**

GILFILLAN Ian, **La Biblia de MySQL**, Edición PAPERBACK, 2003

DUBOIS Paul, **WIDENIUS** Michael, **Flash dinámico - PHP y MySQL,** Editorial KNOWLEDGE 2002

RAWAT Harish, **MOORE** James, **Programación PHP4 Profesional, Editorial** Macraw Hill 2003

WELLING **Luke,** Thomson **Laura,** DesarrolloWeb en PHP y MySQL, **2002**

INTERNET

http://www.apache.org/

http://www.mysql.org/

http://www.php.net/

http://us2.php.net/

http://www.programacion.com/

http://www.desarrolloweb.com/

http://internautas.org/

http://www.wln.net/

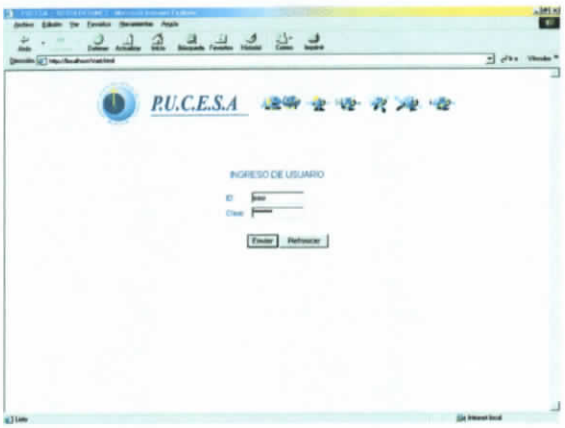
ANEXOS

A. MANUAL DE USUARIO

SISTEMA DE RESOLUCIONES DE LA PUCESA

RECURSOS NECESARIOS.- Para el funcionamiento del Sistema de Resoluciones, deben estar activos los servicios de Base de Datos MYSQL y el del Servidor WEB APACHE.

INGRESO AL SISTEMA.- Por ser una aplicación orientada al modelo cliente – servidor, esta utilizara un explorador Web para su ejecución.
Por esta razón abrimos el Internet Explorer, y en la barra de direcciones, escribimos:
http://www.pucesa.edu.ec/resoluciones/start.html ó también se puede ingresar a través del la página principal de la PUCESA, escogiendo la opción **Services on Line** (Servicios en Línea) y luego **Buscador de Resoluciones**.
Una vez ingresado al sistema aparecerá la siguiente ventana:



En esta sección, se ingresa el ID de usuario (identificador) y la Clave de Acceso, luego escogemos el botón Enviar. Si la clave ingresada o el ID de usuario es incorrecta aparecerá una pagina Web indicando el error, luego automáticamente regresara a pedir la información nuevamente.

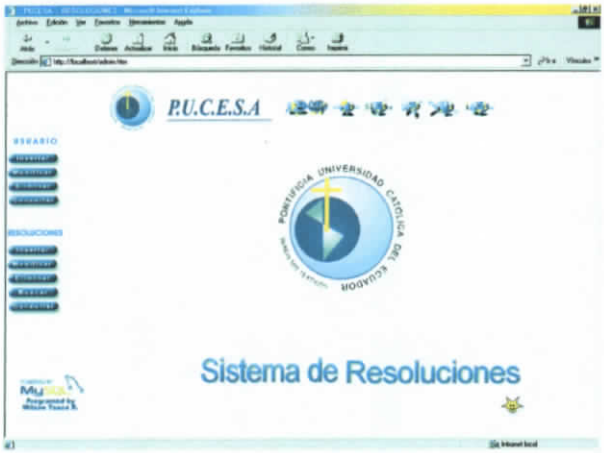
El Sistema de Resoluciones de la PUCESA, contiene tres niveles de acceso, los cuales se detallan a continuación:

1.- **ADMINISTRADOR DEL SISTEMA**.- Este nivel de acceso tiene todos los permisos para ingresar a todas las opciones del Sistema, esto es a la Sección de Usuarios, a la Sección de Sesiones y a la Sección de Consultas

2.- **ADMINISTRADOR DE SESIONES**.- Este nivel de acceso, permite el acceso a la Sección de Sesiones y a la Sección de Consultas.

3.- **USUARIO SIMPLE**.- Este nivel solo permite ingresar a la sección de consultas del Sistema.

El menú completo del Sistema es el que aparece en la ventana siguiente:



CONTENIDO DE LAS OPCIONES DEL SISTEMA:

SECCION USUARIOS

INSERTAR.- Esta opción se utiliza para ingresar un nuevo usuario con su respectiva clave de acceso.

MODIFICACION.- Esta opción se utiliza para actualizar los datos de los usuarios como son su ID de identificación, su clave y el nivel de acceso al Sistema.

ELIMINACION.- Esta sección se utiliza, para dar de baja a usuarios que por varias razones ya no tendrán acceso al Sistema.

CONSULTA.- Esta opción permite la consulta de todos los usuarios del Sistema, los parámetros que aparecen en la consulta son: ID, Clave, Nivel.

SECCION RESOLUCIONES

INSERTAR.- Graba una nueva sesión del Consejo, con sus temas y resoluciones.

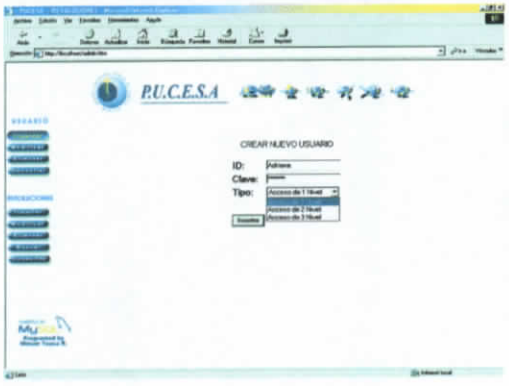
MODIFICACION.- Actualiza los datos de temas y resoluciones de una sesión grabada.

ELIMINACION.- Elimina toda una Sesión del Consejo que ya esta almacenada.

BUSQUEDA.- Permite la búsqueda de una frase en la Base de Datos

CONSULTA.- Presenta todas las Sesiones que han sido ingresadas al Sistema.

CREACION DE USUARIOS.- Para la creación de Usuarios, se escoge la opción Insertar de la sección de Usuarios y aparece la ventana siguiente:



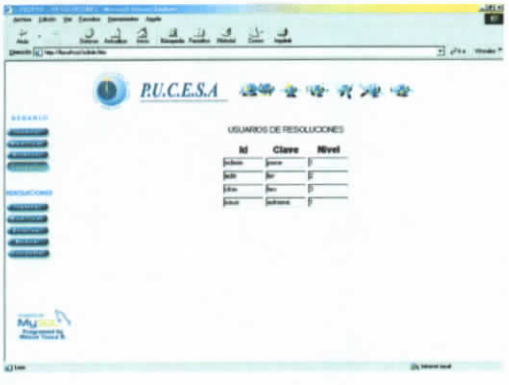
Ingresamos el ID de Usuario, la Clave correspondiente y escogemos en Nivel de Acceso que tendrá el nuevo usuario. Si el ID de usuario ya existe el sistema emitirá un mensaje de error.

MODIFICACION DE USUARIOS.- Para la modificación de datos del Usuario escogemos, la opción Modificar y aparecerá la siguiente pantalla:

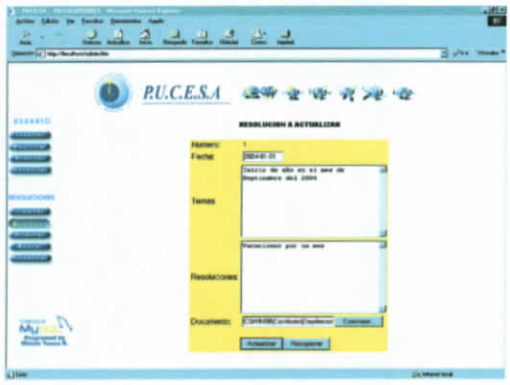


Después de seleccionar el usuario que se desea modificar, aparece una pagina donde se puede editar la información y finalmente Grabarla o Cancelar. Para la eliminación el proceso es similar.

CONSULTA DE USUARIOS.- Para observar la lista de Usuarios del Sistema, se selecciona la opción Consultar y aparecerá la siguiente pantalla:



INGRESO DE NUEVA SESION DEL CONSEJO.- Para ingresar una nueva Sesión de Consejo con sus respectivos Temas y Resoluciones, escogemos la opción Insertar de la sección de Resoluciones. Aparecerá una ventana como la siguiente:

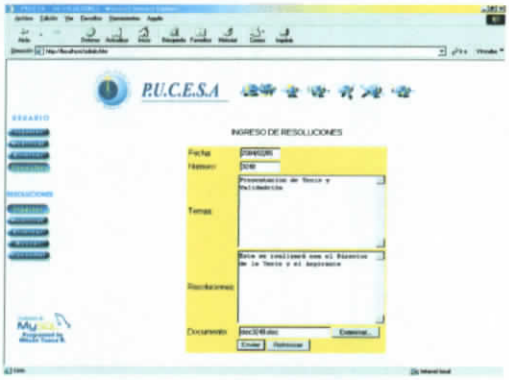


En el formulario ingresamos los siguientes datos:

Numero : Es el número de identificador de la Sesión de Consejo
Fecha : Es la fecha en que se realiza la Sesión
Temas : Son los temas que se trataron en la Sesión
Resoluciones : Son las resoluciones que se resolvieron en la Sesión
Documento : Es el documento de archivo de la Sesión de Consejo.

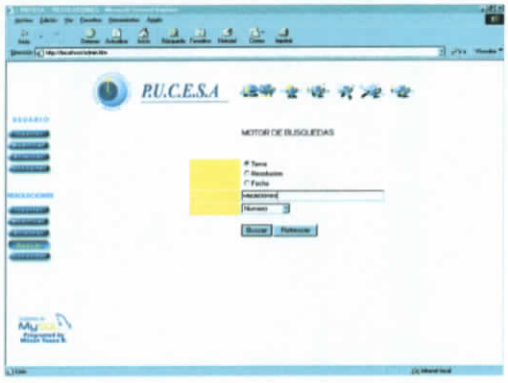
Y luego damos un clic sobre el **Botón Enviar**, para grabar la Sesión de Consejo Actual

MODIFICACION DE SESIONES.- Para la modificación de una Sesión de la Base de Datos, se escoge la opción Modificar de la Sección de Resoluciones, luego seleccionamos el Numero y la Fecha de la Sesión, y aparecerá, la ventana siguiente:



Aquí se realiza la edición de la información en todos los campos y luego se escoge el **Botón Actualizar** para grabar los cambios. Para la eliminación de Sesiones el proceso es similar.

BUSQUEDA DE INFORMACION.- Para ingresar al Motor de Búsquedas del Sistema escogemos el **Botón Buscar** y aparecerá la siguiente ventana.



Para la búsqueda tenemos las siguientes Opciones:

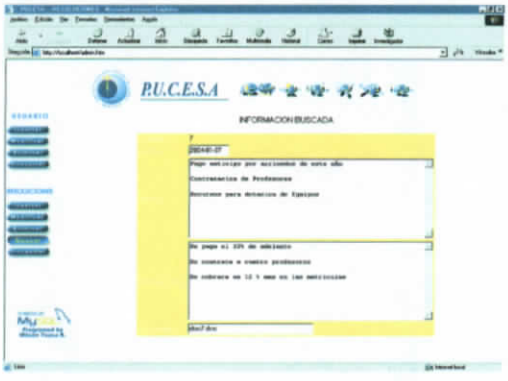
Tipo.- Permite que la búsqueda se realice en el contenido del Tema, la Resolución o la Fecha.
Frase.- Es la frase que se desea buscar en la Base de Datos de Consejo
Ordenado por.- Es el orden en el que se desea observar la consulta

Una ves ejecutada la Búsqueda aparecerán los datos encontrados, indicando el total de registros encontrados en la Base de Datos de Consejo, como se presenta en la ventana siguiente:

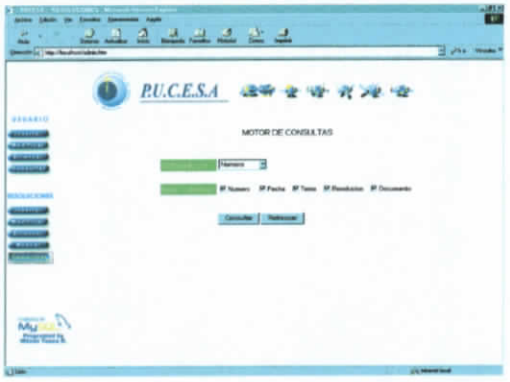


Si existen mas de 5 resoluciones, el sistemas genera automáticamente, páginas que faciliten una mejor búsqueda; es decir es un buscador con las mismas características del Buscador Google.

En las lista generada damos un clic sobre los links y aparecerá la Sesión escogida.



CONSULTAS DE INFORMACION.- Se escoge la opción Consultar, de la sección de Resoluciones y aparece la siguiente ventana:

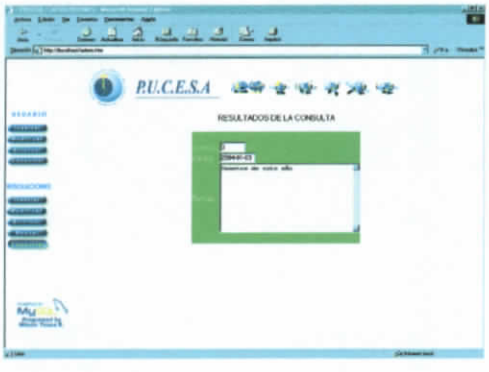


Para la búsqueda tenemos las siguientes opciones:
Ordenado por.- Se utiliza para ordenar la presentación de todos los registros de las Sesiones existentes, de acuerdo al número, fecha, tema o resolución.
Incluir Columnas.- Escogemos las columnas que deseamos que aparezcan en la Consulta, pudiendo seleccionar independientemente: Número, Fecha, Tema, Resolución y Documento.

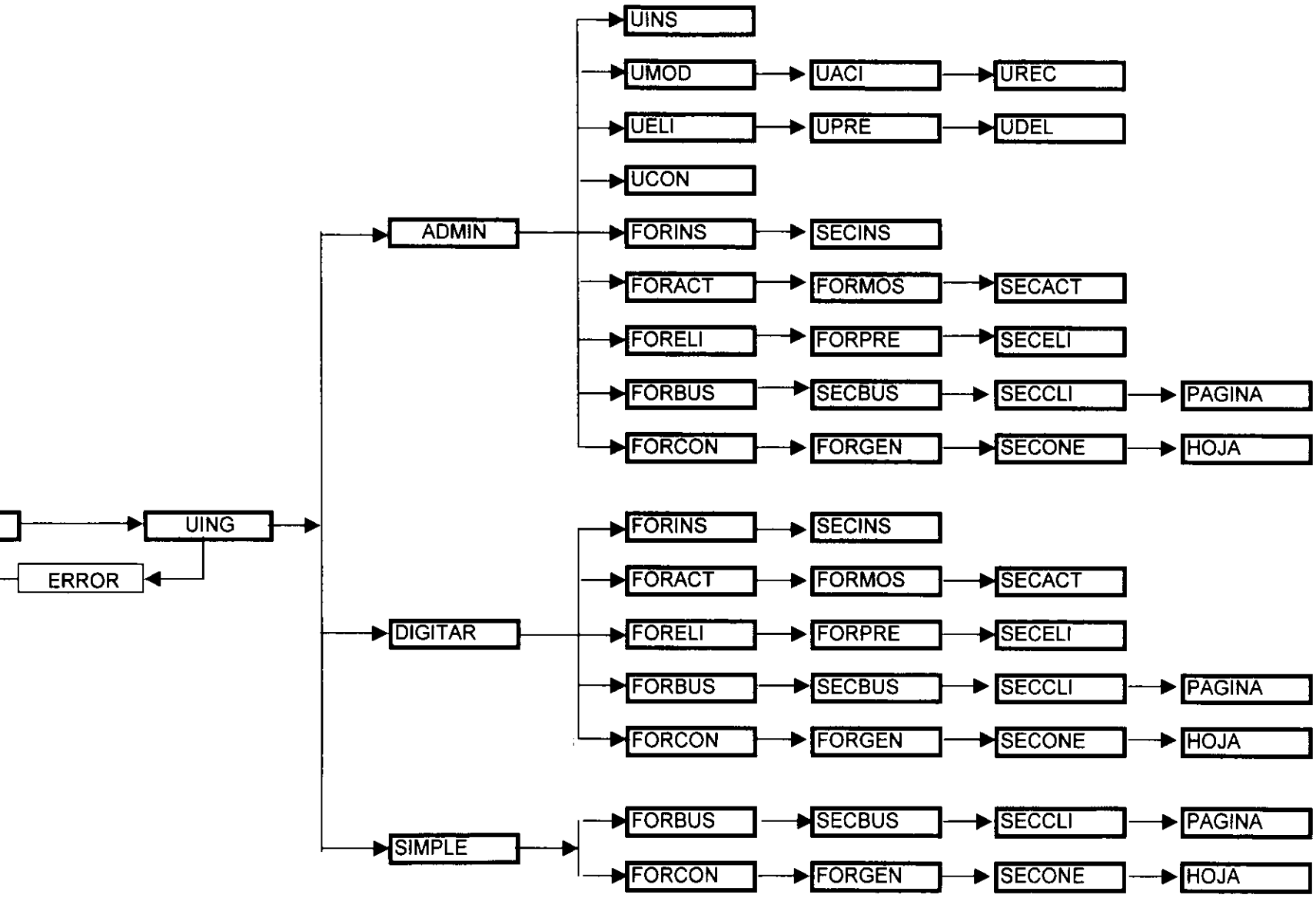
Luego de seleccionar estas opciones, se presiona el **Botón Consultar** y aparece una lista de todos los resultados, con las mismas características anotadas en la sección de Búsqueda.



Dando un clic sobre los ítems resultado de la consulta, aparecerá la Sesión con las columnas escogidas. En este caso: Numero, Fecha y Temas.



B. DIAGRAMA DE FLUJO DE EJECUCION DE LOS SUB PROGRAMAS DEL PROYECTO



C. ASIGNACION DE NOMBRES, FUNCIONES Y PARAMETROS DE ENTRADA Y SALIDA A LOS PROGRAMAS

CODIGO	:	START.HTML
FUNCION	:	Es el programa de ingreso al Sistema, su función es la de validar los usuarios del sistema y permitir su ingreso o no.
PARAMETROS	:	ID, CLAVE
CODIGO	:	UING.PHP
FUNCION	:	Verifica la existencia del Usuario y dependiendo del nivel de de usuario, redirige la navegación al menú correspondiente.
PARAMETROS	:	ID, CLAVE
CODIGO	:	ERROR.HTML
FUNCION	:	Este programa emite un mensaje de error cuando el usuario no existe o la clave no es correcta.
CODIGO	:	ADMIN.HTM
FUNCION	:	Es el programa principal del proyecto, que permite el acceso a todas las funciones de Usuarios y Resoluciones, solo se accede con el Nivel I. Su equivalente es al Administrador del Sistema.
CODIGO	:	DIGITAR.HTM
FUNCION	:	Este el programa permite el acceso a todas las funciones de Resoluciones y solo es accesible en el Nivel II, que es equivalente a la Secretaria.
CODIGO	:	SIMPLE.HTM
FUNCION	:	Este programa permite solo la consulta y la búsqueda de Resoluciones y solo permite su ingreso en el Nivel III.
CODIGO	:	UINS.HTM
FUNCION	:	Se utiliza para la creación de un nuevo usuario.
PARAMETROS	:	ID, CLAVE, NIVEL
CODIGO	:	UINS.PHP
FUNCION	:	Verifica si el nuevo usuario existe o no e inserta los datos en la Tabla correspondiente.
PARAMETROS	:	ID, CLAVE, NIVEL
CODIGO	:	UMOD.PHP
FUNCION	:	Realiza una consulta de todos los Usuarios y permite escoger Un usuario para realizar la actualización de datos.
PARAMETROS	:	ID, CLAVE
CODIGO	:	UACT.PHP
FUNCION	:	Presenta los datos del Usuario escogido y permite la edición de los mismos.
PARAMETROS	:	ID, CLAVE, NIVEL
CODIGO	:	UREC.PHP
FUNCION	:	Graba los cambios realizados con los datos del usuario.
PARAMETROS	:	ID, CLAVE, NIVEL

CODIGO	:	UELI.PHP
FUNCION	:	Permite escoger el usuario que se desea eliminar.
PARAMETROS	:	ID, CLAVE
CODIGO	:	UPRE.PHP
FUNCION	:	Presenta los datos del Usuario que se desea eliminar
PARAMETROS	:	ID, CLAVE, NIVEL
CODIGO	:	UDEL.PHP
FUNCION	:	Elimina el Usuario escogido de la Base de Datos
PARAMETROS	:	ID, CLAVE, NIVEL
CODIGO	:	UCON.PHP
FUNCION	:	Realiza una consulta general de todos los Usuarios y genera una tabla de consulta con todos ellos.
PARAMETROS	:	ID, CLAVE, NIVEL
CODIGO	:	FORINS.PHP
FUNCION	:	Este programa se utiliza para ingresar una nueva sesión del Consejo.
PARAMETROS	:	FECHA, NUM, TEMA, RESOLUCION, DOCUMENTO
CODIGO	:	SECINS.PHP
FUNCION	:	Se utiliza para grabar una nueva sesión del consejo de la PUCESA, verifica si el numero de resolución ya existe en la base de datos
PARAMETROS	:	FECHA, NUM, TEMA, RESOLUCION, DOCUMENTO
CODIGO	:	FORACT.PHP
FUNCION	:	Realiza una consulta a la base de datos y presenta el numero y la fecha para modificar los temas y resoluciones de la sesión
PARAMETROS	:	FECHA, NUM, TEMA, RESOLUCION, DOCUMENTO
CODIGO	:	FORMOS .PHP
FUNCION	:	Este modulo presenta la fecha , los temas ,las resoluciones y el documento de la sesión escogida para su actualización
PARAMETROS		FECHA, NUM, TEMA, RESOLUCION, DOCUMENTO
CODIGO	:	SECACT.PHP
FUNCION	:	Este programa se utiliza para grabar los cambios realizados en las resoluciones por parte del usuario
PARAMETROS	:	FECHA, NUM, TEMA, RESOLUCION, DOCUMENTO
CODIGO	:	FORELI.PHP
FUNCION	:	Este modulo realiza una consulta de todas las sesiones de la PUSESA y genera una lista para escoger la que se desea eliminar.
PARAMETROS	:	FECHA, NUM, TEMA, RESOLUCION, DOCUMENTO
CODIGO	:	FORPRE .PHP
FUNCION	:	Este programa presenta los datos de la sesión escogida para su eliminación
PARAMETROS	:	FECHA, NUM, TEMA, RESOLUCION, DOCUMENTO
CODIGO	:	SECELI.PHP
FUNCION	:	Se conecta con la base de datos y elimina la sesión escogida
PARAMETROS	:	FECHA, NUM, TEMA, RESOLUCION, DOCUMENTO

CODIGO	:	FORBUS.PHP
FUNCION	:	Este programa se utiliza para ingresar una frase por el usuario y que esta sea buscada en la fecha, tema o resolución
PARAMETROS	:	TIPO, FRASE y ORDENADO
CODIGO	:	SECBUS.PHP
FUNCION	:	Realiza una consulta, buscando la FRASE deseada, así como también crea los LINKS automáticamente, para poder revisar las resoluciones obtenidas y genera los LINKS de las páginas si el resultado de la consulta es mayor a 5.
PARAMETROS	:	TIPO, FRASE y ORDENADO, ID, TOTAL, SQL, PAG TEMP
CODIGO	:	SECCLI.PHP
FUNCION	:	Presenta la resolución escogida en la búsqueda para su respectiva consulta.
PARAMETROS	:	ID
CODIGO	:	PAGINA.PHP
FUNCION	:	Realiza una consulta, con la cadena pasada desde el motor de búsquedas, también genera los LINKS automáticamente, para poder revisar las resoluciones obtenidas y crea los LINKS de las páginas si el resultado de la consulta es mayor a 5.
PARAMETROS	:	TOTAL, SQL, ID, PAG, TEMP
CODIGO	:	FORCON.PHP
FUNCION	:	Este modulo se utiliza para escoger las columnas que se desea en la consulta general, así como también en que orden aparecerán las sesiones.
PARAMETROS	:	NUMERO, FECHA, TEMA, RESOLUCION, ORDENADO DOCUMENTO.
CODIGO	:	FORGEN.PHP
FUNCION	:	Realiza una consulta general, así como también crea los LINKS automáticamente, para poder revisar las resoluciones obtenidas y genera los LINKS de las páginas si el resultado de la consulta es mayor a 5.
PARAMETROS	:	PAG, SQL, NUMERO, FECHA, TEMA, RESOLUCION, ORDENADO, DOCUMENTO.
CODIGO	:	SECONE.PHP
FUNCION	:	Genera una tabla con las columnas escogidas por el usuario y en ella Presenta toda la sesión seleccionada.
PARAMETROS	:	ID, NUMERO, FECHA, TEMA, RESOLUCION, DOCUMENTO.
CODIGO	:	HOJA.PHP
FUNCION	:	Realiza una consulta general de las sesionas, también genera los LINKS automáticamente para poder revisar las sesiones obtenidas por la consulta en grupos de 10.
PARAMETROS	:	TOTAL, SQL, ID, PAG, NUMERO, FECHA, TEMA, RESOLUCION, DOCUMENTO.

D. DOCUMENTACION DE LOS PROGRAMAS

PROGRAMA START.HTLM

```
<HTML><HEAD>
<TITLE>:: PUCESA :: RESOLUCIONES</TITLE>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type><!-- Fireworks 4.0  Dreamweaver 4.0
target.  Created Sun Nov 17 17:04:57 GMT-0500 (Hora est. del Pacífico de SA) 2002-->
<script language="JAVASCRIPT">
function validar()
{ if (datos.pa.value=="")
   {alert("Favor ingresar clave")
    document.retorna = false;   }
   else
    document.retorna = true;
 if (datos.id.value=="")
   {alert("Favor ingresar Id usuario")
    document.retorna = false;    }
   else
    document.retorna = true;    }
</script>
<BODY bgColor=#ffffff leftMargin=0 text=#003366 topMargin=0 marginheight="0"
marginwidth="0"><BR>
<center><font  size="2" face="Arial">INGRESO DE USUARIO</font></center>
<form action="uing.php" method="POST" name="datos" onsubmit="validar(); return document.retorna">
 <table border="0" align="center">
  <tr>    <td><font size="1" face="Arial">ID:
    </font></td>
    <td><font size="1" face="Arial"> 
    <input type="text" size="12" maxlength="35" name="id"></font></td>  </tr>
 <tr>  <td><font size="1" face="Arial">Clave: </font></td>
    <td><p align="left"><font size="2" face="Arial"> 
    <input type="password" size="12" maxlength="8" name="pa"></font></p></td>  </tr>
 </tr> <td> </td>  </tr>
 <tr> <td> </td>
        <td><font size="1" face="Arial">
        <input type="submit" value="Enviar" onclick="validar();return document.retorna;">
        <input type="reset" value="Refrescar"></font></td>  </tr>
 </table>
</form></BODY>
</HTML>
```

PROGRAM ADMIN.HTML

```
<HTML>
<HEAD><TITLE>:: PUCESA :: RESOLUCIONES</TITLE>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<META content="MSHTML 5.00.2614.3500" name=GENERATOR></HEAD>
<FRAMESET border=0 cols=* frameBorder=NO frameSpacing=0 rows=105,450*>
 <FRAME name=topFrame noResize scrolling=no src="tope.html">
 <FRAMESET border=0 cols=150,* frameBorder=no frameSpacing=0 rows=*>
   <FRAMESET border=0 cols=* frameBorder=no frameSpacing=0 rows=400,100*>
    <FRAME name=leftFrame noResize scrolling=no src="main.htm">
    <FRAME name=downFrame noResize scrolling=no src="down.htm">
   </FRAMESET>
  <FRAME name=mainFrame src="http://localhost/blanco.htm">
 </FRAMESET>
</FRAMESET>
</HTML>
```

PROGRAMA UING.PHP

```
<?php ob_start(); ?>
<html>
<title>Control de Acceso al Sistema</title>
<?php
 $ide  = $HTTP_POST_VARS["id"];
 $pas  = $HTTP_POST_VARS["pa"];
 echo $ide;
 echo '<br>';
 echo $pas;
 echo '<br>';
 mysql_connect("localhost","root");
 mysql_select_db("consejo");
 $sSql = "select * from usuario Where id='$ide' and clave='$pas'";
 $result=mysql_db_query("consejo",$sSql);
 $url="http://localhost/error.htm";
 while ($row=mysql_fetch_array($result))
 {
    switch($row[nivel]){
    case "1" : $url="http://localhost/admin.htm"; break;
    case "2" : $url="http://localhost/digitar.htm"; break;
    case "3" : $url="http://localhost/simple.htm"; break;
     };
 }
 mysql_free_result($result);
 header("Location: " .$url);
?></html>
```

PROGRAM ERROR.HTLM

```
<HTML>
<HEAD><TITLE>:: PUCESA :: ERROR DE USUARIO</TITLE></HEAD>
<meta http-equiv="refresh" content="12; url=http://localhost/start.html">
<br><center><font  size="2" face="Arial"><p>USUARIO NO EXISTE</p>
<br><p>CONSULTE AL ADMINISTRADOR</p>
<br><img src="http://localhost/diez.gif">
<br><p>!Esta pagina se redigira en 10 segundos...!</p>
</font>
</center>
</BODY>
</HTML>
```

PROGRAMA DIGITAR. HTML

```
<HTML>
<HEAD><TITLE>:: PUCESA :: RESOLUCIONES</TITLE>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<META content="MSHTML 5.00.2614.3500" name=GENERATOR></HEAD>
<FRAMESET border=0 cols=* frameBorder=NO frameSpacing=0 rows=105,450*>
 <FRAME name=topFrame noResize scrolling=no src="tope.html">
 <FRAMESET border=0 cols=150,* frameBorder=no frameSpacing=0 rows=*>
   <FRAMESET border=0 cols=* frameBorder=no frameSpacing=0 rows=400,100*>
     <FRAME name=leftFrame noResize scrolling=no src="res.htm">
     <FRAME name=downFrame noResize scrolling=no src="down.htm">
   </FRAMESET>
  <FRAME name=mainFrame src="http://localhost/blanco.htm">
 </FRAMESET>
</FRAMESET>
</HTML>
```

PROGRAMA SIMPLE.HTML

```
<HTML>
<HEAD><TITLE>:: PUCESA :: RESOLUCIONES</TITLE>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-Type>
<META content="MSHTML 5.00.2614.3500" name=GENERATOR></HEAD>
<FRAMESET border=0 cols=* frameBorder=NO frameSpacing=0 rows=105,450*>
 <FRAME name=topFrame noResize scrolling=no src="tope.html">
 <FRAMESET border=0 cols=150,* frameBorder=no frameSpacing=0 rows=*>
   <FRAMESET border=0 cols=* frameBorder=no frameSpacing=0 rows=400,100*>
     <FRAME name=leftFrame noResize scrolling=no src="mon.htm">
     <FRAME name=downFrame noResize scrolling=no src="down.htm">
   </FRAMESET>
  <FRAME name=mainFrame src="http://localhost/blanco.htm">
 </FRAMESET>
</FRAMESET>
</HTML>
```

PROGRAM UINS.HTML

```
<HTML>
<HEAD> <TITLE>Ingreso de nuevo usuario</TITLE></HEAD>
<BODY>
<script language="JAVASCRIPT">
function validar()
{ if (datos.id.value=="")
   {alert("Favor ingresar Id usuario")
     document.retorna = false;    }
   else document.retorna = true;
 if (datos.cl.value=="")
   {alert("Favor ingresar clave")
     document.retorna = false;    }
   else
     document.retorna = true;
}
</script>
<div align="center">
<center><font size="2" face="Arial">CREAR NUEVO USUARIO</center>
<FORM method="POST" action="uins.php" name="datos"  onsubmit="validar(); return document.retorna">
<TABLE>
 <TR>  <TD>ID: <br></TD>
  <TD><INPUT TYPE="TEXT" NAME="id"><br></TD> </TR>
 <TR>  <TD>Clave: <br></TD>
  <TD><INPUT TYPE="PASSWORD" NAME="cl"><br></TD> </TR>
 <TR> <TD>Tipo: <br></TD>
 <TD>  <select name = "ni" size="1" style="width:150px;">
   <option value="1"> Acceso de 1 Nivel </option>
   <option value="2"> Acceso de 2 Nivel </option>
   <option value="3"> Acceso de 3 Nivel </option>
  </select></TD> </TR>
<TR><TD><P> </P></TD></TR>
 <TR> <TD><INPUT TYPE="SUBMIT" value="Insertar"></TD> </TR>
</TABLE>
</FORM>
</font>
</div>
</BODY>
</HTML>
```

PROGRAMA UINS.PHP

```
<HTML>
<HEAD>
<TITLE>Insertar.php</TITLE>
</HEAD>
<BODY>
<?
//Conexion con la base
 $link = mysql_connect("localhost","root");
 $id  = $HTTP_POST_VARS["id"];
 $cl  = $HTTP_POST_VARS["cl"];
 $ni  = $HTTP_POST_VARS["ni"];
//Ejecucion de sentencias SQL
 $sSQL = "select * from usuario where id='$id'";
 $result=mysql_db_query("consejo",$sSQL);
//Contamos las filas de la consulta
 $total = mysql_num_rows($result);
 if ($total == 0){
    $message = "Nuevo usuario creado";
    $color = "Black";
    mysql_db_query("consejo","insert into usuario (id,clave,nivel) values ('$id','$cl','$ni')");
 } else
 {
    $message = "Usuario ya existe";
    $color = "Red";
 };
 echo '<font size="4" face="Arial Black" color='.$color.'>';
 echo '<br>';
 echo '<br>';
 echo '<div align="center">'.$message.'</div></font>';
 mysql_free_result($result);
?>
</BODY>
</HTML>
```

PROGRAMA UMOD.PHP

```
<HEAD>
<TITLE>Actualizacion de Usuarios</TITLE>
</HEAD>
<BODY>
<div align="center">
<center><font size="2" face="Arial">MODIFICACION DE USUARIOS</font></center>
<?
//Conexion con la base
mysql_connect("localhost","root");
echo '<FORM METHOD="POST" ACTION="uact.php">';
echo '<center><font size="1" face="Arial">Escojer ID y Clave</font></center> <br> <br>';
//Creamos la sentencia SQL y la ejecutamos
$sSQL="Select id From usuario Order by id";
$result=mysql_db_query("consejo",$sSQL);
//Mostramos los registros en forma de men£ desplegable
echo '<select name="id">';
{echo '<option>'.id; }
while ($row=mysql_fetch_array($result))
{echo '<option>'.$row["id"];}
echo '</select>';
```

```
//Creamos la sentencia SQL y la ejecutamos
$sSQL="Select clave From usuario Order by id";
$result=mysql_db_query("consejo",$sSQL);
//Mostramos los registros en forma de men£ desplegable
echo '<select name="clave">';
{echo '<option>'.'clave'; }
while ($row=mysql_fetch_array($result))
{echo '<option>'.$row["clave"];}
mysql_free_result($result)
?>
</select>
<INPUT TYPE="SUBMIT" value="Mostrar">
</FORM>
</div>
</BODY>
</HTML>
```

PROGRAMA UACT. PHP

```
<HTML><HEAD><TITLE>ACTUALIZACION DE USUARIOS</TITLE></HEAD>
<BODY>
<div align="center">
<font size="2" face="Arial">
<br>
<div align="center">DATOS DE USUARIO A ACTUALIZAR
</font>
</div>
<? //Conexion con la base
 mysql_connect("localhost","root");
echo '<FORM NAME= "form1" METHOD="POST" ACTION="urec.php">';
 //Creamos la sentencia SQL y la ejecutamos
 $id = $HTTP_POST_VARS["id"];
 $clave = $HTTP_POST_VARS["clave"];
 $sSQL="Select * From usuario Where id='$id' and clave='$clave'";
 $result=mysql_db_query("consejo",$sSQL);
 while ($row=mysql_fetch_array($result))
 {  $i = $row["id"];
    $cl = $row["clave"];
    $ni = $row["nivel"];
}
echo'<table border="0"> ';
echo'<tr> <td><font size="2" face="Arial">ID: </font></td> ';
echo'<td><font size="2" face="Arial"> ';
echo'<input type="text" readonly size="10" maxlength="10" name="id" value='.$i.'></font></td></tr> ';
echo'<tr> <td><font size="2" face="Arial">Clave: </font></td> ';
echo'<td><font size="2" face="Arial"> ';
echo'<input type="text" size="10" maxlength="10" name="clave" value='.$cl.'></font></td></tr> ';
echo'<tr> <td><font size="2" face="Arial">Nivel: </font></td> ';
echo'<td><font size="2" face="Arial"> ';
echo'<input type="text" size="10" maxlength="10" name="nivel" value='.$ni.'></font></td></tr> ';
echo'<tr><td> </td> <td><font size="2" face="Arial">';
echo '<br>';
echo'<input type="submit" value="Grabar" >';
echo'<input type="reset" value="Rescatar"></font></td>';
echo'</tr>';
echo' </table>';
echo'</form>';
?>
</BODY>
</HTML>
```

PROGRAMA UREC. PHP

```
<HTML>
<HEAD>
<TITLE>GRABACION DE CAMBIOS</TITLE>
</HEAD>
<BODY>
<?
//Conexion con la base
mysql_connect("localhost","root");
  $id     = $HTTP_POST_VARS["id"];
  $clave  = $HTTP_POST_VARS["clave"];
  $nivel  = $HTTP_POST_VARS["nivel"];
//Creamos la sentencia SQL y la ejecutamos
$sSQL="Update usuario Set id='$id',clave='$clave',nivel='$nivel' Where id='$id'";
mysql_db_query("consejo",$sSQL);
?>
<center>
<p><font size="2" face="Arial">Datos de Usuario Actualizados</font></p>
</center>
</BODY>
</HTML>
```

PROGRAMA UELI.PHP

```
<HTML>
<HEAD>
<TITLE>Eliminacion de Usuarios</TITLE>
</HEAD>
<BODY>
<div align="center">
<center><font size="2" face="Arial">ELIMINACION DE USUARIOS</font></center>
<?  //Conexion con la base
mysql_connect("localhost","root");
echo '<FORM METHOD="POST" ACTION="upre.php">';
echo '<center><font size="1" face="Arial">Escoger ID y Clave</font></center>';
//Creamos la sentencia SQL y la ejecutamos
$sSQL="Select id From usuario Order by id";
$result=mysql_db_query("consejo",$sSQL);
//Mostramos los registros en forma de menú desplegable
echo '<select name="id">';
{echo '<option>'.'id'; }
while ($row=mysql_fetch_array($result))
{echo '<option>'.$row["id"];}
echo '</select>';
//Creamos la sentencia SQL y la ejecutamos
$sSQL="Select clave From usuario Order by id";
$result=mysql_db_query("consejo",$sSQL);
//Mostramos los registros en forma de menú desplegable
echo '<select name="clave">';
{echo '<option>'.'clave'; }
while ($row=mysql_fetch_array($result))
{echo '<option>'.$row["clave"];}
mysql_free_result($result) ?>
</select>
<INPUT TYPE="SUBMIT" value="Mostrar">
</FORM>
</div>
</BODY>
</HTML>
```

PROGRAMA UPRE. PHP

```
<HTML>
<HEAD>
<TITLE>ACTUALIZACION DE USUARIOS</TITLE>
</HEAD>
<BODY>
<div align="center"><font size="2" face="Arial">
<div align="center">DATOS DE USUARIO A ELIMINAR</font></div>
<?
  //Conexion con la base
  mysql_connect("localhost","root");
 echo '<FORM NAME= "form1" METHOD="POST" ACTION="udel.php">';
  //Creamos la sentencia SQL y la ejecutamos
$id = $HTTP_POST_VARS["id"];
$clave = $HTTP_POST_VARS["clave"];
$sSQL="Select * From usuario Where id='$id' and clave='$clave'";
$result=mysql_db_query("consejo",$sSQL);
  while ($row=mysql_fetch_array($result))
  {   $i = $row["id"];
      $cl = $row["clave"];
      $ni = $row["nivel"];
 }
 echo'<table border="0"> ';
 echo'<tr> <td><font size="2" face="Arial">ID: </font></td> ';
 echo'<td><font size="2" face="Arial"> ';
 echo'<input type="text" readonly size="10" maxlength="10" name="id" value='.$i.'></font></td></tr> ';
 echo'<tr> <td><font size="2" face="Arial">Clave: </font></td> ';
 echo'<td><font size="2" face="Arial"> ';
 echo'<input type="text" readonly size="10" maxlength="10" name="clave" value='.$cl.'></font></td></tr> ';
 echo'<tr> <td><font size="2" face="Arial">Nivel: </font></td> ';
 echo'<td><font size="2" face="Arial"> ';
 echo'<input type="text" readonly size="10" maxlength="10" name="nivel" value='.$ni.'></font></td></tr> ';
 echo'<tr><td> </td> <td><font size="2" face="Arial">';
 echo '<br>';
 echo'<input type="submit" value="Eliminar" >';
 echo '</font></td>';
echo' </table>';
 echo'</form>'; ?>
</BODY>
</HTML>
```

PROGRAMA UDEL.PHP

```
<HTML>
<HEAD>
<TITLE>ELIMINACION DE USUARIOS</TITLE>
</HEAD>
<BODY>
<? //Conexion con la base
  mysql_connect("localhost","root");
  $id     = $HTTP_POST_VARS["id"];
  $clave  = $HTTP_POST_VARS["clave"];
  $nivel  = $HTTP_POST_VARS["nivel"];
  $sSQL="Delete from usuario Where id='$id' and clave='$clave'";
mysql_db_query("consejo",$sSQL); ?>
<center> <p><font size="2" face="Arial">Usuario eliminado</font></p></center>
</BODY>
</HTML>
```

PROGRAMA UCON.PHP

```
<HTML>
<HEAD><TITLE>CONSULTA DE USUARIOS</TITLE></HEAD>
<BODY>
<font size="2" face="Arial">
<center><div align="center">USUARIOS DE RESOLUCIONES</div></center>
<?
//Conexion con la base
mysql_connect("127.0.0.1","","");
//Ejecutamos la sentencia SQL
$result=mysql_db_query("consejo","select * from usuario order by id");
?>
<form >
<table align="center" border=0>
<tr><th>Id</th>
<th>Clave</th>
<th>Nivel</th></tr>
<?
//Mostramos los registros
while ($row=mysql_fetch_array($result))
{
echo '<tr>';
echo '<td><input type="text" readonly size="10" value="'.$row["id"].'"> </td>';
echo '<td><input type="text" readonly size="10" value="'.$row["clave"].'"> </td>';
echo '<td><input type="text" readonly size="10"value="'.$row["nivel"].'"> </td></tr>';
}
mysql_free_result($result)
?>
</form>
</table>
</font>
<BODY>
</BODY>
</HTML>
```

PROGRAMA FORINS.PHP

```
<HTML>
<HEAD><title>Resoluciones de sesiónes de Consejo PUCESA</title></HEAD>
<body >
<script language="JAVASCRIPT">
function validar()
{ if (datos.fecha.value=="")
   {alert("favor ingresar fecha")
     document.retorna = false;    }
   else
     document.retorna = true;
 if (datos.id.value=="")
   {alert("favor ingresar numero")
     document.retorna = false;    }
   else
     document.retorna = true;
 if (datos.link.value=="")
   {alert("favor ingresar documento")
     document.retorna = false;    }
   else
     document.retorna = true;}
</script>
<center>
<font size="2" face="Arial">INGRESO DE RESOLUCIONES</font></center>
```

```
<form action="secins.php" method="POST" name="datos" onsubmit="validar(); return document.retorna">
  <table border="0" align="center" bgcolor="#FFCC33">
    <tr><td><font size="2" face="Arial">Fecha:
      </font></td>
      <td><font size="1" face="Arial"> <input
      type="date" size="10" maxlength="35" name="fecha"></font></td>      </tr>
    <tr><td><font size="2" face="Arial">Numero: </font></td>
      <td><p align="left"><font size="2" face="Arial"> <input
      type="text" size="10" maxlength="8" name="id" onKeypress="if (event.keyCode < 45 || event.keyCode >
57) event.returnValue = false;"></font></p> </td></tr>
      <tr> <td><font size="2" face="Arial">Temas:
      </font></td>
      <td><font size="1" face="Arial">
      <textarea name="temas" rows="9" cols="35">
      </textarea></font></td>      </tr>
      <tr> <td><font size="2" face="Arial">Resoluciones:
      </font></td>
      <td><font size="1" face="Arial">
      <textarea name="resolu" rows="9" cols="35">
      </textarea></font></td> </tr>
      <tr>  <td><font size="2" face="Arial">Documento: </font></td>
      <td><p align="left"><font size="2" face="Arial"> <input
      type="file" size="25" maxlength="20" name="link" > </font></p>
      </td>  </tr>
      <tr> <td><font size="1" face="Arial">
      <input type="submit" value="Enviar" onclick="validar();return document.retorna;">
      <input type="reset" value="Refrescar"></font></td>      </tr>
  </table>
</form>
</BODY>
</HTML>
```

PROGRAMA SECINS.PHP

```
<HTML>
<HEAD><TITLE>Insertar Resolucion PUCESA</TITLE></HEAD>
<BODY>
<?
//Conexion con la base
 $connect = mysql_connect("localhost","root");
 $id     = $HTTP_POST_VARS["id"];
 $fecha  = $HTTP_POST_VARS["fecha"];
 $temas  = $HTTP_POST_VARS["temas"];
 $resolu = $HTTP_POST_VARS["resolu"];
 $link   = $HTTP_POST_VARS["link"];
 $sSQL = "Select * From libro Where id='$id'";
 $result = mysql_db_query("consejo",$sSQL);
 $total = mysql_num_rows($result);
 if ($total == 0){
    $message = "Nuevo registro creado";
    $color = "Black";
    mysql_db_query("consejo","insert into libro (id,fecha,temas,resolu,link) values
('$id','$fecha','$temas','$resolu','$link')");
 }  else  {  $message = "Numero de sesion ya existe";
    $color = "Red";  };
 echo '<font size="4" face="Arial Black" color='.$color.'>';
echo '<div align="center">'.$message.'</div></font>';
 mysql_free_result($result);
?></BODY>
</HTML>
```

PROGRAMA: FORACT.PHP

```
<HTML>
<HEAD><TITLE>Actualizacion de Resoluciones de la PUCESA</TITLE></HEAD>
<BODY>
<div align="center">
<center><font size="2" face="Arial">MODIFICACION DE RESOLUCIONES</font></center>
<?
 //Conexion con la base
 mysql_connect("localhost","root");
 echo '<FORM METHOD="POST" ACTION="formos.php">
 <center><font size="1" face="Arial">Escojer numero y fecha</font></center>
 <br>
 <br>';
 //Creamos la sentencia SQL y la ejecutamos
sSQL="Select id From libro Order by id";
 $result=mysql_db_query("consejo",$sSQL);
/Mostramos los registros en forma de menú desplegable
 echo '<select name="id">';
 {echo '<option>'.'id'; }
 while ($row=mysql_fetch_array($result))
 {echo '<option>'.$row["id"];}
 echo '</select>';
//Creamos la sentencia SQL y la ejecutamos
 $sSQL="Select fecha From libro Order by id";
 $result=mysql_db_query("consejo",$sSQL);
 //Mostramos los registros en forma de menú desplegable
 echo '<select name="fecha">';
 {echo '<option>'.'fecha'; }
 while ($row=mysql_fetch_array($result))
 {echo '<option>'.$row["fecha"];}
 mysql_free_result($result)
?>
</select>
<INPUT TYPE="SUBMIT" value="Mostrar">
</FORM>
</DIV>
</BODY>
</HTML>
```

PROGRAM FORMOS.PHP

```
<HTML>
<HEAD><TITLE>ACTUALIZACION DE RESOLUCION</TITLE></HEAD>
<BODY>
 <center>
  <font size="1" face="Arial Black">RESOLUCION A ACTUALIZAR  </font>
 </center>
<?
 //Conexion con la base
 mysql_connect("localhost","root");
echo '<FORM NAME= "form1" METHOD="POST" ACTION="secact.php">';
 //Creamos la sentencia SQL y la ejecutamos
 $id = $HTTP_POST_VARS["id"];
 $fecha = $HTTP_POST_VARS["fecha"];
 //echo $id;
 //echo $fecha;
 $sSQL="Select * From libro Where id='$id' and fecha='$fecha'";
 $result=mysql_db_query("consejo",$sSQL);
 while ($row=mysql_fetch_array($result))
 {  $i = $row["id"];
```

```
    $fe = $row["fecha"];
    $te = $row["temas"];
    $re = $row["resolu"];
    $li = $row["link"]; }
echo'<table border="0" align="center" bgcolor="#FFCC33">';
echo'<tr> <td><font size="2" face="Arial">Numero: </font></td> ';
echo'<td><p align="left"><font size="1" face="Arial"> ';
echo $i;
echo'</font></p> </td> </tr>';
echo'<input type="hidden" size="10" maxlength="35" name="id" value='.$i.'></font></td></tr> ';
echo'<tr> <td><font size="2" face="Arial">Fecha: </font></td> ';
echo'<td><font size="2" face="Arial"> ';
echo'<input type="date" size="10" maxlength="35" name="fecha" value='.$fe.'></font></td></tr> ';
echo'<tr><td><font size="2" face="Arial">Temas:</font></td>';
echo'<td><font size="2" face="Arial"><textarea name="temas" rows="9"
cols="35">'.$te.'</textarea></font></td></tr>';
echo'<tr><td><font size="2" face="Arial">Resoluciones:</font></td>';
echo'<td><font size="2" face="Arial"><textarea name="resolu" rows="9"
cols="35">'.$re.'</textarea></font></td></tr>';
echo'<tr><td><font size="2" face="Arial">Documento: </font></td><td><p align="left"><font size="2"
face="Arial"> ';
echo'<input type="file" size="25" maxlength="8" name="link" value='.$li.'> </font></p></td></tr>';
echo'<tr><td> </td> <td><font size="2" face="Arial">';
echo '<br>';
echo'<input type="submit" value="Actualizar" >';
echo'<input type="reset" value="Recuperar"></font></td>';
echo'</tr>';
echo' </table>';
echo'</form>';
?>
</BODY>
</HTML>
```

PROGRAMA SECACT.PHP

```
<HTML>
<HEAD><TITLE>GRABACION DE CAMBIOS</TITLE></HEAD>
<BODY>
<?
//Conexion con la base
 mysql_connect("localhost","root");
 $id     = $HTTP_POST_VARS["id"];
 $fecha  = $HTTP_POST_VARS["fecha"];
 $temas  = $HTTP_POST_VARS["temas"];
 $resolu = $HTTP_POST_VARS["resolu"];
 $link   = $HTTP_POST_VARS["link"];
//Creamos la sentencia SQL y la ejecutamos
$sSQL="Update libro Set fecha='$fecha',temas='$temas',resolu='$resolu',link ='$link' Where id='$id'";
mysql_db_query("consejo",$sSQL);
?>
<center>
<font size="2" face="Arial"> Resolucion Actualizada </font>
</center>
</BODY>
</HTML>
```

PROGRAMA FORELI .HTP

```
<HTML>
<HEAD><TITLE>Eliminacion de Resoluciones de la PUCESA</TITLE></HEAD>
<BODY>
<div align="center">
<center><font size="2" face="Arial">ELIMINACION DE RESOLUCIONES</font></center>
<?
 //Conexion con la base
 mysql_connect("localhost","root");
 echo '<FORM METHOD="POST" ACTION="forpre.php">
 <center><font size="1" face="Arial">Escojer numero y fecha</font></center>
 <br>
 <br>';
 //Creamos la sentencia SQL y la ejecutamos
 $sSQL="Select id From libro Order by id";
 $result=mysql_db_query("consejo",$sSQL);
 //Mostramos los registros en forma de menú desplegable
 echo '<select name="id">';
 {echo '<option>'.'id'; }
 while ($row=mysql_fetch_array($result))
 {echo '<option>'.$row["id"];}
 echo '</select>';
 //Creamos la sentencia SQL y la ejecutamos
 $sSQL="Select fecha From libro Order by id";
 $result=mysql_db_query("consejo",$sSQL);
 //Mostramos los registros en forma de menú desplegable
 echo '<select name="fecha">';
 {echo '<option>'.'fecha'; }
 while ($row=mysql_fetch_array($result))
 {echo '<option>'.$row["fecha"];}
 mysql_free_result($result)
?>
 </select>
<INPUT TYPE="SUBMIT" value="Mostrar">
</FORM>
</div>
</BODY>
</HTML>
```

PROGRAMA FORPRE.PHP

```
<HTML>
<HEAD><TITLE>ACTUALIZACION DE RESOLUCION</TITLE></HEAD>
<BODY>
 <center> <font size="1" face="Arial Black">RESOLUCION A ACTUALIZAR </font>
 </center>
<?
 //Conexion con la base
 mysql_connect("localhost","root");
 echo '<FORM NAME= "form1" METHOD="POST" ACTION="secact.php">';
 //Creamos la sentencia SQL y la ejecutamos
 $id = $HTTP_POST_VARS["id"];
 $fecha = $HTTP_POST_VARS["fecha"];
 $sSQL="Select * From libro Where id='$id' and fecha='$fecha'";
 $result=mysql_db_query("consejo",$sSQL);
 while ($row=mysql_fetch_array($result))
 {  $i = $row["id"];
    $fe = $row["fecha"];
    $te = $row["temas"];
    $re = $row["resolu"];
```

```
    $li = $row["link"]; }
echo'<table border="0" align="center" bgcolor="#FFCC33">';
echo'<tr> <td><font size="2" face="Arial">Numero: </font></td> ';
echo'<td><p align="left"><font size="1" face="Arial"> ';
echo $i;
echo'</font></p> </td> </tr>';
echo'<input type="hidden" size="10" maxlength="35" name="id" value='.$i.'></font></td></tr> ';
echo'<tr> <td><font size="2" face="Arial">Fecha: </font></td> ';
echo'<td><font size="2" face="Arial"> ';
echo'<input type="date" size="10" maxlength="35" name="fecha" value='.$fe.'></font></td></tr> ';
echo'<tr><td><font size="2" face="Arial">Temas:</font></td>';
echo'<td><font size="2" face="Arial"><textarea name="temas" rows="9"
cols="35">'.$te.'</textarea></font></td></tr>';
echo'<tr><td><font size="2" face="Arial">Resoluciones:</font></td>';
echo'<td><font size="2" face="Arial"><textarea name="resolu" rows="9"
cols="35">'.$re.'</textarea></font></td></tr>';
echo'<tr><td><font size="2" face="Arial">Documento: </font></td><td><p align="left"><font size="2"
face="Arial"> ';
echo'<input type="file" size="25" maxlength="8" name="link" value='.$li.'> </font></p></td></tr>';
echo'<tr><td> </td> <td><font size="2" face="Arial">';
echo '<br>';
echo'<input type="submit" value="Actualizar" >';
echo'<input type="reset" value="Recuperar"></font></td>';
echo'</tr>';
echo' </table>';
echo'</form>';
?>
</BODY>
</HTML>
```

PROGRAMA SECELI.PHP

```
<HTML>
<HEAD>
  <TITLE>ELIMINACION DE RESOLUCIONES</TITLE>
</HEAD>
<BODY>
<?
//Conexion con la base
mysql_connect("localhost","root");
//Creamos la sentencia SQL y la ejecutamos
$id = $HTTP_POST_VARS["id"];
$fecha = $HTTP_POST_VARS["fecha"];
$sSQL="Delete from libro Where id='$id' and fecha='$fecha'";
mysql_db_query("consejo",$sSQL);
?>
<font size="0" face="Arial Black">
<div align="center">Resolucion Eliminada</div>
</font>
</BODY>
</HTML>
```

PROGRAMA: FORBUS.PHP

```
<HTML>
<HEAD><TITLE>MOTOR DE BUSQUEDAS</TITLE></HEAD>
<body>
<center><font size="2" face="Arial">MOTOR DE BUSQUEDAS</font></center>
<script language="JavaScript" type "text/javascript">
function validar()
{ if (datos.busca.value=="")
   {alert("favor ingresar frase a buscar")
     document.retorna = false;
    }
    else
     document.retorna = true;
}
</script>
<form action="seebus.php" method="POST" name="datos" onsubmit="validar(); return document.retorna">
  <table border="0" align="center">
     <tr>
       <td bgcolor="Orange"><font size="2" face="Arial" color=white>Tipo:</font></td>
       <td><font size="1" face="Arial">
        <input type="radio" checked name="op" value="tema" onclick="eliges(this.form)">Tema<br>
        <input type="radio" name="op" value="resolu" onclick="eliges(this.form)">Resolucion<br>
        <input type="radio" name="op" value="fecha" onclick="eliges(this.form)">Fecha<br>
        </font></td>
     </tr>
     <tr>
       <td bgcolor="Orange"><font size="2" face="Arial" color=white>Frase:</font></td>
       <td><font size="1" face="Arial">
       <input type="text" name="busca" size="40" maxlength="40">
       </textarea></font></td>
     </tr>
     <tr>
       <td bgcolor="Orange"><font size="2" face="Arial" color=white> Ordenado por:</font></td>
       <td><font size="1" face="Arial">
       <select name = "orden" size="1" style="width:100px;">
          <option value="id">Numero </option>
          <option value="fecha">Fecha </option>
          <option value="temas">Temas </option>
          <option value="resolu">Resoluciones </option>
       </select> </font></td>
     </tr>
     <tr>
        <td> <font size="1" face="Arial">
        </font></td>
     </tr>
     <tr>
       <td> </td>
       <td><font size="1" face="Arial">
       <br>
       <input type="submit" value="Buscar" onclick="validar();return document.retorna;">
       <input type="reset" value="Refrescar"></font></td>
     </tr>
  </table>
</FORM>
</BODY>
</HTML>
```

```
    $pag = '<a
href="pagina.php?pag='.$bit.'&sql='.$temp.'&tot='.$total.'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$d
o.'">'.$bit.'<a/>';
    $cin = 1;
    for ($c=1;$c<$total+1;$c++)
    {
     if ($cin > 5)
      {
       $bit++;
       $pag = $pag.' '.$pag = '<a
href="pagina.php?pag='.$bit.'&sql='.$temp.'&tot='.$total.'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$d
o.'">'.$bit.'<a/>';
       $cin=1;
      }
     $cin++;
    }
    echo '<center><strong><font color="#66CC33" size="2">P&#225;ginas:' ';
    echo $pag;
    echo '</center></strong>';
  }
  mysql_free_result($result);
?>
</table>
</form>
</font>
</BODY>
</HTML>
```

PROGRAMA SECCLI.PHP

```
<HTML>
<HEAD><TITLE>PRESENTACION DE INFORMACION</TITLE></HEAD>
<BODY>
<center>
<font size="2" face="Arial">INFORMACION BUSCADA
</center></font>
<?
 //Conexion con la base
 mysql_connect("localhost","root");
 echo '<FORM NAME= "form1">';
 //Creamos la sentencia SQL y la ejecutamos
 $id = $HTTP_GET_VARS["id"];
 $sSQL="Select * From libro Where id='$id'";
 $result=mysql_db_query("consejo",$sSQL);
 while ($row=mysql_fetch_array($result))
 {  $i = $row["id"];
    $fe = $row["fecha"];
    $te = $row["temas"];
    $re = $row["resolu"];
    $li = $row["link"];  }
 echo'<table border="0" bgcolor="#FF9944" align="center"> ';
 echo'<tr><td><font size="2" face="Arial" color="White">Numero:</font></td>';
 echo'<td><p align="left"><font size="2" face="Arial">';
 echo $i;
 echo'</font></p> </td> </tr>';
 echo'<tr><td><font size="2" face="Arial" color="White">Fecha: </font></td>';
 echo'<font size="2" face="Arial">';
 echo'<td><input type="date" size="10" readonly maxlength="37" name="fecha"
value='.$fe.'></font></td></tr>';
 echo'<tr><td><font size="2" face="Arial" color="White">Temas:</font></td>';
 echo'<td><font size="2" face="Arial"><textarea name="temas" readonly rows="10"
cols="35">'.$te.'</textarea></font></td></tr>';
 echo'<tr><td><font size="2" face="Arial" color="White">Resoluciones:</font></td>';
 echo'<td><font size="2" face="Arial"><textarea name="resolu" readonly rows="10"
cols="35">'.$re.'</textarea></font></td></tr>';
 echo'<tr><td><font size="2" face="Arial" color="White">Documento: </font></td>';
 echo'<td><p align="left"><font size="2" face="Arial">';
 echo'<input type="text" size="35" maxlength="8" readonly name="link" value='.$li.'> </font></p></td></tr>';
 echo'<font size="2" face="Arial">';
 echo '<br>';
 echo' </table>';
 echo'</form>';
?>
</BODY>
</HTML>
```

PROGRAMA PAGINA.PHP

```
<HTML>
<HEAD><TITLE>RESULTADOS DE LA BUSQUEDA</TITLE></HEAD>
<BODY>
<font size="3" face="Arial">
<center>RESULTADOS DE LA BUSQUEDA</center>
</font>
<font size="0" face="Arial">
<?
 //Creamos la sentencia SQL
  $page = $HTTP_GET_VARS["pag"];
  $sSQL = $HTTP_GET_VARS["sql"];
  $total = $HTTP_GET_VARS["tot"];
 //Conexion con la base
  mysql_connect("localhost","root");
 //Ejecutamos la sentencia SQL
  $result=mysql_db_query("consejo",$sSQL);
  echo '<table border="0" align="center">';
  echo '<td bgcolor="orange">';
  echo '<font color="White"> Se encontraron '.$total.' items con la frase buscada </font>';
  echo '</td>';
  echo '</table>';
  echo '<form>';
  echo '<table align="center" border="0" cellpadding="0" cellspacing="0">';
//Mostramos los registros
 $con = 1 + ((($page*5)-5);
 $cin = 1;
 $last = 0;
 if ($page > 1)
  {
    while ($row=mysql_fetch_array($result))
    {
     if  ($last > ($page*5)-7) break;
    $last ++;
    }
  }
  while ($row=mysql_fetch_array($result))
  {
    if ($con < ($page * 5)+1)
     {
      echo '<font color=black><p><strong>'.$con.'.-<a href="seccli.php?id='.$row["id"].'"> N&#250;mero:
'.$row["id"].' Fecha: '.$row["fecha"].' Documento: '.$row["link"].'</a></strong><br>';
      echo '<font color=green>'.substr($row["temas"],0,45).'<br>';
      echo '<font color=orange>'.substr($row["resolu"],0,45).'</p>';
     };
     $con ++;
  };
  $bit = 1;
  $pag = '';
  $temp = $sSQL;
   $pag = '<a
href="pagina.php?pag='.$bit.'&sql='.$temp.'&tot='.$total.'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$d
o.'">'.$bit.'<a/>';
   $cin = 1;
   for ($c=1;$c<$total+1;$c++)
   {
    if ($cin > 5)
     {
      $bit++;
```

```
        $pag = $pag.' '.$pag = '<a
href="pagina.php?pag='.$bit.'&sql='.$temp.'&tot='.$total.'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$d
o.'">'.$bit.'<a/>';
        $cin=1;
       }
       $cin++;
     };
     if ($page > 1)
     {
       $ant = $page - 1;
       $pag= '<a
href="page.php?pag='.$ant.'&sql='.$temp.'&tot='.$total.'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$do.'
">Anterior<a/>'.' '.$pag;
     };
     if ($page < $bit)
     {
      $sig = $page + 1;
      $pag = $pag.' '.'<a
href="page.php?pag='.$sig.'&sql='.$temp.'&tot='.$total.'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$do.'
">Siguiente<a/>';
     };
     if ($page == $bit)
     for ($i;$i<=(($bit*5)-$total)*3.9;$i++)
     echo '<br>';
     echo '<center><strong><font color="orange" size="2">P&#225;ginas:'.' ';
     echo $pag;
     echo '</center></strong>';
     mysql_free_result($result);
     echo '</table>';
     echo '</form>';
     echo '</font>';
?>
</BODY>
</HTML>
```

PROGRAMA FORCON.PHP

```
<HTML>
<HEAD><TITLE>MOTOR DE CONSULTAS</TITLE></HEAD>
<body>
<center><font size="2" face="Arial">MOTOR DE CONSULTAS</font></center>
<form action="forgen.php" method="POST">
  <table border="0" align="center">
    <tr>
      <td bgcolor="#660C33"><font size="2" face="Arial" color=white> Ordenado por:</font></td>
      <td><font size="1" face="Arial">
      <select name = "orden" size="1" style="width:100px;">
        <option value="id">Numero </option>
        <option value="fecha">Fecha </option>
        <option value="temas">Temas </option>
        <option value="resolu">Resoluciones </option>
      </select> </font></td>
    </tr>
    <tr> <td> </td>
    </tr>
    <tr>
      <td bgcolor="#660C33"><font size="2" face="Arial" color=white> Incluir columnas:</font></td>
      <td><font size="1" face="Arial">
      <input type="checkbox" checked name="fac" value="id">Numero   
      <input type="checkbox" checked name="fac1" value="fecha">Fecha  
      <input type="checkbox" checked name="fac2" value="temas">Tema  
      <input type="checkbox" checked name="fac3" value="resolu">Resolucion  
      <input type="checkbox" checked name="fac4" value="link">Documento 
      </font></td>
    </tr>
    <tr>
      <td> <font size="1" face="Arial"> 
      </font></td>
    </tr>
    <tr>
      <td> </td>
      <td><font size="1" face="Arial">
      <br>
      <input type="submit" value="Consultar">
      <input type="reset" value="Refrescar"></font></td>
    </tr>
  </table>
</form>
</body>
</html>
```

PROGRAMA FORGEN.PHP

```
<HTML>
<HEAD><TITLE>CONSULTA GENERAL DE RESOLUCIONES</TITLE></HEAD>
<BODY>
<font size="3" face="Arial"><center>RESULTADOS DE LA CONSULTA</center></font>
<font size="0" face="Arial">
<?
  $nu  = $HTTP_POST_VARS["fac"];
  $fe  = $HTTP_POST_VARS["fac1"];
  $te  = $HTTP_POST_VARS["fac2"];
  $re  = $HTTP_POST_VARS["fac3"];
  $do  = $HTTP_POST_VARS["fac4"];
  $ord  = $HTTP_POST_VARS["orden"];
  $link = mysql_connect("localhost","root");
  $sSQL= 'select * from libro order by '.$ord;
  $temp='select * from libro order by '.$ord;
  $result=mysql_db_query("consejo",$sSQL);
// Busca el numero de filas encontradas en la consulta
  $total = mysql_num_rows($result);
  echo '<table border="0" align="center">';
  echo '<td bgcolor="#66CC33">';
  echo '<font color="White"> Se encontraron '.$total.' filas en la Base de Consejo</font>';
  echo '</td>';
  echo '</table>';
  echo '<form>';
  echo '<table align="center" border="0" cellpadding="0" cellspacing="0">';
//Mostramos los registros
 $con = 1;
 $flag = 0;
 while ($row=mysql_fetch_array($result))
 {
      if ($con < 6)
       {
       echo '<font color=black><p><strong>'.$con.'-<a href="secone.php
?id='.$row["id"].'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$do.'"> N&#250;mero: '.$row["id"].' Fecha:
'.$row["fecha"].' Documento: '.$row["link"].'</a></strong><br>';

       echo '<font color=green>'.substr($row["temas"],0,45).'<br>';
       echo '<font color=orange>'.substr($row["resolu"],0,45).'</p>';
       } else $flag = 1;
       $con ++;
 }
 if ($flag == 1)
  {
   $bit = 1;
   $pag = '';
   $pag = '<a
href="page.php?pag='.$bit.'&sql='.$temp.'&tot='.$total.'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$do.'
">'.$bit.'</a>';
   $cin = 1;
   for ($c=1;$c<$total+1;$c++)
   {
    if ($cin > 5)
     {
      $bit++;
      $pag = $pag.' '.$pag = '<a
href="page.php?pag='.$bit.'&sql='.$temp.'&tot='.$total.'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$do.'
">'.$bit.'</a>';
      $cin=1;
```

```
        }
        $cin++;
    }
    echo '<center><strong><font color="#66CC33" size="2">P&#225;ginas:' ';
    echo $pag;
    echo '</center></strong>';
  }
  mysql_free_result($result);
?>
</table>
</form>
</font>
</BODY>
</HTML>
```

PROGRAMA SECONE.PHP

```
<HTML>
<HEAD><TITLE>RESULTADOS DE LA BUSQUEDA</TITLE></HEAD>
<BODY>
<font size="2" face="Arial"><center>RESULTADOS DE LA CONSULTA</center>
<?
  function how($c) {
     return strlen($c); }
  $i    = $HTTP_POST_VARS["fac"];
  $fec  = $HTTP_POST_VARS["fac1"];
  $tem  = $HTTP_POST_VARS["fac2"];
  $res  = $HTTP_POST_VARS["fac3"];
  $lin  = $HTTP_POST_VARS["fac4"];
  $ord  = $HTTP_POST_VARS["orden"];
  $bus = chop($bus);
  if (how($i)>0) $cad = $i;
  if (how($fec)>0)
        if(how($cad)>0) $cad = $cad ." , ". $fec;
        else $cad = $fec;
  if (how($tem)>0)
        if(how($cad)>0) $cad = $cad ." , ". $tem;
        else $cad = $tem;
  if (how($res)>0)
        if(how($cad)>0) $cad = $cad ." , ". $res;
        else $cad = $res;
  if (how($lin)>0)
        if(how($cad)>0) $cad = $cad ." , ". $lin;
        else $cad = $lin;
//Conexion con la base
   mysql_connect("127.0.0.1","","");
//Ejecutamos la sentencia SQL
   $sSQL='select '.$cad.' from libro order by '.$ord;
   $result=mysql_db_query("consejo",$sSQL);
   echo '<form>';
   echo '<table align="center" border=1 cellpadding="0" cellspacing="0">';
   echo '<tr>';
   echo '<font face=arial size=1>';
    if ($i - = "id")       echo '<th>Num</th>';
    if ($fec == "fecha")  echo '<th>Fecha</th>';
    if ($tem == "temas")  echo '<th>Temas</th>';
    if ($res == "resolu") echo '<th>Resoluciones</th>';
    if ($lin == "link")   echo '<th>Documento</th>';
   echo '</tr>';
//Mostramos los registros
  while ($row=mysql_fetch_array($result))
  {    echo '<tr>';
     if ($i == "id")       echo '<td><input type="text" readonly size="6" value="'.$row["id"].'"> </td>';
     if ($fec == "fecha")  echo '<td><input type="text" readonly size="9" value="'.$row["fecha"].'"> </td>';
     if ($tem == "temas")  echo '<td><textarea rows="8" readonly cols="35">'.$row["temas"].'</textarea> </td>';
     if ($res == "resolu") echo '<td><textarea rows="8" readonly cols="35">'.$row["resolu"].'</textarea> </td>';
     if ($lin == "link")   echo '<td><input type="text" readonly size="30"value="'.$row["link"].'"> </td></tr>';
     echo '<tr>';   }
  mysql_free_result($result)
?>
</font>
</table>
</form>
</BODY>
</HTML>
```

PROGRAMA HOJA.PHP

```
<HTML>
<HEAD>
<TITLE>RESULTADOS DE LA BUSQUEDA</TITLE>
</HEAD>
<BODY>
<font size="3" face="Arial">
<center>RESULTADOS DE LA BUSQUEDA</center>
<br>
</font>
<font size="0" face="Arial">
<?php
   $page = $HTTP_GET_VARS["pag"];
   $sSQL = $HTTP_GET_VARS["sql"];
   $total = $HTTP_GET_VARS["tot"];
   $ini = $HTTP_GET_VARS["hoj"];

   $nu  = $HTTP_GET_VARS["num"];
   $fe  = $HTTP_GET_VARS["fec"];
   $te  = $HTTP_GET_VARS["tem"];
   $re  = $HTTP_GET_VARS["res"];
   $do  = $HTTP_GET_VARS["doc"];

   mysql_connect("localhost","root");
   $result=mysql_db_query("consejo",$sSQL);

   echo '<table border="0" align="center">';
   echo '<td bgcolor="#66CC33">';
   echo '<font color="White"> Se encontraron '.$total.' filas en la Base de Consejo. P&#225;gina # '.$page.'</font>';
   echo '</td>';
   echo '</table>';

   echo '<form>';
   echo '<table align="center" border="0" cellpadding="0" cellspacing="0">';
   $con = 1 + ((($page*5)-5);
   $cin = 1;
   $last = 0;
   if ($page > 1)
   {
    while ($row=mysql_fetch_array($result))
    {
     if ($last > ($page*5)-7) break;
     $last ++;
    }
   }
   while ($row=mysql_fetch_array($result))
   {
     if ($con < ($page * 5) + 1)
      {
       echo '<font color= black><p><strong>'.$con.'.-<a
href="secone.php?id='.$row["id"].'&dat='.$row["fecha"].'&num='.$nu.'&fec='.$fe.'&tem='.$te.'&res='.$re.'&doc='.$do
.'"> N&#250;mero: '.$row["id"].' Fecha: '.$row["fecha"].' Documento: '.$row["link"].'</a></strong><br>';
       echo '<font color=green>'.substr($row["temas"],0,45).'<br>';
       echo '<font color=orange>'.substr($row["resolu"],0,45).'</p>';
      }
     $con ++;
```

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

ENCUESTA DE LA DISERTACIÓN DE GRADO DEL SISTEMA WEB DE RESOLUCIONES BAJO SISTEMA OPERATIVO LINUX DE LA PUCESA

ORIENTADA A LOS USUARIOS DEL SISTEMA

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	RESPUESTA	
	SI	NO
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	☑	☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	☑	☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	☑	☐
CUALES: - Escolastico - Personal Profesionales - Laboratorios - Autoridades - Biblioteca		
4.- ¿En la secretaria de su escuela cuentan con Internet?	☑	☐

"GRACIAS POR SU COLABORACIÓN"

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

ENCUESTA DE LA DISERTACIÓN DE GRADO DEL SISTEMA WEB DE RESOLUCIONES BAJO SISTEMA OPERATIVO LINUX DE LA PUCESA

ORIENTADA A LOS USUARIOS DEL SISTEMA

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	**RESPUESTA**	
	SI	**NO**
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	☒	☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	☒	☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	☒	☐
4.- ¿En la secretaria de su escuela cuentan con Internet?	☐	☒

5 Sist. autoevaluación por carreras

CUALES: Sist. Línea con las sedes

"GRACIAS POR SU COLABORACIÓN"

Pastoral.

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

ENCUESTA DE LA DISERTACIÓN DE GRADO DEL SISTEMA WEB DE RESOLUCIONES BAJO SISTEMA OPERATIVO LINUX DE LA PUCESA

ORIENTADA A LOS USUARIOS DEL SISTEMA

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	**RESPUESTA**	
	SI	**NO**
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	☑	☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	☑	☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	☑	☐

CUALES: Matriculación, Notas

4.- ¿En la secretaria de su escuela cuentan con Internet?	**SI**	**NO**
	☐	☑

"GRACIAS POR SU COLABORACIÓN"

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

ENCUESTA DE LA DISERTACIÓN DE GRADO DEL SISTEMA WEB DE RESOLUCIONES BAJO SISTEMA OPERATIVO LINUX DE LA PUCESA

ORIENTADA A LOS USUARIOS DEL SISTEMA

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	RESPUESTA	
	SI	NO
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	☒	☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	☒	☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	☒	☐

CUALES: Integración de los Programas Académicos de todas las Escuelas y Carreras, Calificaciones de los Estudiantes, y los Docen

4.- ¿En la secretaria de su escuela cuentan con Internet? SI ☐ NO ☒

"GRACIAS POR SU COLABORACIÓN"

Necesidades: Computadora, existe una anti... que y es lenta,

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

ENCUESTA DE LA DISERTACIÓN DE GRADO DEL SISTEMA WEB DE RESOLUCIONES BAJO SISTEMA OPERATIVO LINUX DE LA PUCESA

ORIENTADA A LOS USUARIOS DEL SISTEMA

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	**RESPUESTA**	
	SI	**NO**
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	☒	☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	☒	☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	☒	☐

CUALES: Sistema Administrativo Financiero Académico

4.- ¿En la secretaria de su escuela cuentan con Internet?	**SI**	**NO**
	☒	☐

"GRACIAS POR SU COLABORACIÓN"

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

ENCUESTA DE LA DISERTACIÓN DE GRADO DEL SISTEMA WEB DE RESOLUCIONES BAJO SISTEMA OPERATIVO LINUX DE LA PUCESA

ORIENTADA A LOS USUARIOS DEL SISTEMA

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	RESPUESTA	
	SI	NO
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	☒	☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	☒	☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	☒	☐

CUALES: .....

4.- ¿En la secretaria de su escuela cuentan con Internet?	SI	NO
	☒	☐

"GRACIAS POR SU COLABORACIÓN"

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

ENCUESTA DE LA DISERTACIÓN DE GRADO DEL SISTEMA WEB DE RESOLUCIONES BAJO SISTEMA OPERATIVO LINUX DE LA PUCESA

ORIENTADA A LOS USUARIOS DEL SISTEMA

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	**RESPUESTA**	
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	**SI** ☒	**NO** ☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	**SI** ☒	**NO** ☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	**SI** ☐	**NO** ☐

CUALES: .....

4.- ¿En la secretaria de su escuela cuentan con Internet?	**SI** ☐	**NO** ☒

"GRACIAS POR SU COLABORACIÓN"

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

SEDE AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

ENCUESTA DE LA DISERTACIÓN DE GRADO DEL SISTEMA WEB DE RESOLUCIONES BAJO SISTEMA OPERATIVO LINUX DE LA PUCESA

ORIENTADA A LOS USUARIOS DEL SISTEMA

(Marque con una X en SI ó NO según su criterio)

PREGUNTA	**RESPUESTA**	
1.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la consulta ágil y rápida de los Usuarios?	**SI** ☑	**NO** ☐
2.- ¿El Sistema de Resoluciones Implementado en el Sitio Web de la PUCESA permite la actualización de la Información?	**SI** ☑	**NO** ☐
3.- ¿Según su opinión, considera que sistemas de esta naturaleza deben implementarse en otros aspectos de la PUCESA?	**SI** ☐	**NO** ☐

CUALES: .....

4.- ¿En la secretaria de su escuela cuentan con Internet?	**SI** ☐	**NO** ☒

"GRACIAS POR SU COLABORACIÓN"

GLOSARIO

Address: (traducción literal: dirección). Este término se puede refererir a la dirección IP, o a una dirección de correo electrónico.

Ancho de banda: *(ing.: bandwidth)* La cantidad de información que se puede enviar a través de una conexión. Se mide normalmente en bits por segundo (bps). Una página completa de texto en inglés tiene aproximadamente 16.000 bits. Un módem rápido puede enviar alrededor de 15.000 bits en un segundo.

Applet: Aplicación realizada en Java para ser ejecutada en el sistema cliente.

Archie: Herramienta de software para encontrar archivos localizados en FTP anónimos. Es necesario conocer el nombre exacto del archivo para acceder a él.

ARPANet: Este término proviene de Advanced Research Project Agency Network (Red de Proyectos de Investigación Avanzados), el precursor de la red Internet. Fue desarrollado a finales de los años 60 y a comienzos de los 70 por el Departamento de Defensa americano, como un experimento en redes mundiales que sobreviviera una guerra nuclear.

ASCII: Acrónimo del código estándar americano para el intercambio de información (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange). Es el código estándar de conjunto de caracteres que cualquier ordenador puede entender, usado para representar las letras latinas, en mayúsculas, minúsculas, números, puntuación, etc. Hay 128 códigos estándar ASCII, cada uno de los cuales puede representarse por un número binario de 7 dígitos. Sin embargo, otros conjuntos de caracteres como Latin-1 están comenzando a usarse. Los documentos HTML no se limitan a ASCII.

Autenticación: *(ing.: authentification)* Este término se refiere a la acción de verificar la identidad de una persona o de un proceso.

Backbone: (traducción literal: espina dorsal). Línea o serie de conexiones de de alta velocidad que forman una ruta dentro de una red.

BITNET: (Because It's time NETwork) red de puntos separados de Internet, pero cuyo correo electrónico está en intercambio entre BITNET e Internet

Bookmark: (traducción literal: marcapáginas) Se utiliza este término para designar la característica que tienen algunos navegadores, como los de Netscape, de archivar la dirección URL de una página Web como si de una agenda se tratara. De esta manera, cuando queremos acceder a dicha página, basta con utilizar esta función, y nos conectaremos su dirección.

BOF: Acrónimo de Begin Of File. Vale TRUE si se esta en el primer registro y FALSE si no.

bps: Acrónimo de bits por segundo. Es la medida estándar de la velocidad de transmisión de datos a través de un módem.

Browser: (traducción literal: examinador) Programa cliente que se utiliza para buscar diferentes recursos de Internet. Se trata de una herramienta de navegación sin la cual no se podría acceder a los recursos de Internet. Los browsers más usados son Netscape Navigator, Microsoft Internet Explorer, y Mosaic de la NCSA.

Cabecera: *(ing.: header)* Este término se refiere a la información acerca de un documento Web o un mensaje de correo que se encuentra al principio del documento o mensaje. La información que contiene una cabecera puede hacer referencia al autor, o el generador del texto. No se debe confundir ese término con el de encabezamiento.

Caché: Almacenamiento local y temporal de un programa, de los mensajes de respuesta y el subsistema que controla el almacenamiento, la recuperación y eliminación de sus mensajes. Un caché, almacena respuestas para reducir el tiempo de respuesta y el consumo de anchobanda de red en demandas equivalentes futuras.

Cadena: *(ing.: string)* Secuencia de caracteres. Cada palabra es una cadena. Una búsqueda preguntará por una cadena de búsqueda, refiriéndose no sólo a palabras, sino a una secuencia de caracteres, formen éstos una palabra una frase, o ninguna de las dos posibilidades.

Ciberespacio: *(ing.: cyberspace)* Este término, que fue acuñado por primera vez por el escritor William Gibson en 1984, describe la gran variedad de recursos de información disponibles a través de las redes de ordenadores.

Cliente: *(ing.: client)* Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otro ordenador, a menudo a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente.

Conexión: *(ing.: connection)* Circuito virtual de transporte que se establece entre dos programas de aplicación con fines comunicativos.

Contraseña: *(ing.: password)* Palabra o cadena de caracteres, normalmente secreta, para acceder a través de una barrera. Se usa como herramienta de seguridad para identificar usuarios de una aplicación, archivo, o red. Puede tener la forma de una palabra o frase de carácter alfanumérico, y se usa para prevenir accesos no autorizados a información confidencial.

Cookies: (traducción literal: pequeño tesoro) Mecanismo de conexiones por parte del servidor, usado para almacenar y recuperar información en el cliente.

Correo electrónico: *(ing.: Electronic Mail, o e-mail).* Los mensajes, normalmente en forma de texto, enviados de una persona a otra sobre un tema en concreto a través del ordenador. El correo electrónico puede enviarse automáticamente a un gran número de direcciones a través de una lista de correo (mailing list).

Dirección IP: *(ing.: IP address)* Representación numérica de la localización de un ordenador dentro de una red. Consiste en cuatro números de hasta 4 cifras separados por puntos.

Download: Término prestado del inglés, cuya traducción literal significa «descargar». Se refiere a la acción de importar archivos de un ordenador remoto a otro local por medio de una conexión, como se puede realizar a través de un FTP. En el argot de Internet, se usa para describir esta acción, la frase «bajar de la red» o, simplemente «traerse» un archivo o un programa.

DNS: Acrónimo de Domain Name System (Sistema de Nombres de Dominio). Sistema para traducir los nombres de los ordenadores en direcciones IP numéricas.

Domain name: (traducción literal: nombre de dominio). Ver nombre de dominio.

E-mail: (Electronic Mail). Ver correo electrónico.

EOF: Acrónimo de End of File. Vale TRUE si sé esta en el último registro y FALSE sino. Se usa mucho como condición en bucles while, los cuales se ejecutan hasta llegar al último registro.

Ethernet: Un método muy común de comunicar ordenadores en una red LAN. Ethernet manejará 10 millones de bits por segundo y puede usarse con casi todos los tipos de ordenadores.

FAQ: Acrónimo de Frequently Asked Questions (Preguntas más frecuentes). Documento que contiene las preguntas de interés general más usuales acerca de un tema, con sus respuestas. Hay miles de FAQs sobre temas tan diversos como la Criptografía o el cuidado de los animales domésticos, sin olvidar, por supuesto aspectos de la red Internet, que son los más usuales.

Firewall: (Traducción literal: cortafuegos). Se trata de uno de los métodos usados para proteger una red de intrusiones no autorizadas. Esto se realiza a través de dos mecanismos: uno para bloquear el tráfico de la red, y otro para dejar fluir dicho tráfico, todo esto controlado para evitar accesos no deseados.

Freeware: Se llama así al software de dominio público, es decir, el que no es comercial y puede distribuirse gratuitamente, aunque no se puede modificar, pues el autor mantiene los derechos de copyright.

FTP: Acrónimo de «File Transfer Protocol» (Protocolo de transferencia de archivos), un método muy usado para transferir archivos de un ordenador a otro remoto a través de Internet. FTP es un modo especial de entrar en otro servidor de web en Internet, para enviar o transferir archivos. Mediante FTP se puede obtener no sólo archivos, sino también muchas aplicaciones, entrando en los servidores en los que éstas que se encuentran disponibles, usando el nombre de cuenta anónimo *(ing.: anonymous)*, por lo que estos servidores se llaman «servidores FTP anónimos».

Gateway: (Traducción literal: pasarela). Es un servidor que actúa como intermediario para otro servidor. Al contrario de lo que ocurre con un «proxy», un gateway recibe demandas como si fuera el servidor original para el recurso solicitado; el cliente demandante no es consciente de que está comunicándose con un gateway. Se usan normalmente como puertas del lado del servidor, a través de firewall de red como traductores de protocolo para acceder a recursos almacenados en sistemas que no son HTTP.

GIF: Acrónimo de **G**raphics **I**nterchange **F**ormat (Formato de intercambio de gráficos). Es un formato de archivos gráficos. En los documentos de WWW, IMG a menudo enlaza con archivos de extensión «.gif»

Gopher: Herramienta de búsqueda que presenta información en un sistema de menús jerárquicos parecidos a un índice. Se trata de un método de hacer menús de material disponible a través de Internet. El Gopher es un programa de estilo Cliente -Servidor, que requiere que el usuario tenga un programa cliente Gopher.
Aunque Gopher se extendió rápidamente por todo el mundo, ha sido sustituido en los últimos dos años por el Hipertexto., también conocido como WWW (World Wide Web), todavía hay miles de usuarios servidores de Gopher en la Internet.

GUI: Acrónimo de Graphical User Interface (Interface Gráfico de Usuario). Colección gráfica de iconos, carpetas, escritorio, cajas de diálogo, etc que se activan o desactivan por medio del ratón. En definitiva, se llama así a todo programa o aplicación que se ejecuta en un entorno gráfico, y cuyo interface es gráfico.

GPL: *(General Public Licence)* Se refiere a la licencia conocida como GNU en su versión 2 (GPL), que cubre la mayor parte del software de la **Free Foundation** y muchos mas programas.

Hipertexto: *(ing.: hypertext)* Cualquier texto que contiene enlaces a otros documentos. El lector selecciona determinadas palabras o frases en el documento, que están unidas a otro documento o parte del mismo mediante un enlace, lo que hace que aparezca el otro documento.

Host: (traducción literal: anfitrión) En una red local, ordenador que realiza todas las funciones de mantenimiento centralizadas, y pone a disposición de otros usuarios los programas y proporciona otros servicios. En Internet, se llama así a un ordenador conectado a la red, que tiene su propio número IP y nombre de dominio, y que sirve información a través de WWW.

Href: Marca de enlace usada en HTML, para designar la dirección de destino del enlace

HTML: Acrónimo de **H**iper**T**ext **M**arkup **L**anguage (Lenguaje de Marcas de Hipertexto). El lenguaje de código que se usa para crear documentos de hipertexto para usar en la WWW. HTML parece un código pasado de moda, donde un bloque de texto se rodea con códigos que indican cómo debería parecer. Sin embargo, en HTML también se pueden especificar que un bloque de texto o una palabra se una a otra fila en la Internet. Archivos HTML están dirigidos a ser visualizados usando un programa cliente WWW, como Mosaic.

HTTP: Acrónimo de HyperText Transport Protocol (Protocolo de Transporte de Hipertexto). Protocolo para mover archivos de hipertexto a través de la Internet. Requiere un programa cliente HTTP en un extremo y un programa servidor de HTTP en el otro. HTTP es el protocolo más importante usado en el WWW.

IMG: Abreviatura de «image». Se usa para designar un enlace con un archivo gráfico. Sólo se pueden visualizar con browsers gráficos, que en ocasiones permiten desactivar el proceso de las imágenes, si va a ocupar mucho tiempo.

Internet: La gran colección de redes interconectadas que usan protocolo TCP/IP y que evolucionó de ARPANET a finales de los 60 y principios de los 70. Internet conecta hoy por hoy a 60.000 redes independientes dentro de la red mundial global.

IP: *(ing.: Internet Protocol)* Protocolo de Internet. Es un número dividido en cuatro parte separadas por puntos. Cada ordenador tiene un sólo número, si no lo tiene, no está realmente en Internet. La mayoría de ordenadores tienen uno o más nombres de dominio que son más fáciles de recordar.

ISDN: Acrónimo de Integrated Services Data Network. Ver RDSI

Java: Nuevo lenguaje de programación orientado a objetos, diseñado por Sun Microsystems para el desarrollo de aplicaciones multiplataforma y para la WWW. Se puede describir como una versión simplificada de C++. Además, Java implementa muchas características de seguridad en tiempo de compilación y de ejecución, para asegurar la aplicación que se ejecuta. Pero la novedad de este lenguaje es que es independiente de la plataforma cliente, y las applets se ejecutan en el sistema cliente.

JavaScript: Lenguaje derivado de Java, pero con instrucciones mucho más simples.

Jpeg: Acrónimo de «Joint Photographic Experts Group» (Grupo de expertos fotógrafos) Es, como gif, un formato para archivos gráficos, y un estándar para imágenes en Web. Los archivos de este formato tienen extensión «.jpg». Lo que diferencia a los formatos gif y jpeg es cómo se comprimen los datos (con pérdidas en jpeg, y sin pérdidas en gif) y la profundidad de bits (8 bits para gif, y 24 bits para jpeg).

LAN: Acrónimo de Local Area Network (Red de Área Local). Red de ordenadores limitada a un área inmediata, que es normalmente el mismo edificio o piso de un edificio.

Login: Nombre que se usa para acceder a un sistema de ordenadores. No es secreta, si lo fuera sería una password (clave). Acción de entrar en un sistema de ordenadores.

Mensaje: *(ing.: message)* La unidad básica de la comunicación HTTP, consistente en una secuencia estructurada de octetos que se ajustan a la sintaxis y transmitidos por medio de la conexión.

Mosaic: El primer visualizador (browser) de WWW disponible para Macintosh, Windows y UNIX con el mismo dispositivo. «Mosaic» comenzó la popularidad del Web. El código fuente de Mosaic ha sido tomado por varias empresas. Hay otros programas tan buenos o mejores que Mosaic, siendo el más notable «Netscape».

NetBEUI: Este término proviene de **NetB**IOS **E**xtended **U**ser **I**nterface. Se trata de un controlador de dispositivo de red. Es el controlador de transporte proporcionado con LAN Manager (Administrador de Red Local de Microsoft), y es el protocolo de comunicación entre redes LAN.

Nodo: Cualquier ordenador conectado a una red.

Nombre de dominio: *(ing.: domain name)* Nombre que identifica el punto de Internet. Los nombres de dominio tienen dos o más partes, separadas por puntos. La parte de la izquierda es la más específica, mientras que la de la derecha es la más general. Un ordenador puede tener más de un nombre de dominio, pero un determinado nombre de dominio sólo se refiere a una máquina. Normalmente, todos los ordenadores de una red tendrán el mismo nombre que la parte derecha de sus nombres de dominio. Es posible que un nombre de dominio exista pero no esté conectado a un ordenador. Esto ocurre a menudo, de tal modo que un grupo o empresa puede tener una dirección de correo electrónico sin tener que establecer un punto real en Internet.

Nombre de usuario: *(ing.: userid, username)* El nombre que utiliza el usuario cuando accede a otro ordenador.

Operadores lógicos: *(ing.: logic operators).* También llamados operadores booleanos (del álgebra de Bool), se usan en los buscadores para restringir una búsqueda y eliminar resultados no deseados. Son los operadores AND (y), para añadir un término; NOT (no) para excluirlo, y OR (o) para hacer una elección.

OSI: Acrónimo de Open System Interconnection (Interconexión de sistemas abiertos). El modelo de referencia OSI proporciona la base para el desarrollo de estándares relativos a las redes. Este modelo enumera siete capas que definen las actividades que deben tener lugar cuando se comunican los dispositivos a través de una red. Estas siete capas (de arriba a abajo) son: aplicación, presentación, sesión, transporte, red, enlace y física.
El modelo representa las relaciones entre una red y los servicios que puede soportar como una jerarquía de capas de protocolos. Cada capa usa los servicios ofrecidos por capas más bajas además de sus propios servicios para crear otros nuevos que estén disponibles para capas superiores. En resumen, cada una de las siete capas del modelo de referencia OSI realiza tareas únicas y específicas, conoce las capas inmediatamente adyacentes, usa los servicios de la capa que está por debajo, y realiza funciones y proporciona servicios para las capas superiores.

Página de inicio: *(ing.: home page)* También llamada página principal, es la primera página que aparece cuando se accede a un servidor de páginas Web, y es desde donde se puede explorar dicho servidor.

Página Web: *(ing.: Web page)* No se trata de un página en el sentido literal, sino un documento completo editado en la World Wide Web. La página principal (en inglés home page) es la primera página que aparece cuando se entra en un puesto de Web al que se ha llamado.

POP: Acrónimo de Post Office Protocol (Protocolo de Oficina de Correo). Protocolo para almacenar y recibir correo electrónico. Algunos programas de correo electrónico usan este protocolo, como Eudora.

PPP: Acrónimo de Point to Point Protocol (Protocolo de Punto por Punto). Es más conocido como el protocolo que permite que un ordenador use una línea telefónica regular y un módem para realizar conexiones TCP/IP.

Proveedor de servicio: *(ing.: provider)* Empresa que proporciona acceso a Internet, o a servicios de correo electrónico, FTP, Gopher, etc, por medio de una tarifa mensual.

Protocolo: *(ing.: protocol)* Lista de comandos estandarizada a la que responde un servidor. Un servidor FTP, por ejemplo, debe responder a comandos como «get» y «put».

Proxy: Programa intermediario que actúa a la vez como servidor y cliente para realizar demandas de otros clientes. Las demandas se tratan o bien de manera interna o pasándolas, con posible traducción, a otros servidores.
Un proxy debe interpretar y, si es necesario, reescribir un mensaje de pedido antes de enviarlo. Se usan normalmente como portales por parte del cliente a través de muros de contención y aplicaciones de ayuda para manejar demandas vía protocolos no implementados por el agente de usuario. Los servidores proxy implementan el rendimiento del servidor, al servir las páginas de manera local en una «cache».

Puerto: *(ing.: port)* Se llama así a un lugar donde la información entra o sale de un ordenador o ambas cosas. Por ejemplo, el «puerto serie» de un ordenador es donde se conectaría un módem. En Internet, puerto también se refiere a menudo a un número que es parte del URL, apareciendo tras el signo «:», justo después del nombre de dominio.

RDSI: Acrónimo de Red Digital de Servicios Integrados. Red especial diseñada para manejar más que datos: video, texto, voz, datos, imágenes, gráficos, etc, usando líneas telefónicas existentes. En inglés, esto se llama ISDN (Integrated Services Data Network).

Recurso: *(ing.: resource)* Objeto de datos de red o servicio que puede identificarse por un URI. Se llama así a la información que se encuentra en Internet, ofrecida por los servidores.

Red: *(ing.: network).* Grupo de ordenadores y otros dispositivos periféricos conectados unos a otros para comunicarse y transmitir datos entre ellos.

Router: (Traducción literal: encaminador) Ordenador con fines especiales (o paquete de software) que maneja la conexión entre dos o más redes. Los routers usan su tiempo mirando las direcciones de destino de los paquetes, pasando a través de ellas y decidiendo por que ruta enviarlos.

SCSI: Acrónimo de Small Computer System Interface. Es un interfaz del estándar ANSI (American National Standards Institute), para la comunicación en alta velocidad de datos paralelos entre ordenadores y sus dispositivos periféricos. La velocidad que proporciona el interfaz SCSI es un elemento importante que permite conectar hasta siete dispositivos SCSI diferentes en una conexión. Se lee «escasi».

Search Engine: (Motor de búsqueda) Herramienta que realiza búsquedas en sus propias bases de datos desde el ordenador cliente. También llamados buscadores, estas aplicaciones son muy útiles para navegar por Internet, pues nos indican dónde encontrar la información necesaria, pudiendo ir a ese lugar mediante un enlace.

SGML: Acrónimo de Standard Generalized Markup Language (Lenguaje Estándar de Marcas Gerenalizado).Conjunto de estándares usados para unir los elementos de un documento electrónico, para facilitar su producción en distintos medios. Este lenguaje es el padre del HTML, con el que se construyen los documentos en hipertexto.

Shareware: Software protegido por leyes de copyright, que se encuentra disponible gratuitamente durante cierto tiempo para su evaluación por el usuario. Tras pasar dicho tiempo, el programa expira y no podrá volver a ser utilizado, a no ser que el usuario registre el programa por un precio. Comparar con freeware.

SLIP: Acrónimo de Serial Line Internet Protocol (Protocolo de Internet de Línea en serie). Es un estándar para usar una línea telefónica y un módem para hacer de un ordenador un servidor de Internet. Este protocolo está siendo reemplazado de a poco por PPP.

SMTP: Acrónimo de Simple Mail Transfer Protocol

Socket: Número de identificación compuesto por dos números: la dirección IP y el número de puerto TCP. En la misma red, el número IP es el mismo, mientras que el número de puerto es el que cambia. En máquinas de distintas redes, pueden tener el mismo número de puerto sin llevar a confusión, pues el número IP las distingue.

SSL: Acrónimo de Secure Socket Layer (Capa de toma de corriente segura) Protocolo de bajo nivel utilizado par encriptar transacciones en un protocolo de mayor nivel como el HTTP, FTP y NNTP, entre clientes y servidores

T1: Conexión por medio de línea telefónica que transporta datos con velocidades de hasta 1.544.000 bps. Aunque no es lo suficientemente rápida para soportar vídeo con movimiento a pantalla completa en tiempo real, es ésta la velocidad más usada para conectar redes en Internet.

T3: Conexión por medio de línea telefónica que puede transportar datos a 45.000.000 bps.

TCP/IP: Acrónimo de Transmission Control Protocol/Internet Protocol (Protocolo de Internet/Protocolo de Control de Transmisión). Es el tipo de protocolos que define la Internet. Diseñado originalmente por el sistema operativo UNIX, el software TCP/IP no está disponible para la mayor parte de los sistemas operativos. Para acceder a Internet, el ordenador debe tener software TCP/IP.

TELNET: Permite la conexión remota a otros ordenadores de Internet. El comando y programa usado para acceder de un lugar a otro en Internet.

Túnel: *(ing.: tunnel)* Programa intermedio que actúa como un repetidor ciego entre dos conexiones. Una vez que está activo, un túnel no se considera parte de la comunicación HTTP, aunque el túnel puede haber sido iniciado o por una demanda HTTP. Se cierra cuando ambos extremos de las conexiones se cierran. Los túneles se usan cuando se necesita un portal, y el intermediario no puede, o no debe interpretar la comunicación transmitida.

UDP: Acrónimo de User Datagram Protocol (Protocolo de datagrama a nivel de usuario), perteneciente a la familia de protocolos TCP/IP. Este protocolo no es tan fiable como TCP, pues se limita a recoger el mensaje y enviar el paquete por la red. Para garantizar el éxito de la transferencia, UDP hace que la máquina de destino envía un mensaje de vuelta. Si no es así, el mensaje se envíe de nuevo. Con este protocolo no se establece una conexión entre las dos máquinas

Unix: Sistema operativo diseñado para ser usado por mucha gente al mismo tiempo (es multiusuario) y tiene TCP/IP. Es el sistema operativo más común para servidores en Internet.

URL: Acrónimo de Uniform Resource Locator (Localizador de Recursos Uniforme). Es el modo estándar de proporcionar la dirección de cualquier recurso en Internet, que es parte de la WWW.
Las URLs pueden ser absolutas o relativas. Una URL absoluta consiste en un prefijo que denota un método («http» para puntos de Internet, «gopher» para gophers, «ftp» para transferencia de ficheros, etc.), seguido por dos puntos y dos barras (://), una dirección, que consiste en un nombre de dominio, seguido por una barra, un nombre de vía, y un ancla opcional (precedido por un símbolo * que apunta a un lugar dentro de una página web). Una URL relativa designa un elemento relativo en el que la designación se hace. Es similar a dar el número de teléfono sin el prefijo de provincia para llamar desde la misma ciudad.

Usenet: Sistema mundial de grupos de discusión con comentarios que pasan entre cientos de miles de ordenadores. No todos los ordenadores Usenet están en Internet, ya que Usenet está completamente descentralizada, con más de 10.000 áreas de discusión, llamadas newsgroups.

Userid: Identificación de usuario.

Username: Ver nombre de usuario.

WAN: Acrónimo de Wide Area Network (Red de Área Extendida. Una red que cubre un área más grande que un sólo edificio.

WAIS: Acrónimo de Wide Area Information Servers (Servidores de Información de Área Extendida) Paquete de software comercial que permite indizar grandes cantidades de información y hacer que esos índices puedan buscarse a través de Internet. Una característica primordial de WAIS es que los resultados de búsqueda están medidos de acuerdo a lo relevantes que son, y otras búsquedas subsiguientes.

WWW: Acrónimo de World Wide Web (traducción literal: tela de araña mundial). Se denomina así al gran universo de recursos a los que se puede acceder usando Gopher, FTP, HTTP, Telnet, Usenet, WAIS y otras herramientas. Es el universo de servidores HTTP que permiten mezclar texto, gráficos, archivos de sonido, etc juntos.

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
PROGRAMAS ACADEMICOS AMBATO
BIBLIOTECA