



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA ELECTRÓNICA Y
TECNOLOGÍA EN COMPUTACIÓN**

**“IMPLEMENTACION DE UN SITIO WEB ACADÉMICO PARA EL COLEGIO
MILITAR N° 6 COMBATIENTES DE TAPP”.**

MEMORIA DE GRADO

Previa a la obtención del título de:

TECNÓLOGO EN INFORMÁTICA APLICADA

Presentado por:

VINICIO LUIS SILVA ESCOBAR

RIOBAMBA – ECUADOR

2009

Agradezco a todas aquellas personas que de un modo u otro, han contribuido a la realización de esta monografía.

Una gratitud especial para los profesores de la Escuela que me han brindado su ayuda en la elaboración del proyecto y en especial al **Ingeniero Edwin Altamirano**, director de la memoria de grado, además y a mis compañeros por su continua colaboración y ayuda desinteresada.

Dedico este trabajo a mis padres por ser el pilar fundamental, que con todo su amor y comprensión han sabido guiarme por el camino correcto, a mis hermanos y a mi hija Mabel, que son quienes me dan fortaleza, cariño, motivándome siempre para salir adelante; y a todos mis amigos por brindarme su apoyo incondicional.

NOMBRE	FIRMA	FIRMA
Dr. Romeo Rodríguez DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA.	_____	_____
Ing. Paul Romero. DIRECTOR DE ESCUELA. ING. ELECTRÓNICA Y TLGIA. EN COMPUTACIÓN	_____	_____
Ing. Edwin Altamirano DIRECTOR DE TESIS.	_____	_____
Ing. Geovanny Vallejo MIEMBRO	_____	_____
Tlgo. Carlos Rodríguez DIRECTOR DEL CENTRO DE DOCUMENTACIÓN.	_____	_____
NOTA ESCRITA:	_____	

Yo: Vinicio Luis Silva Escobar, soy responsable de las ideas, doctrinas y resultados expuestos en esta Memoria, y el patrimonio intelectual de la Memoria de Grado, pertenecen a la **Escuela Superior Politécnica de Chimborazo**.

.....

Vinicio Silva.

INTRODUCCIÓN

En la actualidad la gran mayoría de las empresas e instituciones, y en especial las educativas están implementando nuevos sistemas informáticos con una nueva y mejorada tecnología, que automatice sus antiguos sistemas manuales, trayendo consigo la eminente disminución de tiempo y recursos en las actividades institucionales, a más de ello es imprescindible que esta información sea suministrada por la institución hacia sus usuarios; para poder articular en forma óptima los procesos educativos. Además de ello los estamentos gubernamentales de control requieren de la publicación de la información financiera de las instituciones educativas a través del Internet.

Para ello es necesario combinar perfectamente las cualidades y adelantos de los equipos de cómputo con las habilidades de aprendizaje y acoplamiento humano, aplicando modernas técnicas para el desarrollo de Aplicaciones Web que se adapten adecuadamente a las necesidades tanto de los administradores del portal como a los usuarios en general, permitiendo así que la Institución se involucre más competitivamente al mundo automatizado de la información.

Es así que el Colegio Militar N°6 “Combatientes de Tapi “, al ser una de las instituciones educativas más importantes y representativas del la ciudad de Riobamba, ha visto necesario la innovación de sus procesos educativos, y para ello el desarrollo de una página web para la promoción de la Institución y su gestión académica.

CAPÍTULO I

ELEMENTOS DE DISEÑO WEB

1.1 WWW

World Wide Web (WWW): Es un sistema de información, el sistema de información propio de Internet. Sus características son:

- Información por hipertexto: Diversos elementos (texto o imágenes) de la información que se nos muestra en la pantalla están vinculados con otras informaciones que pueden ser de otras fuentes. Para mostrar en pantalla esta otra información bastará con hacer clic sobre ellos.
- Gráfico: En la pantalla aparece simultáneamente texto, imágenes e incluso sonidos.
- Global: Se puede acceder a él desde cualquier tipo de plataforma, usando cualquier navegador y desde cualquier parte del mundo.
- Pública: Toda su información está distribuida en miles de ordenadores que ofrecen su espacio para almacenarla. Toda esta información es pública y toda puede ser obtenida por el usuario.
- Dinámica: La información, aunque esta almacenada, puede ser actualizada por el que la publica sin que el usuario deba actualizar su soporte técnico.

- Independiente: Dada la inmensa cantidad de fuentes, es independiente y libre.

1.2 HTML

El HTML (Hyper Text Markup Language), es un lenguaje que permite estructurar documentos que pueden ser visualizados por visores de páginas Web en Internet. No existe un estándar en este lenguaje porque cada software de las diferentes empresas incluye directivas que solamente funcionan en sus visores.

Consta de una serie de órdenes o directivas que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido.). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa.

1.2.1 ESTRUCTURA BÁSICA DE UN DOCUMENTO HTML

Un documento escrito en HTML contendría básicamente las siguientes directivas:

<code><HTML></code>	Inicio del documento.
<code><HEAD></code>	Inicio de la cabecera.
<code><TITLE></code>	Inicio del título del documento.
<code></TITLE></code>	Final del título del documento.
<code></HEAD></code>	Final de la cabecera del documento.
<code><BODY></code>	Inicio del cuerpo del documento.
<code></BODY></code>	Final del cuerpo del documento.
<code></HTML></code>	Final del documento.

El documento se hallará situado en algún host al que se pueda acceder a través de Internet. Para indicar la situación del documento en Internet se utiliza los URL. Por ejemplo:

`http://www.comil.edu.ec/alumno/galería.htm`

Donde:

<i>http://</i>	Indicador de página Web
<i>comil.edu.ec</i>	Dominio (nombre) del ordenador
<i>/alumno/</i>	Directorio dentro del ordenador
<i>galería.html</i>	Fichero que contiene la página Web

1.2.2 División de un Documento HTML

a) Cabecera del documento

`<HEAD><HEAD>` Delimita la cabecera del documento, dentro de la cabecera es importante definir el título de la página.

`<TITLE><TITLE>` Define el título de la página, él que aparecerá en la barra de nuestro visor de páginas Web.

Ejemplo:

`<TITLE>Aplicación HTML<TITLE>`

Se tiene otras directivas adicionales dentro de la cabecera del documento:

`<META>` Indica al visor de Internet las palabras claves y contenido de nuestras páginas Web. Esta directiva lleva adicionalmente dos parámetros, *name* y *content*.

Ejemplos:

```
<META name = "Página de Bertha y Oscar" content = "Texto, Imágenes y  
Música">
```

Indica al visor el nombre de la página y sus contenidos principales.

Otro uso de esta directiva es el de indicar un refresco automático en los documentos. Si se indica un URL se sustituirá el documento por el indicado una vez transcurridos el número de segundos especificados. Si no se incluye ningún URL se volverá a cargar en el visor el documento en uso transcurrido los minutos indicados.

```
<META http-equiv= "refresh" content =  
"10;URL=http://www.vrml.org/VRML2.0/">
```

Transcurridos 10 segundos se accederá a la página Web de VRML2.0

<BASE> Indica la localización de los ficheros, gráficos, sonidos, etc. a los que se hace referencia en nuestra página Web. Si no se incluye esta directiva el visor entiende que dichos elementos se encuentran en el mismo lugar donde se encuentra nuestra página Web.

b) Cuerpo del documento

<BODY></BODY> Indica el inicio y final de la página Web. Será entre el inicio y el final de esta directiva donde se pone los contenidos de la página, textos, gráficos, enlaces, etc. Esta directiva tiene una serie de parámetros opcionales que permite indicar la "apariencia" global del documento:

background = "nombre de fichero gráfico" Indica el nombre de un fichero gráfico que servirá como "fondo" de nuestra página. Si la imagen no rellena todo el fondo del documento, esta será reproducida tantas veces como sea necesario.

bgcolor = "código de color" Indica un color para el fondo de nuestro documento. Se ignora si se ha usado el parámetro background.

<i>text = "código de color"</i>	Indica un color para el texto que incluya en el documento. Por defecto es negro.
<i>link = "código de color"</i>	Indica el color de los textos que dan acceso a un Hyperenlace. Por defecto es azul.
<i>vlink = "código de color"</i>	Indica el color de los textos que dan acceso a un Hyperenlace que ya se ha visitado con el visor. Por defecto es púrpura.

El código de color es un número compuesto por tres pares de cifras hexadecimales que indican la proporción de los colores "primarios", rojo, verde y azul. El código de color se antecede del símbolo #.

Ejemplos :

#000000	Color Negro
#FF0000	Color Rojo
#00FF00	Color Verde
#0000FF	Color Azul
#FFFFFF	Color Blanco

El primer par de cifras indican la proporción de color Rojo, el segundo par de cifras la proporción de color Verde y las dos últimas de color Azul. Cada par de cifras hexadecimales permiten un rango de 0 a 255. Combinando las proporciones de cada color primario se obtendrá diferentes colores.

De cualquier forma la mayoría de los editores de HTML permiten obtener el código de color correspondiente escogiendo directamente el color de una paleta.

1.2.3 ELEMENTOS DE UN DOCUMENTO HTML

1.2.3.1 Espaciados y saltos de línea

En HTML sólo se reconoce un espacio entre palabra y palabra, el resto de los espacios son ignorados por el visor.

Tampoco se respetan las tabulaciones, retornos de carro etc. Para ello existen una serie de directivas que indican estos códigos.

<PRE></PRE> obliga al visor a visualizar el texto tal y como han sido escritos, respetando tabulaciones, espacios, retornos de carro, etc.

**
** Para indicar un salto de línea.

<P> Para un cambio de párrafo (deja una línea en blanco en medio).

<P> Esta directiva puede usarse también como directiva "cerrada" **<P></P>** indicando de esta manera los atributos de un párrafo en concreto. Cuando se usa de esta manera tiene el parámetro *align* que indica al visor la forma de "justificar" el párrafo. Los valores posibles de este parámetro son LEFT, RIGHT, CENTER y JUSTIFY.

<HR> Muestra una línea horizontal de un tamaño determinado. Tiene los siguientes parámetros opcionales:

align = posición Alinea la línea a la izquierda (left), a la derecha (right) o al centro (center).

noshade No muestra sombra, evitando el efecto en tres dimensiones.

size = número Indica el grosor de la línea en píxeles.

width = num / % Indica el ancho de la línea en tanto por ciento en función del ancho de la ventana del visor. También se puede especificar un número que indicaría el ancho de la línea en píxeles.

Ejemplo:

<HR align= center size= 20 width= 50%>

La directiva <HR> sin ningún parámetro mostrará una línea horizontal que ocupará todo el ancho de la página

1.2.3.2 Cabeceras

En un documento de HTML se pueden indicar seis tipos de cabeceras (tamaños de letra) por medio de las directivas <H1><H2><H3><H4><H5> y <H6>. El texto que se escriba entre el inicio y el fin de la directiva será el afectado por las cabeceras. La cabecera <H1> será la que muestre el texto en mayor tamaño.

Los textos marcados como "cabeceras" provocan automáticamente un retorno de carro sin necesidad de incluir la directiva
.

1.2.3.3 Atributos del Texto

Para indicar atributos del texto (negrilla, subrayado, etc.) se tiene varias directivas. Algunas de ellas no son reconocidas por determinados visores de Internet, es por ello que según el visor que esté utilizando, verá el resultado correctamente o no.

 Permite cambiar el tamaño, el color, y el tipo de letra de un texto determinado. Utiliza para ello los parámetros *size*, *bgcolor* y *face*.

<i>size = valor</i>	Da al texto un tamaño en puntos determinado.
<i>size = +/- valor</i>	Da al texto un tamaño tantas veces superior (+) o inferior (-) como indique el valor.
<i>color = "código de color"</i>	Escribe el texto en el color cuyo código se especifica.
<i>face = "nombre de</i>	Escribe el texto en el tipo de letra especificado. Si este tipo de letra no existe en el ordenador que "lee" la página usará el font

font" predeterminado del navegador.

Existen otras directivas que realizan las mismas operaciones que las antes vistas en los atributos del texto.

Para incluir comentarios en la página Web se utiliza la directiva `<!-- -->`.

Los comentarios no serán mostrados por el visor y son útiles para realizar anotaciones en el documento HTML que indican lo que se está haciendo en una determinada parte del documento. La directiva de comentario es útil para incluir código JavaScript en nuestra página Web.

1.2.3.4 Listas de elementos

Existen tres tipos de listas, numeradas, sin numerar y de definición.

Las listas numeradas representarán los elementos de la lista numerando cada uno de ellos según el lugar que ocupan en la lista. Para este tipo de lista se utiliza la directiva ``. Cada uno de los elementos de la lista irá precedido de la directiva ``. La directiva `` puede llevar los siguientes parámetros:

start = Indica que número será el primero de la lista. Si no se indica se entiende que empezará por el número 1.

type = tipo Indica el tipo de numeración utilizada. Si no se indica se entiende que será una lista ordenada numéricamente.

Los tipos posibles son:

I = Numéricamente. (1,2,3,4,... etc.)

a = Letras minúsculas. (a,b,c,d,... etc.)

A = Letras mayúsculas. (A,B,C,D,... etc.)

i = Números romanos en minúsculas. (i,ii,iii,iv,v,... etc.)

I = Números romanos en mayúsculas. (I,II,III,IV,V,... etc.)

**** Se utiliza esta directiva para delimitar la lista.

**** Para indicar cada uno de los elementos.

La directiva **** puede contener el parámetro:

Las listas de definición muestran los elementos tipo diccionario, o sea, término y definición. Se utiliza para ellas la directiva **<DL></DL>**. El elemento marcado como término se antecede de la directiva **<DT>**, el marcado como definición se antecede de la directiva **<DD>**.

Existen otros dos tipos de listas menos comunes. Las listas de Menú o Directorio se comportan igual que las listas sin numerar. La lista de Menú utiliza la directiva **<MENU></MENU>** y los elementos se anteceden de ****. El resultado es una lista sin numerar más "compacta", es decir, con menos espacio interlineal entre los elementos. La lista de Directorio utiliza la directiva **<DIR></DIR>** y los elementos se anteceden de ****. Los elementos tienen un límite de 20 caracteres.

Todas las listas se pueden "anidar", es decir, incluir una lista dentro de otra, con lo que se consigue una estructura tipo "índice de materias".

1.2.3.5 Tablas

Las tablas permiten presentar cualquier elemento de nuestra página (texto, listas, imágenes, etc.) en diferentes filas y columnas separadas entre sí. Es una herramienta muy útil para "ordenar" contenidos de distintas partes de nuestra página. La tabla se define mediante la directiva **<TABLE></TABLE>**. Los parámetros opcionales de esta directiva son:

<i>border = número</i>	Indica el ancho del borde de la tabla en puntos.
<i>cellspacing = número</i>	Indica el espacio en puntos que separa las celdas que están dentro de la tabla.
<i>cellpadding = número</i>	Indica el espacio en puntos que separa el borde de cada celda y el contenido de esta.
<i>width = número ó %</i>	Indica la anchura de la tabla en puntos o en porcentaje en función del ancho de la ventana del visor. Si no se indica este parámetro, el ancho se adecuará al tamaño de los contenidos de las celdas.
<i>height = número ó %</i>	Indica la altura de la tabla en puntos o en porcentaje en función del alto de la ventana del visor. Si no se indica este parámetro, la altura se adecuará a la altura de los contenidos de las celdas.
<i>bgcolor = código de color</i>	Especifica el color de fondo de toda la Tabla.

Tabla I.1: Etiquetas para tablas en html

Para definir las celdas que componen la tabla se utilizan las directivas `<TD>` y `<TH>`. `<TD>` indica una celda normal, y `<TH>` indica una celda de "cabecera", es decir, el contenido será resaltado en negrita y en un tamaño ligeramente superior al normal. Los parámetros opcionales de ambas directivas son:

align = LEFT / CENTER / RIGHT / Indica cómo se debe alinear el contenido de la celda, a la izquierda (LEFT), a la derecha (RIGHT), centrado (CENTER) o justificado (JUSTIFY).

JUSTIFY

<i>valign = TOP / MIDDLE / BOTTOM</i>	Indica la alineación vertical del contenido de la celda, en la parte superior (TOP), en la inferior (BOTTOM), o en el centro (MIDDLE).
<i>rowspan = número</i>	Indica el número de filas que ocupará la celda. Por defecto ocupa una sola fila.
<i>colspan = número</i>	Indica el número de columnas que ocupará la celda. Por defecto ocupa una sola columna.
<i>width = número ó %</i>	Indica la anchura de la columna en puntos o en porcentaje en función del ancho de la ventana del visor. Si no se indica este parámetro, el ancho se adecuará al tamaño de los contenidos. Este parámetro sólo funciona en los navegadores modernos.
<i>bgcolor = código de color</i>	Especifica el color de fondo del elemento de la Tabla.

Para indicar que acaba una fila de celdas se utiliza la directiva `<TR>`. A continuación se muestra un ejemplo de una tabla que contiene sólo texto. Como se indicó anteriormente el contenido de las celdas puede ser cualquier elemento de HTML, un texto, una imagen, un hipere enlace, una lista, etc.

Las directivas `<TD>` y `<TH>` son cerradas según el estándar de HTML, es decir, que un elemento de tabla `<TD>` debería cerrarse con un `</TD>`, sin embargo los visores asumen que un elemento de la tabla, queda automáticamente "cerrado" cuando se "abre" el siguiente.

1.2.3.6 Mapas

Un Mapa es una imagen que permite realizar diferentes hipere enlaces en función de la "zona" de la imagen que se pulse. Las directivas para crear mapas son `<MAP></MAP>` y `<AREA>`.

<MAP> Identifica al mapa y tiene el parámetro *name* para indicar el nombre del mapa.

<AREA> Define las áreas sensibles de la imagen y tiene los siguientes parámetros obligatorios:

<i>shape = "tipo"</i>	Indica el tipo de área a definir.
<i>coords = "coordenadas"</i>	Indica las coordenadas de la figura indicada con <i>shape</i> .
<i>href = "URL"</i>	Indica la dirección a la que se accede si se pulsa en la zona delimitada por el área indicada.

Los tipos de área pueden ser los siguientes:

<i>rect</i>	Área rectangular. Se deben especificar las coordenadas de la esquina superior izquierda y las de la esquina inferior derecha.
<i>poly</i>	Polígono. Se deben especificar las coordenadas de todos los vértices del polígono. El visor se encarga de "cerrar" la figura.
<i>circle</i>	Círculo. Se debe especificar en primer lugar las coordenadas del centro del círculo y a continuación el valor del radio (en puntos).

Si dos áreas se superponen, se ejecutará la que se encuentre en primer lugar en la definición del mapa. Es importante definir una última área que abarque la totalidad del gráfico para direccionar a un URL "por defecto", con el objeto de contemplar en caso de no pulsar sobre un área definida.

Para activar el mapa se debe indicar la imagen a mostrar, indicando que dicha imagen es tratada como un mapa. Para ello se escribe la siguiente directiva:

```
<IMG src = "grafico.gif" usemap = "#casa">
```

El proceso más laborioso es el determinar las coordenadas. Existen programas que permiten realizar este proceso de una manera sencilla, simplemente definiendo las zonas con el ratón.

1.2.3.7 Formularios

Los formularios permiten dentro de una página Web solicitar información al visitante y procesarla. En un formulario se puede solicitar diferentes datos (campos) cada uno de los cuales quedará asociado a una variable. Una vez que se hayan introducido los valores en los campos, el contenido de estos será enviado a la dirección (URL) donde tenga el programa que pueda procesar las variables. Para poder realizar este último paso de procesar las variables se necesita realizar un programa externo en algún lenguaje de programación como PERL, C++ o Visual Basic. A este programa externo se le suele llamar *CGI* (Common Gateway Interface).

La declaración del formulario se pone entre las directivas *<FORM></FORM>*. En el interior de la declaración se indican los elementos (variables) de entrada. La directiva *<FORM>* tiene los parámetros *action* y *method*.

action = Indica el programa que va a "tratar" a las variables que se envíen
"programa" con el formulario. Para nuestro caso se envían las variables por
 correo electrónico, con lo que el "programa" será "mailto:
 direccion_de_correo".

method = Indica el método según el que se transferirán las variables. POST
POST/GET produce la modificación del documento de destino (como en el
 caso de enviar por correo las variables). GET no produce cambios
 en el documento destino (como en el caso de una consulta a una
 base de datos. Por ejemplo: una página de búsqueda en Internet).

a) Campos de entrada

Para la introducción de las variables se utiliza la directiva **<INPUT>**. Esta directiva tiene los parámetros:

type = Indica el tipo de variable a introducir. Cada tipo de **text/password/etc.** variable tiene sus propios parámetros.

name = nombre Indica el nombre que se le dará al campo

type = text name = campo Indica que el campo a introducir será un texto. Sus parámetros son:

maxlength = Número máximo de caracteres a introducir en el campo
número

size = número Tamaño en caracteres que se mostrará en pantalla.

value = "texto" Valor inicial del campo. Normalmente será " ", o sea, vacío.

type = password name = campo Indica que el campo será una palabra de paso. Mostrará asteriscos (*) en lugar de las letras escritas. Sus parámetros opcionales son los mismos de text.

type = checkbox name = campo El campo se elegirá marcando una casilla. Se permite marcar varias casillas. Los valores de las casillas serán indicados por:

value = "valor" Valor de la variable a enviar.

checked La casilla aparecerá marcada por defecto.

type = radio name = campo El campo se elegirá marcando una casilla. Sólo permite marcar una sola de las casillas. Los valores de las casillas serán indicados por:

value = "valor" Valor de la variable a enviar.

type = image name = campo El campo contendrá el valor de las coordenadas del punto de la imagen pinchado. Debe indicarse la imagen con el parámetro:

src = "fichero de imagen" Indica el lugar del fichero de imagen.

type = hidden name = campo El usuario no puede modificar su valor, ya que el campo no es visible se manda siempre con el valor indicado por el parámetro:

value = "valor" Valor de la variable a enviar.

type = submit Representa un botón. Al pulsar este botón la información de todos los campos se envía al programa indicado en *<FORM>*. Tiene el parámetro:

value = "texto" Indica el texto que aparecerá en el botón.

type = reset Representa un botón. Al pulsar este botón se borra el contenido de todos los campos. Tiene el parámetro:

value = "texto" Indica el texto que aparecerá en el botón.

b) Campos de selección

Este tipo de campo despliega una lista de opciones, entre las que se debe escoger una o varias. Se utiliza para ellos la directiva *<SELECT> </SELECT>*. Sus parámetros son:

name = campo Nombre del campo

size = número Número de opciones visibles. Si se indica 1 se presenta como un menú desplegable, se indica más de uno se presenta como una lista con barra de desplazamiento.

multiple Permite seleccionar más de un valor para el campo.

Las diferentes opciones de la lista se indican con la directiva **<OPTION>**. Esta directiva puede incluir el parámetro *selected* para indicar cuál es la opción por defecto. En caso de que no se especifique, se tomará por defecto la primera opción de la lista.

c) Áreas de texto.

Representa un campo de texto de múltiples líneas. Normalmente se utiliza para que se incluyan en los comentarios. La directiva usada es **<TEXTAREA>** **</TEXTAREA>**, y sus parámetros son:

<i>name = campo</i>	Nombre del campo.
<i>cols = número</i>	Número de columnas de texto visibles.
<i>rows = número</i>	Número de filas de texto visibles.
<i>wrap = VIRTUAL</i>	Justifica el texto automáticamente en el interior de la caja. La opción PHYSICAL envía las líneas de texto separadas en líneas físicas. La opción VIRTUAL envía todo el texto seguido.
<i>PHYSICAL</i>	

Se observa que en el correo se separan las variables con el símbolo **&**, los espacios se sustituyen por el signo **+** y se representan los códigos de retorno de carro y avance de línea del campo de texto con los caracteres **%0D** y **%0A** respectivamente.

Si en vez de enviar estas variables por correo electrónico, fuesen enviadas a un programa (CGI), este programa podría tratarlas y dar como respuesta una nueva página Web.

1.2.4 ELEMENTOS MULTIMEDIA

1.2.4.1 Imágenes

Para incluir una imagen en una página Web se utiliza la directiva ****. Hay dos formatos de imágenes que todos los navegadores modernos reconocen. Son las imágenes *GIF* y *JPG*. Cualquier otro tipo de fichero gráfico o de imagen (BMP, PCX, CDR, etc.) no será mostrado por el visor, a no ser que disponga de un programa externo que permita su visualización.

La directiva **** tiene los siguientes parámetros:

- src = "imagen"*** Indica el nombre del fichero gráfico a mostrar.
- alt = "Texto"*** Mostrara el texto indicado en el caso de que el navegador utilizado para ver la página no sea capaz de visualizar la imagen.
- lowsrc = "imagen"*** Muestra una segunda imagen "superpuesta" sobre la primera una vez que se carga la página.
- align = TOP/ MIDDLE/ BOTTOM*** Indica cómo se alineará el texto que siga a la imagen. TOP alinea el texto con la parte superior de la imagen, MIDDLE con la parte central, y BOTTOM con la parte inferior.
- border = tamaño*** Indica el tamaño del "borde" de la imagen. A toda imagen se le asigna un borde que será visible cuando la imagen forme parte de un Hiperenlace.
- height = tamaño*** Indica el alto de la imagen en puntos o en porcentaje. Se usa para variar el tamaño de la imagen original.
- width = tamaño*** Indica el ancho de la imagen en puntos o en porcentaje. Se usa para variar el tamaño de la imagen original.
- hspace = margen*** Indica el número de espacios horizontales, en puntos, que separará la imagen del texto que lo sigue y lo antecede.

vspace = margen Indica el número de puntos verticales que separará la imagen del texto que le sigue y lo antecede.

ismap / usemap Indica que la imagen es un MAPA

1.2.4.2 Marquesina

La directiva `<MARQUEE></MARQUEE>` crea una marquesina con un texto en su interior que se desplaza. Sus parámetros son los siguientes:

align = top/middle/bottom Indica si el texto del interior de la marquesina se alinea en la zona alta (top), en la baja (bottom) o en el centro (middle) de la misma.

bgcolor = "código de color" Indica el color del fondo de la marquesina.

behavior = "alternate" / "slide" Alterna el movimiento de la marquesina a la izquierda y derecha (alternate) o a la derecha o izquierda y al llegar a cualquiera de los dos lados desaparece e inicia nuevamente (slide).

direction = "left" / "right" Indica hacia que lugar se desplaza el texto, hacia la izquierda (left) o hacia la derecha (right).

height = número o % Indica la altura de la marquesina en puntos o porcentaje en función de la ventana del visor.

width = número o % Indica la anchura de la marquesina en puntos o porcentaje en función de la ventana del visor.

loop = número/infinite Indica el número de veces que se desplazará el texto por la marquesina. Si se indica infinite, se desplazará indefinidamente.

scrolldelay = número Indica el número de milisegundos que tarda en reescribirse el texto por la marquesina, a mayor número más lentamente se desplazará el texto.

1.2.4.3 Gifs Animados.

Un Gif animado es un gráfico que da la impresión de animación. Para conseguir este efecto se deben crear por separado los gráficos GIF que componen la secuencia de la animación. Una vez realizado esto se utiliza un programa para "fundir" todos estos gráficos en uno sólo. Al mostrar este gráfico en nuestra página Web, el visor mostrará la secuencia de todos los gráficos individuales de forma continua, con lo que dará el aspecto de animación.

1.2.4.4 Gifs Transparentes

Un Gif transparente es un gráfico en el cual se define su color de fondo como transparente. Con esto se consigue el efecto de que el gráfico se encuentre como "flotando" en nuestra página.

Para visualizar un gif animado se utiliza la directiva ``.

1.2.4.5 Sonido

Nuestra página Web puede tener un sonido que se active al entrar en la página. Esta característica de Ms Explorer utiliza la directiva `<BGSOUND>` y tiene los siguientes parámetros:

src = "fichero" Indica el nombre del fichero que contiene el sonido (.wav, .mid).

loop = num / Indica el número de veces que se reproducirá el sonido. Si se

infinite indica infinite, el sonido se reproducirá de forma continua hasta que abandone la página.

Para usar esta función en Netscape se utiliza la directiva **<EMBED>**. Esta directiva se utiliza realmente para "incrustar" un objeto en nuestra página Web. Dicho objeto puede ser un fichero de sonido, un vídeo, un gráfico BMP, etc. Para que pueda mostrarse el fichero deberá tener el plug-in adecuado que lo trata. *plug-in* es un programa que extiende las capacidades del navegador en un modo específico, dado por ejemplo la capacidad de mostrar vídeo, audio, ficheros de un determinado formato (ficheros PDF, presentaciones de ASAP, fichero VRML, etc.). Tiene los siguientes parámetros:

src = "fichero" Indica el nombre del fichero que contiene el sonido (.wav, .mid) o el vídeo (.avi).

autostart = true Indica que la reproducción se inicie inmediatamente.

loop = true Para que la reproducción no se detenga. (al terminar, vuelve a comenzar automáticamente).

volume = número Volumen al que se reproducen los ficheros de sonido.

width = número Anchura y Altura de la representación del objeto. (Si es un
height = número sonido no es necesario este parámetro).

controls = Visualiza una serie de controles que permiten iniciar la
smallconsole reproducción del fichero, así como realizar una pausa.

1.2.4.6 Vídeo

```
<IMG DYN SRC="fichero de vídeo" LOOP=n ó INFINITE CONTROLS  
START=FILEOPEN ó MOUSEOVER>
```

En este caso es un nuevo atributo para la etiqueta ****, en vez de mostrar una imagen mostrará un vídeo en formato AVI (Vídeo para Windows), se puede utilizar todas los atributos normales de ****, pudiendo usar las distintas alineaciones y

colocaciones respecto al texto, además se podrá variar el tamaño del vídeo con los atributos <HEIGHT> y <WIDTH>.

- <*imgsrc*> Indica el URL del fichero que se mostrará.
- <*loop*> Indica el número de veces que se mostrará el vídeo.
- <*controls*> Indicará si se mostrarán los botones de control, que permitirán, volver a reproducir.
- <*start*> Indica si la secuencia de vídeo empezará al abrir el fichero (FILEOPEN) o cuando pase el cursor del ratón encima (MOUSEOVER).
- <*src*> De no poder mostrar el vídeo por tratarse de otro navegador (por ejemplo Netscape), se podrá indicar una imagen que se muestre de manera alternativa, incluyendo este atributo

1.2.5 Hiperenlaces

La característica principal de una página Web es que se puede incluir Hiperenlaces. Un Hiperenlace es un elemento de la página que hace que el navegador acceda a otro recurso, otra página Web, un archivo, etc.

Para incluir un Hiperenlace se utiliza la directiva <A>. El texto o imagen que se encuentre dentro de los límites de esta directiva será sensible, esto quiere decir que si pulsa con el ratón sobre él, realizará el Hiperenlace indicado por la directiva <A>. Si el Hiperenlace esta indicado por un texto, este aparecerá subrayado y en distinto color, si se trata de una imagen, esta aparecerá con un borde rodeándola. Esta directiva tiene el parámetro:

- href* Indica el lugar a donde llevará el Hiperenlace si lo pulsa.

Si sitúa el ratón encima de la frase y pulsa, el navegador accederá a la página Web indicada por el parámetro *href*, es decir, accederá a la página situada en *http://www.microsoft.com/*. Lo mismo se puede hacer con un gráfico.

Un Hiperenlace también puede ubicar en una zona de la página. Para ello se debe marcar en la página las diferentes secciones en las que se divide. Se lo realiza con el parámetro *name*.

```
<A name = "seccion1"></A>
```

Esta instrucción marca el inicio de una sección dentro de la página. La sección se llamará *seccion1*. Para hacer un enlace a esta sección dentro de la página se lo hará de la siguiente forma:

```
<A href = "#seccion1">Primera Parte</A>
```

O también:

```
A href = "http://www.jet.es/mipagina.htm#seccion1">Primera Parte</A>
```

Un Hiperenlace puede hacerse a cualquier tipo de fichero. Con las directivas anteriores se a visto como hacer enlaces a páginas Web o secciones dentro de una página web, pero se puede hacer un Hiperenlace a un grupo de noticias, o a otro servicio de Internet.

Así mismo puede hacer que el Hiperenlace de como resultado el envío de un correo electrónico a una dirección de correo determinada.

Ejemplo:

```
<A href =  
"mailto:bertha_bautista@comil.com">Escribeme</A>
```


También se puede realizar un Hiperenlace a un fichero cualquiera. En este caso el navegador intentará "ejecutar" el fichero, y si no puede hacerlo preguntará si desea grabarlo en su ordenador. Esta es una forma sencilla de permitir a los visitantes de la página copiar ficheros a su ordenador.

Ejemplo:

```
<A href = "turismo.zip">Pulsa aquí para llevarte una  
copia.</A>
```

El parámetro *onMouseOver* permite que se muestre en la barra inferior del navegador un texto explicativo sobre el Hiperenlace, en vez de mostrar su dirección. Este parámetro no funciona en todos los navegadores.

Ejemplo:

```
<A href = "turismo.zip" onMouseOver="window.status='Este link copia el  
manual a tu disco duro' ; return true;">Pulsa aquí para llevarte una copia.</A>
```

CAPÍTULO II

BASE DE DATOS

2.1 ¿Qué es una base de datos?

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. A continuación se presenta una guía que explicará el concepto y características de las bases de datos.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más **columnas** y **filas**. Las columnas guardan una parte de la información sobre cada elemento que se quiera guardar en la tabla, cada fila de la tabla conforma un registro.

2.2 Características de las bases de datos

Entre las principales características de los sistemas de base de datos se puede mencionar:

- Independencia lógica y física de los datos.

- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

2.3 Sistema de Gestión de Base de Datos (SGBD)

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

2.4 Ventajas de las bases de datos

- Control sobre la redundancia de datos
- Consistencia de datos
- Compartición de datos
- Mantenimiento de estándares
- Mejora en la integridad de datos
- Mejora en la seguridad (usuarios no autorizados)
- Mejora en la accesibilidad a los datos
- Mejora en la productividad
- Mejora en el mantenimiento
- Aumento de la concurrencia
- Mejora en los servicios de copias de seguridad

2.5 Desventajas de las bases de datos

- Complejidad:
- Coste del equipamiento adicional:
- Vulnerable a los fallos:

2.6 Tipos de Campos

Cada Sistema de Base de Datos posee tipos de campos que pueden ser similares o diferentes. Entre los más comunes podemos nombrar:

- **Numérico:** entre los diferentes tipos de campos numéricos podemos encontrar enteros “sin decimales” y reales “decimales”.
- **Booleanos:** poseen dos estados: Verdadero “Si” y Falso “No”.
- **Memos:** son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados.
- **Fechas:** almacenan fechas facilitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.
- **Alfanuméricos:** contienen cifras y letras. Presentan una longitud limitada (255 caracteres).
- **Autoincrementables:** son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta: Servir de identificador ya que resultan exclusivos de un registro.

2.7 Estructura de una Base de Datos

Una base de datos, a fin de ordenar la información de manera lógica, posee un orden que debe ser cumplido para acceder a la información de manera coherente. Cada base de datos contiene una o más tablas, que cumplen la función de contener los campos.

En el siguiente ejemplo mostramos una tabla “comentarios” que contiene 4 campos.

	Field	Type
<input type="checkbox"/>	<u>id</u>	int(11)
<input type="checkbox"/>	<u>titulo</u>	varchar(100)
<input type="checkbox"/>	<u>texto</u>	blob
<input type="checkbox"/>	<u>fecha</u>	varchar(10)

Tabla II.1: Tabla comentarios con 4 campos

Los datos quedarían organizados como mostramos en siguiente ejemplo:

	<u>id</u>	<u>titulo</u>	<u>texto</u>	<u>fecha</u>
<input type="checkbox"/>	1	saludos	[BLOB - 0 B]	22-10-2007
<input type="checkbox"/>	2	como estas ???	[BLOB - 0 B]	23-10-2007

Tabla II.2: contenido de tabla comentarios

Por consiguiente una base de datos posee el siguiente orden jerárquico:

- Tablas
- Campos
- Registros
- Lenguaje SQL

2.8 Lenguaje de base de datos SQL

Introducción

El lenguaje de consulta estructurado (**SQL**) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet. **SQL** se utiliza para crear objetos QueryDef, como el argumento de origen del método OpenRecordSet y como la propiedad RecordSource del control de datos. También se puede utilizar con el método Execute para crear y manipular directamente las bases de datos Jet y crear consultas **SQL** de paso a través para manipular bases de datos remotas cliente - servidor.

2.8.1 Componentes del SQL

El lenguaje **SQL** está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

2.8.2 Comandos

Existen dos tipos de comandos **SQL**:

- Los **DLL** que permiten crear y definir nuevas bases de datos, campos e índices.
- Los **DML** que permiten generar consultas para ordenar, filtrar y extraer datos.

Comandos **DLL**

Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Empleado para eliminar tablas e índices
ALTER	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Tabla II.3: Comandos Dll sql

Comandos **DML**

Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

Tabla II.4: Comandos DML sql

2.8.3 Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Comando	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

Tabla II.5: Cláusulas SQL

2.8.4 Operadores Lógicos

Operador	Uso
AND	Es el “y” lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el “o” lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

Tabla II.6: Operadores Lógicos SQL

2.8.5 Operadores de Comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor ó Igual que
>=	Mayor ó Igual que
BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo
In	Utilizado para especificar registros de una base de datos

Tabla II.7: Operadores de comparación SQL

2.8.6 Funciones de Agregado

Las funciones de agregado se usan dentro de una cláusula **SELECT** en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

Comando	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

Tabla II.8: funciones agregado

2.9 El interfaz World Wide Web a una base de datos.

Así pues, el interfaz es un elemento fundamental a la hora de hacer uso de una base de datos, aquí es donde entra la World Wide Web (WWW).

La World Wide Web se puede considerar como el elemento de Internet que integra el cliente universal, y por tanto, puede usarse como el interfaz universal a todo tipo de aplicaciones.

Muchos fabricantes de equipos y desarrolladores han empezado a mover sus aplicaciones e interfaces para que puedan ser operados con clientes basados en ésta. Así, por ejemplo, nos encontramos con fabricantes de elementos de interconexión de redes (*switches*, conmutadores, o *routers*) o impresoras de red que incluyen en su hardware un servidor de WWW para que sus herramientas puedan ser manejadas, configuradas, en fin, gestionadas, desde un navegador de WWW.

La ventaja para el fabricante es clara ya que le evita tener que crear una herramienta para el cliente, ahorrando en el desarrollo; no olvidemos que para un fabricante de hardware un cliente puede ser, potencialmente, cualquier combinación posible de uso de hardware, sistema operativo y software. Así pues en lugar de tener que realizar un interfaz para, por ejemplo, Solaris, otro para Linux y otro para Windows NT, el fabricante indica al usuario que sólo necesita un navegador de WWW como interfaz a la gestión de su equipo. No será necesario que programe un interfaz para cada plataforma en la que quiera que su hardware sea gestionado sino que, al incorporar tecnologías estándar de la WWW tanto de presentación (HTML) como de interacción cliente servidor (protocolo HTTP) su herramienta puede ser utilizada desde cualquier lugar y con cualquier plataforma que tenga un cliente de WWW.

El movimiento del acceso de bases de datos de sistemas propietarios a un sistema abierto como es la WWW, empezó ya hace años. Un buscador no es más que una base de datos de documentos al que se accede mediante un interfaz WWW. La ventaja, al margen de la que ofrezca la aplicación en sí, es que, por un lado, cualquiera puede "preguntar" a la base de datos sin conocer el funcionamiento de lo que hay detrás, por otro los "clientes" potenciales son todos aquellos con acceso a la WWW porque los estándares usados son los mismos que para acceder a cualquier otro servidor.

CAPITULO III

HERRAMIENTAS DE DESARROLLO WEB

3.1 PHP

PHP es una tecnología que nos permite la generación de un contenido dinámico en un servidor Web. Esta tecnología fue creada originalmente en 1994 por Rasmus Lerdorf como un conjunto de herramientas (conocidas como PHP/FI y posteriormente PHP 2.0) que permitirían tener un dinamismo en el Web.

Dada la naturaleza de ser open-source (código abierto) estas herramientas ganaron gran popularidad y en 1997 Zeev Zuraski y Andi Gutmans rediseñaron el parser, bautizándolo con el nombre de PHP3. Poco después se volvió a rediseñar y dio lugar a lo que actualmente conocemos como PHP4. Dentro de PHP existen muchos conceptos de Perl, C y Java por lo que gracias a esta sintaxis es más fácil familiarizarse con él, incluso para programadores principiantes.

El código PHP se encuentra incrustado en documentos HTML por medio de tags como veremos a continuación en este ejemplo:

```
<HTML>
<HEAD>
  <TITLE> Prueba PHP </TITLE>
</HEAD>
<BODY>
  <P align ="center"> Script PHP </P>
```

```
<?php
    echo "Hola Mundo";
?>
</BODY>
</HTML>
```

En este ejemplo podemos ver que para incluir código PHP en HTML activamos el tag “<?php”, posteriormente incluimos las instrucciones correspondientes y para finalizar cerramos nuestros tag con “?>”. Dentro de un mismo documento se pueden activar y desactivar los tags de PHP tantas veces sea necesario, para hacer esto existen varias opciones como:

- Utilizar los tags “<?php” y “?>”.
- Utilizar el modo abreviado de los tags “<? y ?>” (siempre y cuando este habilitada esta opción).
- Con el tag <SCRIPT> de HTML : <SCRIPT language =”php”> y </SCRIPT>.
- Con los tags “<%” y “%>” también utilizados en JSP y ASP, requiere configuración especial.
- Para los comentarios podemos usar “//” en cada línea o “/* y */” si queremos encerrar un conjunto de líneas.

3.1.1 Características principales de PHP

- Es una tecnología que genera contenidos dinámicos y se encuentra embebido en archivos HTML.
- Dispone de un gran número de librerías de conexión con la mayoría de los manejadores de bases de datos para el manejo de información permanente en un servidor.
- Nos proporciona soporte a múltiples protocolos de comunicación en Internet (HTTP, IMAP, FTP, LDAP, SNMP, etc.).
- El código del intérprete esta accesible para permitir posibles mejoras o sugerencias acerca de su desarrollo (Está escrito en lenguaje C).
- No es necesario adoptar alguna tecnología especial en lo que a Web servers se refiere ya que puede correr sobre cualquier plataforma.

- Consume muy pocos recursos del servidor, no es necesario tener un equipo muy poderoso.

3.1.2 Funcionamiento del intérprete de PHP

Las páginas que contienen únicamente HTML se pueden desarrollar y probar sin la intervención de un servidor Web, esto se debe a que todos los navegadores son capaces de interpretar el código HTML e incluso código como JavaScript además de que se ejecutan localmente. A diferencia de estos 2, PHP es ejecutado del lado del servidor antes de ser enviado de vuelta al usuario que realizó una petición.

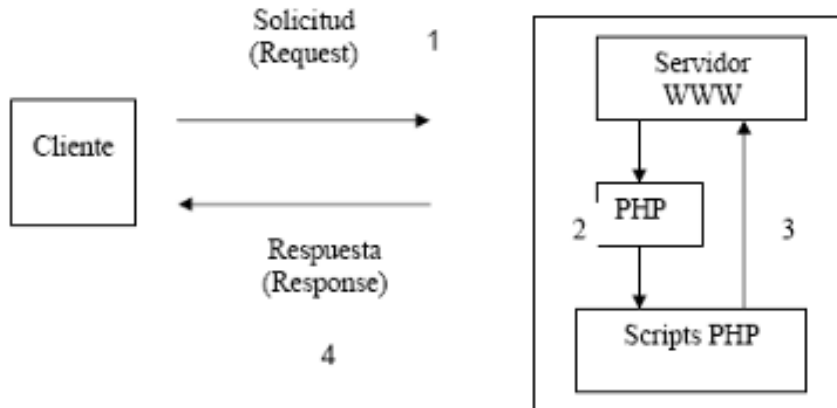


Figura III.1: Solicitud de un archivo PHP

Los pasos que se ejecutan de acuerdo al diagrama son los siguientes:

1. El usuario pulsa algún enlace solicitando algún archivo PHP, el navegador envía esta solicitud al servidor empleando el protocolo HTTP.
2. La solicitud es recibida y se localiza un documento, por la extensión se determina que se trata de un documento PHP y se lanza al intérprete.
3. El intérprete ejecuta el script solicitado y genera un resultado en una página HTML que se devuelve al servidor para posteriormente regresarse al cliente.
4. Se visualiza el documento por el usuario en el navegador.

3.1.3 Tipos de datos y variables

En PHP no es necesario declarar las variables ni el tipo de información que van a contener, las variables quedan automáticamente declaradas al utilizarlas por primera vez y su tipo queda determinado por la clase de información. Para declarar una variable, lo hacemos con el signo “\$” antes del nombre de la variable y los tipos posibles de variables son:

Tipo	Descripción
Integer	Números Enteros
Double	Números en coma flotante
String	Cadenas de caracteres
Array	Tipo especial de colección de variables
Object	Tipo especial de dato complejo
Boolean	Valor lógico: TRUE o FALSE

Tabla III.1: Tipos de datos en PHP

3.1.3.1 Datos complejos

Arrays

Este tipo de dato es una colección de valores que comparten el mismo nombre y pueden ser manipulados de manera global o individual como podemos ver a continuación:

```
$MiArreglo[5]=[a,b,c,d,e,f]; //Aquí se declara el arreglo y el valor que tendrá en cada posición
```

```
$MiArreglo[2]; //Nos regresara el valor c;
```

3.1.3.2 Variables generales en PHP

Estas variables se encuentran disponibles para el programador y se pueden acceder en cualquier momento, algunas variables son:

Variable	Contenido
Argv	Array de argumentos pasados en la ejecución de scripts .
Argc	Números de parámetros pasados al script en su ejecución.
PHP_SELF	Nombre del script que se está utilizando. Se usa comúnmente para la captura de formas enviadas por el usuario.
HTTP_COOKIE_VARS	Array asociativo (clave, valor) a través de cookies.
HTTP_GET_VARS	Array asociativo (clave, valor) pasados usando el método GET.
HTTP_POST_VARS	Array asociativo (clave, valor) pasados usando el método POST.
HTTP_POST_FILES	Array asociativo que contiene información de los archivos recibidos usando el método POST.
HTTP_ENV_VARS	Array asociativo (clave, valor) del entorno.
HTTP_SERVER_VARS	Array asociativo (clave, valor) del servidor.
HTTP_SESSION_VARS	Array asociativo (clave, valor) de sesión.

Tabla III.2: Variables del servidor

Al ser variables deben ir precedidas del signo “\$” por ejemplo en este archivo llamado “script.php”:

```
<HTML>
<BODY>
    <?php
        echo “Nombre del Script: $PHP_SELF”; ?>
</BODY>
</HTML>
```

Generara en el navegador algo así: Nombre del Script: script.php

3.1.4 Sentencias De Control

Las sentencias de control permiten ejecutar bloque de códigos dependiendo de unas condiciones, para ejecutar de manera repetida un bloque de sentencias y en general para

cambiar la secuencia normal de un programa. Para PHP el 0 es equivalente a Falso y cualquier otro número es Verdadero.

3.1.4.1 Sentencia if

La sentencia if permite llevar a cabo la ejecución condicional de sentencias.

```
if ( Expresion ){  
    sentencias;  
}
```

Se ejecutan las sentencias si al evaluar la expresión se obtiene un valor booleano TRUE.

```
if ( Expresion ){  
    sentenciasA;  
}  
else{  
    sentenciasB;}  
}
```

Si al evaluar la expresión se obtiene un valor booleano TRUE se ejecutarán las Sentencias A, en caso contrario se ejecutarán las sentenciasB.

Las sentencias if se pueden anidar indefinidamente dentro de otras sentencias if, lo cual proporciona una flexibilidad completa para ejecuciones condicionales en las diferentes partes de un programa.

3.1.4.2 Sentencia switch

Cuando se requiere comparar una variable con una serie de valores diferentes, puede utilizarse la sentencia switch, en la que se indican los posibles valores que puede tomar la variable y las sentencias que se tienen que ejecutar sí es que la variable coincide con alguno de dichos valores.

```
switch( variable ){  
    case valor1:  
        sentencias;  
        break;  
    case valor2:  
        sentencias;  
        break;  
}
```

```
...
case valorN:
sentencias;
break;
default:
sentencias;}
```

Cada case ejecutará las sentencias correspondientes, con base en el valor de la variable, que puede evaluarse con valores de tipo int, float o string. Si el valor de la variable no coincide con ningún valor, entonces se ejecutan las sentencias por default, sí es que las hay.

La sentencia break al final de cada case transfiere el control al final de la sentencia switch; de esta manera, cada vez que se ejecuta un case todos los enunciados case restantes son ignorados y termina la operación del switch.

3.1.4.3 Ciclo for

El ciclo for repite una sentencia, o un bloque de sentencias, mientras una condición se cumpla. Se utiliza la mayoría de las veces cuando se desea repetir una sentencia un determinado número de veces. La forma general de la sentencia for es la siguiente;

```
for(inicialización;condición;incremento) {
sentencias;
}
```

En su forma más simple, la inicialización es una sentencia de asignación que se utiliza para establecer una variable que controle el ciclo.

La condición es una expresión que comprueba la variable que controla el ciclo y determina cuando salir del ciclo.

El incremento define la manera en como cambia la variable que controla el ciclo.

3.1.4.4 Ciclos while y do-while.

Al igual que los ciclos for repiten la ejecución de un bloque de sentencias mientras se cumpla una condición específica. El formato de la sentencia while es la siguiente:


```
while (condición){  
  sentencias;  
}
```

Esta es una condición booleana, que mientras tenga el valor TRUE permite que se ejecuten las sentencias correspondientes.

La sentencia do-while

Al contrario de los ciclos for y while que comprueban una condición en lo alto del ciclo, el ciclo do-while la examina en la parte más baja del mismo. Esta característica provoca que un ciclo do-while siempre se ejecute por lo menos una vez. El formato de la sentencia do-while es el siguiente:

```
do{  
  sentencias;  
}while (condición);
```

3.1.4.5 break

La sentencia break tiene dos usos. El primer uso es terminar un case en la sentencia switch. El segundo es forzar la terminación inmediata de un ciclo, saltando la prueba condicional normal del ciclo.

3.1.4.6 continue

La sentencia continue es similar a la sentencia break. Sin embargo, en vez de forzar la terminación del ciclo, continue fuerza la siguiente iteración y salta cualquier código intermedio.

3.1.4.7 return

Se utiliza la sentencia return para provocar la salida del método actual; es decir, return provocará que el programa vuelva al código que llamó al método.

La sentencia return puede regresar o no un valor. Para devolver un valor, se pone el valor después de la palabra clave return.

3.1.5 Funciones

Una función es un bloque de código que puede ser definido una vez y ser ejecutado desde diferentes partes de un programa. Típicamente una función toma uno o más argumentos, ejecuta un conjunto de operaciones sobre ellos y entonces regresa un resultado. Una función se define con la siguiente sintaxis:

```
function nombreFuncion( /* parametros */)
{
... cuerpo de la función
...}
```

3.2. MySQL

El manejador de bases de datos que veremos y en el que está centrado nuestra aplicación es MySQL Server.

3.2.1 ¿Qué es MYSQL?

- **Administrador de Base de Datos**

Una base de datos es un conjunto de datos estructurados. Esto podría ser cualquier cosa, desde una simple lista de compras hasta una galería de pinturas o la gran cantidad de información que se maneja en una corporación.

Para agregar, acceder y procesar los datos almacenados en una base de datos computacional, se necesita un sistema administrador de base de datos tal como MySQL. Además los computadores son muy buenos manejando grandes cantidades de datos, el administrador de base de datos juega un rol central en la computación, ya sea como utilidad autónoma o parte de otra aplicación.

- **Sistema administrador de base de datos relacionales**

Una base de datos relacional almacena datos en tablas separadas, más bien colocando todos los datos en un gran almacén. Esto agrega velocidad y flexibilidad. Las tablas son enlazadas por relaciones definidas haciendo posible combinar datos desde varias tablas solicitadas. El SQL forma parte de MySQL,

conocido como Lenguaje de Consultas Estructurado, es el lenguaje estandarizado más común usado para acceder base de datos.

- **Software de Fuente Abierta**

Fuente abierta significa que es posible que pueda ser usado y modificado por cualquiera. Alguien puede bajar MySQL desde Internet y usar éste sin tener que pagar nada. Alguien puede estudiar el código fuente y cambiarlo de acuerdo a sus necesidades.

3.2.2 Uso de MySQL

MySQL es muy rápido, seguro y fácil de usar. MySQL también ha desarrollado un conjunto de características muy prácticas, en estrecha cooperación con otros usuarios. MySQL fue desarrollado para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha sido usado exitosamente en ambientes de producción con altas demandas, por varios años. Aunque está bajo un desarrollo constante, MySQL siempre ofrece conjunto de funciones muy poderoso y eficiente. La conectividad, velocidad y seguridad hace de MySQL una suite poderosa para acceder a bases de datos en Internet.

3.2.3 Características Técnicas de MySQL

MySQL es un sistema Cliente/Servidor que consta de un servidor SQL multihilo que soporta diferentes backends, variados programas cliente y de librerías, administrador de herramientas y un programa de interface. MySQL contribución para muchos de los software disponibles.

Es mucho más probable que se encuentre que diversas aplicaciones ya soportan MySQL. Los valores centrales de MySQL son:

- La mejor y más usada base de datos en el mundo.
- Disponible y Accesible para todos
- Fácil de usar
- Se está perfeccionando continuamente mientras permanece rápida y segura.
- Divertida para usar y perfeccionar.

- Libre de molestias.

3.2.4 Características Principales De MySQL

A continuación se describen algunas de las características más importantes de MySQL:

- Escrito en C y C++, testado con GCC 2.7.2.1. Usa GNU autoconf para portabilidad.
- Clientes C, C++, Eiffel, PHP, Python, JAVA, Perl, TCL.
- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y S.O. distintos.
- Sistema de contraseñas y privilegios muy flexibles y seguros.
- Todas las palabras de paso viajan encriptadas en la red.
- Registros de longitud fija y variable.
- 16 índices por tabla, cada índice puede estar compuesto de 1 a 15 columnas o partes de ellas con una longitud máxima de 127 bytes.
- Todas las columnas pueden tener valores por defecto.
- Utilidad *Isamchk* para chequear, optimizar y reparar tablas.
- Todos los datos están grabados en formato ISO8859_1.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- Todos los comandos tienen -help o -? Para las ayudas.
- Soporta diversos tipos de columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc.
- ODBC para Windows 95 (con fuentes), se puede utilizar ACCESS para conectar con el servidor.
- Muy rápida usando joins, optimizada para un barrido multi-joins.
- Todas las funciones y operadores soportan en el **SELECT** y **WHERE** como partes de consultas.
- Todas las cláusulas SQL soportan **GROUP BY** y **ORDER BY**.

3.2.5 PHP y las bases de datos

PHP es capaz de conectarse con muchos manejadores de bases de datos diferentes tales como PostgreSQL, Oracle, Informix, etc. Sin embargo uno de los más utilizados con esta

tecnología y a la facilidad de uso es MySQL. Todas las funciones comienzan con el prefijo “mysql_”, y si se cambia de manejador de bases de datos solo cambiara el prefijo que corresponderá al nombre de cada uno de ellos, por ejemplo si quisiéramos utilizar ODBC, el prefijo de las funciones de manejo de datos seria “odbc_” o en el caso de trabajar con Oracle cambiaria a “oracle_”.

La forma en la que se obtienen los datos de DBMS sigue el esquema:

- Conexión con el gestor.
- Preparación de la consulta SQL.
- Ejecución de la consulta.
- Procesamiento del resultado obtenido en el cursor.
- Liberación de recursos.
- Cierre total de la conexión.

Función	Descripción
mysql_connect	Establece la conexión con un servidor, recibe como parámetro. el nombre de un servidor y opcionalmente el nombre de usuario y contraseña.
mysql_select_db	Selecciona la base de datos sobre la que se trabajara.
mysql_query	Ejecuta una consulta SQL.
mysql_num_fields	Devuelve un número de atributos y almacena el resultado de una consulta.
mysql_fetch_row	Avanza a la siguiente posición en una fila o tupla y obtiene el resultado de cada uno de los atributos de la fila mediante un array.
mysql_free_result	Libera los recursos asociados con un cursor.
mysql_close	Cierra la conexión establecida con el servidor.

Tabla III.3: Funciones de PHP para la utilización de MySQL

3.2.6 Conectarse

Empecemos con el primer script, que nos mostrará como conectarnos a una base de datos

(*conectarse.php*).

conectarse.php

```
<html>
<head>
<title>Ejemplo de PHP</title>
</head>
<body>
<?php function Conectarse()
{
If (!$link=mysql_connect("localhost","root","password")) {
echo "Error conectando a la base de datos.";
exit();}
if (!mysql_select_db("comil",$link)) {
echo "Error seleccionando la base de datos.";
exit();}
return $link;
}
Conectarse();
echo "Conexión con la base de datos conseguida.<br>";
?>
</body>
</html>
```

Como podemos ver en el ejemplo anterior se aisló lo necesario para la conexión en una función, ahora esa función la pondremos en un solo archivo PHP (*conec.php*).

conec.php

```
function Conectarse(){
if (!$link=mysql_connect("localhost","root","comil6")) {
exit();}
if (!mysql_select_db("basecurso",$link)) {
exit();}
return $link
;}
```

Ya que tenemos la función en un archivo separado solo hay que mandarlo llamar cuando sea necesario, de esta forma cuando tengamos aplicaciones complejas que use muchas páginas php y sea necesario cambiarle algo a la conexión solo se le tenga que hacer el cambio a este pequeño archivo y no a todas las páginas.

3.2.7 Agregar registros

Veremos un ejemplo de agregar registros a la base de datos (*insertareg.php* y *agregar.php*).

insertareg.php

```
<html>
<head>
<title>ejemplo de php</title>
</head>
<body>
<h1>ejemplo de uso de bases de datos con php y mysql</h1>
<form action="agregar.php">
<table>
<tr> <td>nombre:</td>
<td><input type="text" name="nombre" size="20" maxlength="30"></td>
</tr>
<tr> <td>direccion:</td>
<td><input type="text" name="direccion" size="20"maxlength="30"></td>
</tr>
<tr> <td>telefono:</td><td><input type="text" name="telefono" size="20"
maxlength="30"></td></tr>
<tr><td>email:</td><td><input type="text" name="email" size="20"
maxlength="30"></td></tr>
<tr><td>imagen:</td><td><input type="text" name="imagen" size="20"
maxlength="30"></td></tr>
</table>
<input type="submit" name="accion" value="grabar">
</form>
```

```
<hr>
<?php
include("conec.php");
$link=conectarse();
$result=mysql_query("select * from tablacurso",$link);
?>
<table border=1 cellspacing=1 cellpadding=1>
<tr>
<td> &nbsp;nombre</td><td> &nbsp;dirección&nbsp;</td><td> &nbsp;telefono&nbsp;</td><td> &nbsp;email&nbsp;</td><td> &nbsp;imagen&nbsp;</td>
</tr>
<?php
while($row = mysql_fetch_array($result)) {
printf("<tr><td> &nbsp;%s</td><td> &nbsp;%s&nbsp;</td><td> %s&nbsp;</td><td> &nbsp;%s&nbsp;</td><td> &nbsp;%s&nbsp;</td><td> &nbsp;<imgsrc=%s> &nbsp;</td></tr>
>",
$row["nombre"],$row["direccion"],$row["telefono"],$row["email"],$row["imagen"]);}
mysql_free_result($result);
?>
</table>
</body>
</html>
```

agregar.php

```
<?php
include("conec.php");
$link=conectarse();
$sql="insert into tablacurso
(nombre,direccion,telefono,email,imagen) values ('$nombre','$direccion',
'$telefono', '$email', '$imagen')";
mysql_query($sql,$link);
header("location: insertareg.php");
```


?>

3.2.8 Modificar registros

Veremos un ejemplo de modificar registros a la base de datos, consta de tres archivos diferentes, el primero para introducir la consulta por el campo nombre, el segundo para realizar los cambios necesarios y el tercero para modificar la base de datos (*consulta.htm*, *busca.php* y *modifica.php*).

consulta.htm

```
<html>
<head>
<title>ejemplo de php</title>
</head>
<body>
<h1>ejemplo de modificar</h1>
<form action="busca.php">
nombre: <input type="text" name="nombre" size="20" maxlength="30">
<input type="submit" name="accion" value="buscar">
</form>
</body>
</html>
```

busca.php

```
<html>
<body>
<?php include("conec.php");
$link=conectarse();
$sql="select * from tablacurso where nombre like '%$nombre%'";
echo $sql;
$result=mysql_query($sql,$link);
?>
<table border=1 cellspacing=1 cellpadding=1>
<tr>
```

```
<td>&nbsp;nombre</td>
<td>&nbsp;dirección&nbsp;</td><td>&nbsp;telefono&nbsp;</td>
<td>&nbsp;email&nbsp;</td><td>&nbsp;imagen&nbsp;</td></tr>
<form name="form1" method="post" action="modifica.php">
<?php
while($row = mysql_fetch_array($result))
{
printf("<tr><td><input type='text' name='nombre' size='20' maxlength='30'
value='%s'></td><td><input type='text' name='direccion' size='20' maxlength
= '30' value='%s'> &nbsp;</td>
<td><input type='text' name='telefono' size='20' maxlength='30' value='%s'>
</td><td>&nbsp;<input type='text' name='email' size='20' maxlength='30'
value='%s'</td><td>&nbsp;<input type='text' name='imagen' size='20'
maxlength='30' value='%s'> &nbsp;</td> </tr>
",$row["nombre"],$row["direccion"],$row["telefono"], $row ["email"],
$row["imagen"]);
}
mysql_free_result($result);
?>
</form>
</body>
</html>
modifica.php
<?php
include("conec.php");
$link=conectarse();
$sql="update tablacurso set nombre='$nombre', direccion='$direccion',
email='$email', telefono='$telefono' imagen='$imagen' where
nombre='$nombre'";
mysql_query($sql,$link);
header("location: consulta5.php");
?>
```

3.2.9 Eliminar registros

Pasemos a la eliminación de registros, este consta de dos archivos, los dos .php el primero es para elegir el registros a borrar y el segundo lo borra (*eliminarreg.php* y *borra.php*).

eliminarreg.php

```
<html>
<head>
<title>Ejemplo de PHP</title>
</head>
<body>
<h1>ejemplo de uso de bases de datos con php y mysql</h1>
<?php
include("conec.php");
$link=Conectarse();
$result=mysql_query("select * from tablacurso",$link);
?>
<Table border=1 cellspacing=1 cellpadding=1>
<tr>
<td> &nbsp;nombre</td><td> &nbsp;dirección&nbsp;</td><td> &nbsp;telefono&nbsp;</td><td> &nbsp;email&nbsp;</td><td> &nbsp;imagen&nbsp;</td>
<td> &nbsp;borra&nbsp;</td></tr>
<?php
while($row = mysql_fetch_array($result)) {
printf("<tr><td> &nbsp;%s</td><td> &nbsp;%s&nbsp;</td><td> &nbsp;%s
&nbsp;</td><td> &nbsp;%s&nbsp;</td><td> &nbsp;%s&nbsp;</td></td>
<td><a href=\"borra.php?
id=%d\">Borra</a></td></tr>", $row["nombre"], $row["direccion"], $row["te
lefono"], $row["email"], $row["imagen"], $row["ID"]);
}
mysql_free_result($result);
?>
```

```
</table>
```

```
</body>
```

```
</html>
```

```
borra.php
```

```
<?php
```

```
include("conec.php");
```

```
$link=Conectarse();
```

```
mysql_query("delete from curso where ID = $id",$link);
```

```
header("Location: eliminareg.php");
```

```
?>
```

CAPÍTULO IV.

ANÁLISIS DEL SISTEMA

Introducción

En el presente capítulo se describirá la más importante fase de la ingeniería de software el análisis ya que se debe tener bien claro que es lo que va realizar en esta página, con que se cuenta para realizarlo y que mejoras se pueden incorporar, debido a que si se hace primero el diseño o se desarrolla directamente se puede provocar inconsistencias en la aplicación por lo cual se tendría que hacer constantes modificaciones al código antes de obtener un resultado adecuado, lo que provoca pérdida de tiempo y esfuerzo.

4.1 Análisis de requerimientos de software

El análisis de requerimientos Es el proceso de establecer con todo detalle los servicios que debe proporcionar el sistema y las restricciones con las cuales debe operar , en pocas palabras este proceso ayuda a comprender hacia dónde va el proyecto, por lo cual se deben realizar actividades que se describen a continuación.

4.1.2 Clasificación de los requerimientos.

- **Requerimientos del entorno.**

Cuando se habla de entorno se debe pensar en todo lo que rodea al sistema, lo usa y lo necesita como una fuente de los servicios necesarios para que funcione. Aunque no podemos cambiar el entorno, existen ciertos tipos de requerimientos que se clasifican en esta categoría:

- a) El sistema debe ser operable a través de Web para poder ser accesible desde cualquier navegador de Internet por ejemplo Netscape o Internet Explorer.
- b) El sistema debe de ser robusto y tolerar los errores que puedan ocurrir en el entorno, tales como congestión en los dispositivos y errores de entrada de datos.

- **Requerimientos ergonómicos**

Si pensamos en los requerimientos ergonómicos debemos pensar en la forma en que el ser humano interactúa con el ser sistema físicamente y pues en este caso el sistema debe ser mayormente operado por el usuario a través del "Mouse" para evitar cansancio en las muñecas, manos y brazos.

- **Requerimientos de Interfaz**

La interfaz se puede decir que es la cara del sistema ante el usuario, es decir, es la manera formal en que los datos que el sistema recibe y procesa, manda al exterior. En este caso el usuario tendrá varias opciones a elegir distintas para cada usuario como son profesor alumno y administrador.

- **Requerimientos funcionales**

Tal como lo dice su nombre son los que describen lo que el sistema debe de hacer, básicamente éstos requerimientos conforme avanza el proyecto se convierten en los algoritmos o lógica aplicada, al final son gran parte del código de la página.

La página web debe utilizar una base de datos donde se guardará una lista de alumnos con sus calificaciones, profesores, archivos, noticias, etc. para que cuando un alumno solicite un recurso en base a sus datos el sistema genere el respectivo reporte tomando en consideración el nivel en que está matriculado.

- **Requerimientos de desempeño**

Estos requerimientos nos informan las características de desempeño que deben de tener el sistema. Debido al gran número de estudiantes pertenecientes al Colegio se tiene que considerar que la conexión al servidor y a la base de datos, debe soportar una sobrecarga de flujo de información, para evitar saturar al sistema y/o crear incongruencias en la base de datos, esto también depende de la robustez del servidor.

- **Entrenamiento**

Antes de poder administrar o utilizar dicha página, se debe entregar un manual al usuario final (administrador, profesor) para que conozcan el funcionamiento y alcances que tiene la Página y así facilite aun más su navegación.

4.2 Identificación de tipos de usuarios

Como se sabe la aplicación desarrollada está orientada al uso académico por lo tanto como primer usuario tenemos al alumno que será el que cuente con menos privilegios. Por otro lado tenemos al profesor a quien se le permite subir archivos para un determinado curso. A demás de un administrador el cual se encargará de mantener al sistema en caso de alguna contingencia, actualizar la base de datos cada vez que sea requerido subir noticias, etc. Por lo tanto, el administrador no tendrá ninguna restricción sobre el uso de la aplicación. Por todo lo anterior podemos concluir que tentativamente tenemos tres tipos de usuarios:

- Alumno
- Profesor
- Administrador

CAPITULO V

DISEÑO E IMPLEMENTACIÓN DE LA PÁGINA WEB

5.1 Introducción

Luego de la evidente revolución informática que se ha dado durante los últimos años, cabe destacar las necesidades y los avances que esta revolución provoca en una sociedad, como la transmisión eficiente y confiable de información de instituciones independientemente si ofrecen servicios o solamente quieren informar quienes son a la sociedad; por lo tanto, se hace de suma importancia tener un sitio en el internet, el cual permita la interacción e intercambio de información entre cualquier usuario y la institución. Las instituciones educativas al igual que las comerciales requieren promocionar sus servicios, sus carreras, objetivos y logros con el fin de que cada vez sean más los jóvenes que se integren al proyecto educativo de la institución.

Para ello es necesario combinar perfectamente las cualidades y adelantos de los equipos de cómputo con las habilidades de aprendizaje y acoplamiento humano, aplicando las más modernas técnicas para el desarrollo de Aplicaciones de software que se adapten adecuadamente tanto al entorno de trabajo como a las necesidades de los usuarios y clientes.

El Colegio “Combatientes de Tapi N°6” de la ciudad de Riobamba, necesita publicar su información y automatizar varios procesos que se efectúan en la Secretaría Académica del Colegio, de tal manera que se optimice todos los Recursos Humanos y Tecnológicos dedicados a esta tarea, se plantea por lo pronto, el desarrollo de una página web para la

promoción de la institución y gestión académica para los procesos permitiendo así, que este Colegio se involucre al mundo globalizado de la información.

5.2 Definición del problema

Dentro del departamento de secretaría en la Institución, se necesita facilitar el manejo de la información desde y hacia los usuarios, ya que pueden llegar a consumir demasiado tiempo debido a que se tiene una gran cantidad de información a procesar, además de que las soluciones para el manejo de alguno de estos asuntos se encuentran dispersas y no están concentradas en una sola aplicación. Es por eso que el resultado de esta monografía fue la creación de un sistema web de apoyo, que tiene la capacidad almacenar documentos digitales, dar avisos, elaboración de consultas, etc. De esta manera el personal académico y estudiantes pueden estar al tanto de los sucesos más recientes en la Institución.

Al ser una Institución educativa y de índole estatal tiene entre otros los siguientes problemas encontrados

- La obligación de la institución de publicar su información financiera y académica para ser revisada por las entidades de control.
- Pérdida de tiempo en lo relacionado a los procesos de matriculación y emisión de los diversos documentos requeridos por los estudiantes.
- Falta de control sobre la gran cantidad de documentos y archivos en secretaría.
- Falta de organización en la información.

Además que esta Institución cuenta ya con una base de datos pero destinada no solo a aspectos académicos sino, a aspectos financieros y administrativos que nada tienen que ver con el diseño de la página web, desarrollada bajo plataforma acces 2000 que será la fuente de datos del proyecto.

5.3 Definición del Alcance

La pagina web proveerá una herramienta para facilitar la promoción de servicios y simplificar varios procesos académicos como registro de datos del estudiante, registro de calificaciones, cálculo de promedios, subida y descarga de archivos, publicación de

noticias etc. del colegio “Combatientes de Tapi· Nº 6, con el fin de agilizar el proceso de análisis de datos y acceso a la información de una forma optima.

5.4 Alternativa de solución

Internet se ha convertido en una herramienta indispensable en los quehaceres de hoy en día, tanto académicos como de carácter personal. Provee de un sinnúmero de opciones para mostrar contenidos. Desde las tradicionales páginas web, hasta los tan de moda blogs. También se puede optar por contenido multimedia. Sin embargo, ya que el tipo de información que se manejará será, principalmente, texto, una buena alternativa es ir por el lado tradicional de página web.

Las páginas web que hay en Internet, se caracterizan por ser versátiles. Están hechas, principalmente, de 3 formas: mediante código estático (HTML, XML, etc); código dinámico (PHP, ASP, JSP, etc), mediante objetos incrustado (Flash, Java Applets). De esta forma su contenido no necesariamente es sólo texto.

Las páginas de código estático, consisten en archivos -en formato HTML o XML- que se envían directamente desde el servidor hasta el cliente. Este último es quien procesa su contenido para mostrarlo al usuario. Por otra parte, los de código dinámico consisten en archivos escritos en un lenguaje de programación, el cual es interpretado por el servidor antes de ser enviado al cliente. Aquellas páginas que contienen objetos incrustados, se desprenden del contenido tradicional de una página web, ya que pueden mostrar contenidos de otra índole (como sonidos y animaciones). Sin embargo, son dependientes de 'plug-ins' en el navegador del usuario para su correcto funcionamiento.

Lo que se busca es un sistema que no agobie al usuario a cumplir con requerimientos poco ortodoxos. Los 'plug-in' que requieren ciertas páginas -como FlashPlayer- para mostrar su contenido no están disponibles para algunos sistemas -como máquinas con Linux-, lo que iría en contra del principio de difusión masiva. Por otra parte, los navegadores web más populares (i.e. Internet Explorer, Mozilla Firefox, Opera, Konqueror) soportan características como JavaScript, Cascading Style Sheets (CSS) y marcos. Por lo que incluir dichas tecnologías en las páginas no debería excluir a ningún grupo de usuarios.

Luego de haber hecho un análisis comparativo entre las tecnologías de desarrollo web se llegó a la conclusión que las más optativas son las opciones de lenguaje ASP y PHP

con sus respectivas ventajas y desventajas con respecto a licencias, conocimiento de del lenguaje, disponibilidad de la Institución a una determinada tecnología, etc.

5.5 Análisis de las alternativas de solución

Debido a que el “COMIL N° 6” es una institución estatal y por normativa gubernamental todo sistema debe ser realizado bajo una plataforma de software libre, y al haber dos posibilidades de programación (PHP, ASP) previo análisis en base a la información recolectada, se propone el desarrollo del sistema en base a código PHP con conexión a base de datos en MYSQL.

5.6 Diseño funcional

En esta parte del desarrollo se establecen los aspectos del proyecto como son: contenidos, objetivos, estrategias, etc. En este momento ya se tiene el conocimiento necesario de lo que contiene la base de datos para el control académico y las tareas que debe satisfacer la pagina web conectada a esta

5.6.1 Objetivos

5.6.1.1 Objetivos generales y específicos

a) Objetivos generales

El propósito principal del trabajo fue la creación de una herramienta que facilite el flujo y manejo de la información Institucional, manejada por la secretaría, tales como la elaboración de reportes, avisos, publicación de notas, etc.

b) Objetivos específicos

- Diseño y creación de un sistema web de apoyo académico de los profesores hacia los estudiantes donde se tendrá un módulo de administración y visualización.

- Explorar las tecnologías PHP y SQL así como también algunos sistemas basados en ellas.
- Evaluación de los sistemas desarrollados y elaboración de una lista de recomendaciones y ventajas de acuerdo al análisis de las tecnologías ya mencionadas.

5.6.2 Diseño propuesto

Como el sistema quiere sacar provecho de las ventajas y utilidades de Internet, resulta lógico pensar en que se necesitará un servidor de páginas web. El servidor web será el encargado de recoger las peticiones desde Internet.

Por otra parte, se debe contar con un sistema de almacenamiento de datos. Este debe ser capaz de almacenar la información relacionada con los diversos registros que el sistema manejará. Es frecuente tener un sistema conformado por un servidor web y un repositorio de datos, tales como una base de datos.

Si se tiene un sistema conformado por un servidor web y una base de datos, las páginas web de contenido estático son inapropiadas. Se requiere de un sistema con dinamismo, con capacidad para mostrar los distintos datos almacenados según se le solicite. Es por ese motivo que se necesita que el servidor web sea capaz de interpretar páginas web escritas en algún lenguaje de programación para que permita un contenido dinámico que sea capaz de generar las distintas vistas, así como también enviar las notificaciones. Por otra parte, debe estar conectado a una base de datos la que contendrá la información necesaria para generar las vistas y los datos de los usuarios del sistema.

5.6.2.1 Requisitos específicos

Para el funcionamiento de la página desarrollada, se requiere conocer los pasos que se debe seguir para el correcto desarrollo.

- **Introducción:** La introducción de los datos correctamente es importante para la obtención de los reportes (tabla). Es decir cualquier software debe permitir que se cumpla algunas fases las cuales son:
 - Ingresos de datos
 - Procesos de información
 - Obtención de resultados.

- **Entradas:** El usuario primero debe elegir la acción que desea realizar (menú), ya que esto implica la entrada al sistema para luego realizar los procesos y salidas.
- **Procesos:** procesa la información que ingresa al sistema (conexión a la base de datos), y realiza: ingresos, consultas, actualizaciones, entre otros procesos.
- **Salidas:** La página sistema presenta o permite visualizar la información de acuerdo a las opciones escogidas por los usuarios (botones, menús), la salida de las correspondientes visualizaciones.

- **Atributos**
 - **Disponibilidad:** El sistema estará disponible para todos los usuarios que tengan acceso a internet independientemente del sistema operativo que se tenga.
 - **Confiable:** Se realizarán los correctivos necesarios, con el propósito de que la página sea confiable para los usuarios.
 - **Precaución:** Se debe considerar todos los requerimientos específicos a fin de que la aplicación al ser implantada cuente con todo lo necesario.

5.6.2.2 Requerimientos funcionales

En este capítulo se describen las necesidades que se deben ser cubiertas por el sistema. Sus funcionalidades mínimas que garanticen el desempeño deseado.

Casos de Uso

Los casos de uso enseñan un flujo de acciones que los usuarios y/o sistema sigue para lograr una acción requerida. Algunos casos de uso para el sistema desarrollado son:

<i>Inicio de Sesión</i>
Descripción: Procedimiento para iniciar una sesión como usuario autorizado en el sistema.
Actores: Usuario, Sistema
Condiciones: El usuario cargó la página principal del sistema en su navegador de Internet.
Flujo: <ol style="list-style-type: none">1. El Sistema muestra el formulario de ingreso.2. El Usuario completa los datos con su “<i>nombre de usuario</i>” y <i>contraseña</i>.3. El Sistema verifica los datos ingresados.4. El usuario es autorizado.5. Se carga la página de inicio de un usuario.
Flujo alternativo: <ol style="list-style-type: none">4b. El usuario no es autorizado.5b. Se carga nuevamente la página principal del sistema.

Tabla V.1.: Caso de Uso – Inicio de Sesión.

<i>Agregar archivo</i>
Descripción: Procedimiento para subir al servidor un archivo .
Actores: profesor, Sistema
Condiciones: El usuario ha iniciado sesión.
Flujo: <ol style="list-style-type: none">1. El Usuario selecciona la opción 'subir archivo'.2. El Usuario completa los datos requeridos para el fin.3. El Sistema recoge los datos recibidos.4. El Sistema almacena los datos.

Tabla V.2.: Caso de Uso – Agregar archivo.

<i>Subir Noticias</i>
Descripción: Procedimiento para subir al noticias al servidor .
Actores: administrador, Sistema
Condiciones: El usuario ha iniciado sesión.
Flujo: <ol style="list-style-type: none">1. El Usuario selecciona la opción 'subir noticias'.2. El Usuario completa los datos requeridos para el fin.3. El Sistema recoge los datos recibidos.4. El Sistema almacena los datos.

Tabla V.3.: Caso de Uso – Agregar archivo.

<i>Desplegar notas</i>
Descripción: Procedimiento para mostrar las notas de un alumno.
Actores: alumno, Sistema
Condiciones: El usuario ha iniciado sesión en el sistema.
Flujo: <ol style="list-style-type: none">1. El Usuario selecciona la opción ' notas'.2. El sistema recoge los parámetros enviados en la petición, los procesa y genera una página web dinámica a partir de los parámetros recibidos

Tabla V.4.: Caso de Uso – Desplegar notas.

<i>Descargar archivo</i>
Descripción: Procedimiento para descargar un archivo del sistema .
Actores: usuario, Sistema
Condiciones: El usuario ha iniciado sesión.
Flujo: <ol style="list-style-type: none">1. El Usuario selecciona la opción 'archivos' .2. El sistema recoge los parámetros enviados en la petición, los procesa y genera una página web dinámica a partir de los parámetros recibidos.3. El Sistema descarga archivo seleccionado por el usuario.

Tabla V.5.: Caso de Uso – Agregar archivo.

<i>Agregar publicación</i>
Descripción: Procedimiento para crear ingresar una publicación al sistema.
Actores: Usuario, Sistema
Condiciones: El usuario ha iniciado sesión en el sistema.
Flujo: <ol style="list-style-type: none">1. El Usuario selecciona la opción 'Agregar noticia'.2. El Usuario completa los datos requeridos para el tipo de publicación a ingresar.3. El Sistema recoge los datos recibidos.4. El Sistema almacena los datos.

Tabla 3.6.: Caso de Uso – Agregar noticia.

<i>Desplegar una Vista</i>
Descripción: Procedimiento de despliegue de una Vista.
Actores: Usuario, Sistema.
Condiciones: Algún usuario ha configurado una vista e insertado el código correspondiente en una página web.
Flujo: <ol style="list-style-type: none">1. El Usuario carga la página web en su navegador.2. El sistema recoge los parámetros enviados en la petición, los procesa y genera una página web dinámica a partir de los parámetros recibidos.3. El Usuario ve la página web en su navegador.

Tabla V.7.: Caso de Uso – Desplegar una Vista.

5.6.3 Modelado de datos

El modelo de los datos no es algo trivial dentro del diseño. Corresponde a como están organizados los datos dentro de la base de datos. Un mal diseño puede significar redundancia de datos, tiempos excesivos en las consultas, entre otros.

Al momento de diseñar el modelo se tuvieron ciertas consideraciones:

La institución posee una base de datos desarrollada en acces 2000 en la que se administra secciones académicas, administrativas y económicas, de donde basamos el desarrollo de nuestra base, para lograr una fácil actualización de datos.

A continuación, se muestra una lista de las tablas que conforman la base de datos del sistema, en ella se explica brevemente la función de cada una, dichas tablas son necesarias para el acceso, recuperación y despliegue en pantalla de datos.

Nombre de la tabla	Atributos	Tipos	Función de la tabla
archivo	<ul style="list-style-type: none"> • cod • nombre • cod_curs • cod_esp • cod_par • comentario • ruta 	<ul style="list-style-type: none"> • int • varchar(255) • int(2) • int(2) • int(2) • text • int(2) 	Contiene la información de los archivos ubicados en el servidor para ser descargados por los alumnos
curso	<ul style="list-style-type: none"> • cod_curso • nombre_c 	<ul style="list-style-type: none"> • int(4) • varchar(40) 	Contiene el curso al cual pertenece el estudiante
especialización	<ul style="list-style-type: none"> • cod_esp • nombre_esp 	<ul style="list-style-type: none"> • int(2) • varchar(40) 	Se utiliza para la asociación especialidades en las consultas
estudiante	<ul style="list-style-type: none"> • cod_est • nomb_est • apellid_est • tel_est • ced_est 	<ul style="list-style-type: none"> • int(5) • varchar(40) • varchar(40) • varchar(10) • varchar(11) 	Contiene los datos de los estudiantes matriculados en la Institución
materia	<ul style="list-style-type: none"> • cod_mater • nombre_mater • abreviat 	<ul style="list-style-type: none"> • int(2) • varchar(40) • varchar(10) 	Para la asociación de materias de la Institución
nota	<ul style="list-style-type: none"> • cod_est • cod_mater • cod_esp • cod_curso • periodo • cod_par • not_1 • not_2 • not_3 • promedio 	<ul style="list-style-type: none"> • int(3) • int(3) • int(2) • int(2) • int(4) • int(2) • float • float • float • float 	Contiene las notas de los estudiantes con sus respectivos datos para ser visualizados por el usuario alumno.
noticias	<ul style="list-style-type: none"> • fecha • autor • titulo • contenido • corto • imagen 	<ul style="list-style-type: none"> • int(5) • varchar (50) • text • longtext • text • varchar(255) 	Contiene las noticias publicadas por la Institución con la dirección de la imagen y datos ordenados por fecha
paralelo	<ul style="list-style-type: none"> • cod_par • nombre_par 	<ul style="list-style-type: none"> • int(2) • varchar(40) 	Datos de los paralelos matriculados
usuario	<ul style="list-style-type: none"> • usuari • clave • nombre • tipo 	<ul style="list-style-type: none"> • varchar(40) • varchar(40) • varchar(40) • int(2) 	Usuarios y claves que tienen permisos para acceder a la base de datos

Tabla V.8: Esquema de tablas de la base de datos de la página web

5.6.4 Funcionalidades

Las funcionalidades corresponden a las cosas básicas que el sistema debe ser capaz de hacer y deben tenerse en cuenta al momento de diseñar. A partir de los casos de uso se pueden determinar funcionalidades para el sistema. Tales como:

- *Mostrar información concerniente a la Institución.* Tener un portal web que promocioe a la Institución y sus servicios.
- *Autorizar usuarios, y denegar acceso a usuarios no autenticados.* Tener un sistema de autenticación que permita a los usuarios iniciar sesión en el sistema y sean identificados inequívocamente.
- *Permitir el ingreso de noticias.* Que mediante la interfaz web del sistema sea posible ingresar a un modulo de noticias.
- *Permitir la gestión de archivos.* Que mediante la interfaz web del sistema sea posible subir archivos para que estos puedan estar disponibles para descargas posteriores con los permisos respectivos.
- *Generar vistas.* El sistema debe ser capaz de generar visualizaciones de consultas realizadas por los usuarios como notas, noticias, galería, etc...
- *Acceso restringido.* De alguna forma, el sistema debe tener un acceso limitado. Debido a que será posible que terceros puedan ingresar registros, se debe limitar el acceso a esta funcionalidad para evitar SPAM o inundación de información inapropiada.

5.6.5 Herramientas

Las herramientas consisten en las distintas posibilidades de software que hay al alcance para poder implementar el sistema. Las distintas herramientas a elegir deben tener la capacidad de poder interactuar entre ellas y de esta forma potenciarse mutuamente.

Como criterio importante en la elección del software, se consideró el costo de la licencia asociado al uso del programa. Por lo cual la balanza se inclinó hacia el lado de los software de código libre.

5.6.5.1 Base de datos

Se utilizará para mantener una relación y coherencia entre la información archivada, los profesores, responsables, autores, etc. De esta forma, la búsqueda e indexación de la información será eficiente y sencilla desde el punto de vista del usuario.

En la elección de un software, se opto por la plataforma MYSQL por ser de código libre y fácil manejo.

5.6.5.2 Lenguaje programación

Es el nexa entre la base de datos y el servidor web. Sirve para agregarle dinamismo al sistema, ya que es posible generar páginas web de forma dinámica. Básicamente consiste en un lenguaje de programación de alto nivel que se interpreta por el servidor, lo que permite crear diferentes contenidos para una misma página, crear conexiones a bases de datos, crear contenidos personalizados, etc.

De entre todas las alternativas existentes hoy en día -PHP, ASP, PERL, JAVA- la balanza se inclinó por PHP. Este lenguaje provee al sistema todas las fortalezas de la programación.

5.6.6 Programación y elaboración del sistema

En la etapa de programación y elaboración del sistema se realiza la codificación de los componentes del sistema de información, a partir de las especificaciones requeridas, así como la construcción de los procedimientos de operación establecidos para el mismo.

El archivo "Index.php" es la página Web que se levanta al iniciar el servidor la cual contiene la interfaz gráfica dividida claramente en tres secciones: una superior conformada por un banner de la institución y los datos de fecha y sesión; una izquierda, donde se ubica la barra de menú y el cuadro de inicio de sesión; y finalmente una sección central donde se muestra la información seleccionada.

5.6.7 Implementación y pruebas

Conociendo los requerimientos incluidos en los diagramas de casos de uso en los numerales anteriores, se dispone de suficientes detalles para generar un código que se utiliza en el dominio de los objetos.

Esta etapa es conocida como puesta a punto del nuevo sistema. Una vez que se ha generado el código, comienza la prueba del programa. La prueba se centra en la lógica interna del software, asegurando que toda la sentencia se ha probado, y en las funciones externas realizando pruebas se aseguran que la entrada definida produce resultados que realmente se requieren. En este momento se pueden realizar todo tipo de pruebas: Individuales de programas; del Sistema integrado, tiempo de desempeño, pruebas de almacenamiento.

Esta es la fase final en donde se pone en marcha el nuevo sistema, se determina la eficiencia del mismo. En esta etapa se realizan las siguientes actividades:

Instalación del sistema realizado.

Capacitación al personal

Evaluación del sistema.

5.7 Desarrollo de Manuales del Sistema

La elaboración del software no solo corresponde a los programas elaborados en el lenguaje de programación seleccionado o herramienta, sino también a la documentación, que permite llevar un control del manejo y mantenimiento del sistema.

El sistema contiene información a la que se hace referencia en los manuales de usuario:

Manual de Usuario: Contiene información necesaria que guía al usuario en el correcto manejo del sistema. *Ver Anexo A*

CONCLUSIONES

- Hoy en día es imposible negar el impacto que tiene Internet y la WWW en particular que está teniendo sobre el modo de trabajar en muchas organizaciones. Es una gran herramienta para la difusión de contenidos, su gran penetración dentro de la sociedad moderna facilita el intercambio de conocimiento entre las personas.
- La página web desarrollada es una herramienta de difusión y promoción; que además da un soporte académico al Colegio Militar N° 6 Combatientes de Tapi cumpliendo con el objetivo de mantener de manera organizado un registro de información académica. Y, permitiendo su difusión de forma sencilla centralizada, evitando incoherencias al momento de publicar en múltiples sitios web.
- Como principal característica del sistema se tiene su simplicidad. Se basa en lenguajes de programación modernos y de gran popularidad, como lo es PHP y MYSQL.
- Aun cuando la Página web desarrollada se puede implementar, eso no significa que su desarrollo deba detenerse. La aplicación tiene la capacidad de seguir evolucionando e integrando nuevas capacidades para poder ofrecer una mayor gama de servicios.
- En el sitio web desarrollado presenta información, que a pesar de utilizar cualquier navegador el sistema responde de manera eficiente al momento de mostrar los datos requeridos.

RECOMENDACIONES

- Para un correcto desarrollo de un sistema informático, este debe cumplir con todos los requerimientos de una institución, es aconsejable ser minuciosos al momento de recopilar la información necesaria.
- Para el desarrollo de páginas o sistemas web se recomienda la selección de una metodología adecuada para que facilite la creación de la aplicación y evitar problemas posteriores.
- La página web desarrollada es una gran herramienta de información, sin embargo es recomendable que se vaya ampliando el campo de gestión, de acuerdo a las nuevas necesidades que aparezcan sobre la marcha.
- Las modificaciones posteriores al sistema web deben ser realizados por personal técnico con conocimientos tanto de programación como HTML, PHP y SQL
- Al momento del desarrollo de una página web es recomendable establecer una resolución gráfica que cumpla con los requerimientos básicos para ser visualizada desde cualquier computador.

RESUMEN

La finalidad de este trabajo es diseñar e implementar un sistema web que permita la promoción, registro y visualización mediante páginas web, de actividades académicas del “Colegio Militar N° 6 Combatientes de Tapi” ubicado en la ciudad de Riobamba.

El sistema web académico se ha desarrollado de acuerdo a las etapas del método deductivo con herramientas como Dreamweaver 8, para su codificación, Appserver 2.0 como servidor web y de base de datos (PHP MYSQL), a demás de Adobe Photoshop Cs3 para edición de imágenes.

El sistema implementado funciona mediante interfaz web, conectada a un servidor de aplicaciones PHP y con conexión a una base de datos desarrollada en MYSQL, con el cual se obtuvo la aprobación de un 95% como un sistema satisfactorio durante las pruebas del mismo por parte del personal de la Institución, cumpliendo así con los objetivos propuestos.

La página web desarrollada es una herramienta de difusión y promoción; que a demás de ser un soporte académico del Colegio, mantiene de manera organizada el registro de información académica, permitiendo su difusión de forma sencilla y centralizada y evitando incoherencias al momento de publicarla en la web

Las modificaciones posteriores al sistema web, deben ser realizados por personal técnico con conocimientos tanto de programación como HTML, PHP y SQL

ANEXO A

MANUAL DE USO DEL SITIO WEB

A lo largo de este apéndice se abordará el manual de usuario de sitio web. En este manual se describe el funcionamiento y explicación de cada una de las secciones del sitio web.

A.1 Plataforma.

El sitio web ha sido desarrollado en lenguaje PHP con conexión de base de datos MySQL, para aprovechar las ventajas con que cuenta este lenguaje.

Este lenguaje permite que sea fácilmente extensible agregando nuevos eventos, que también sean escritos en estos lenguajes.

La parte dinámica del sitio web se ejecuta dentro del servidor que interprete el código PHP y permita la conexión a una base de datos MySQL.

A.2 Perfiles de usuario

Los niveles de seguridad y roles de usuario son muy importantes dentro de un sistema informático, es por ello que existen 3 tipos de usuarios para trabajar en el Sistema web, estos son:

- **Usuario Administrador:** Este usuario tiene control total sobre la página, tiene derecho a todas las secciones del sistema.

- **Función:** Su función principal es la administración de los usuarios del sistema y sus datos, como también la administración de la galería de fotos y el modulo de noticias.

Características El usuario debe tener conocimiento de administración de sistemas, debe tener experiencia en el uso de aplicaciones web y tener un alto grado de responsabilidad. Se recomienda al administrador de red o de sistemas del Colegio.

- **Usuario profesor:** Usuario con privilegios para trabajar con el módulo de profesor.php donde se le permite subir archivos para descargas por parte de los alumnos. No tiene acceso a la administración del sistema

Función Su función principal es la preparación de archivos para subirlos al servidor, para un determinado curso matriculado en el presente año lectivo.

Características del usuario Conocimientos básicos en el manejo de aplicaciones web y un alto grado de seriedad y responsabilidad A este tipo de usuario corresponde los profesores de la Institución.

- **Usuario Alumno:** Usuario con privilegios para visualizar sus notas y descargar archivos en la página alumno.php

Función consulta de sus notas y descargas de archivos subidos al servidor por el usuario profesor

Características del usuario Conocimientos generales en el uso de aplicaciones web. A este grupo corresponde los alumnos matriculados en la Institución

A3 Base de datos

Para subir datos a la base hay que realizar las consultas creadas en la base de datos de Access que ocupa la Institución, estas son consulta estudiante, micurso, miespecializacion, mi materia mi paralelo, nota, usuario que se muestran en la figura A1 A continuación desde el servidor importamos cada una de los archivos de texto por tablas correspondientes (figuraA2)

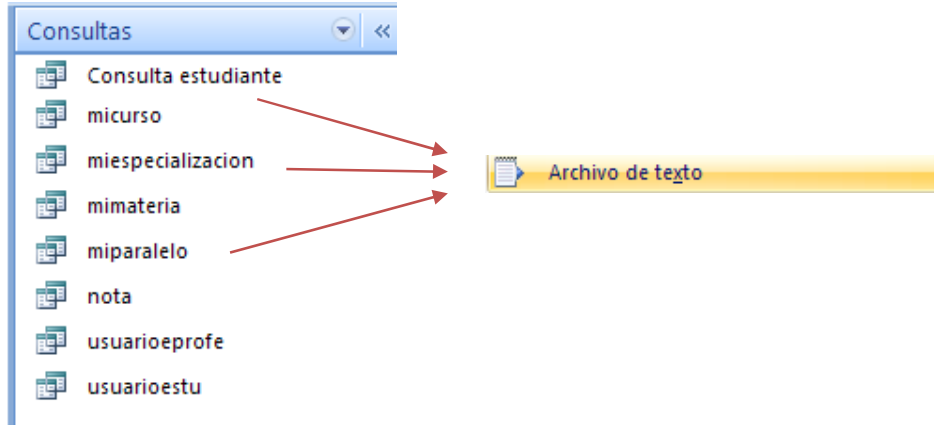


Figura A1.: exportar datos

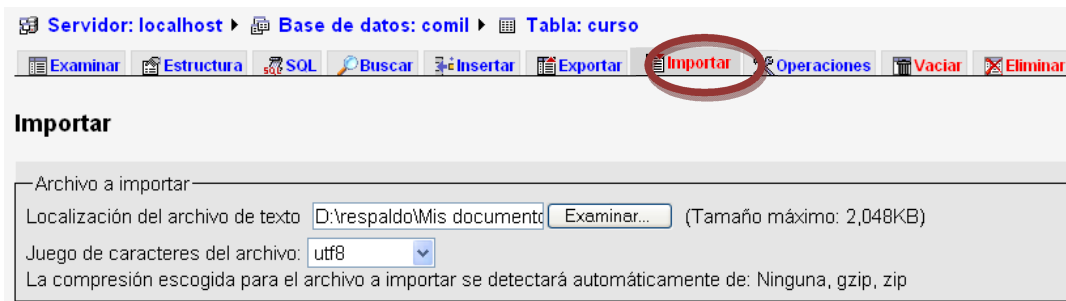


Figura A2.: Importar datos

A.4 Estructura de la página.

El sitio web está compuesto por varias pantallas con un diseño definido (figura A1), dispuesto por una imagen en la parte superior, a la izquierda el menú del sitio web con la entrada de registro de usuario, en la parte central el área de trabajo y en la parte inferior el pie de página.

The screenshot shows the homepage of the Colegio Militar "Combatientes de Tapi". At the top left is the school's logo, which features a shield with a map of Ecuador and the text "COLEGIO MILITAR COMBATIENTES DE TAPI". To the right of the logo, the school's name is written in a stylized font. Below the logo and name is a banner image showing military personnel in uniform. On the right side of the banner, the date "Jueves, 17 de Septiembre de 2009" is displayed.

On the left side of the page, there is a vertical navigation menu with the following items: Inicio, Colegio, Oferta académica, Instrucción Militar, Galería, Noticias, Ley de transparencia, Contactanos, and Descargas. Below the menu is a login section with the slogan "Solo venciéndote, vencerás" and a small version of the school's logo. The login section includes a text box for "Introduce tu clave de acceso", fields for "Usuario:" and "Contraseña:", an "ENTRAR" button, and a search bar with a "Buscar" button. There are also radio buttons for "WWW" and "Comil".

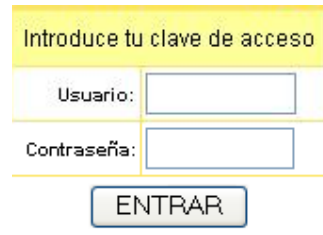
The main content area features a large image of two military officers saluting, with the text "548 visitas" below it. Below the image is a "NOTICIAS" section with four news items, each accompanied by a small thumbnail image:

- Nuevos Laboratorios De Computación En El Comil N° 6**: »La introducción de las Nuevas Tecnologías en los Centros Educativos, viene a replantear el proyecto educativo,... [Leer más](#)
- Fortalecimiento Del Perfil Profesional Del Docente**: »Las autoridades del Colegio Militar N° 6 "Combatientes de Tapi" de la Ciudad de Riobamba, considerando que la ciencia y la tecnología ... [Leer más](#)
- El Colegio Militar "Combatientes De Tapi" Y Su Integración Con La Comunidad Chimboracense**: »El Colegio Militar N° 6 "COMBATIENTES DE TAPI" en cumplimiento a una de las políticas educativas del plan decenal... [Leer más](#)
- Elección De La Señorita Novato**: »El área de cultura estética, dando cumplimiento a la disposición de las ... [Leer más](#)

At the bottom of the page, there is a "Solo venciéndote." slogan in a black oval, followed by the school's contact information: "Campus Académico: Av. de los Héroes, Antonio José de Sucre" and "Teléfonos: Rectorado 032942305 - Secretaría General 032942249 Riobamba - Ecuador".

Figura A3.: Estructura del sitio web

Una vez introducido el URL correspondiente en el navegador, aparecerá la ventana inicio donde a parte del menú con información de la Institución nos muestra un módulo que solicita la introducción de usuario y contraseña (fig A2), la cual direcciona a la página correspondiente al usuario el cual puede ser administrador, profesor o alumno.



Introduce tu clave de acceso

Usuario:

Contraseña:

ENTRAR

Figura A.4: Panel de registro de usuario

A.5 Panel de menú administrador

Este menú (figura A.3) muestra opciones que se encuentra en dos estados, el primero se cuando cargamos sitio web, el otro cuando seleccionamos la opción del menú desplegable



Inicio

Administrar Datos 

estudiantes

usuarios

notas

archivos

Galería

Noticias

Figura A.3: Panel de menús administrador

A continuación se describe cada una de las opciones que muestra el menú desglosados en la tabla A.1.


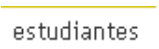
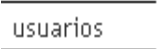
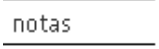
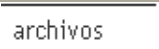


Icono	Función
	Direcciona a la página de inicio del sitio (index.php).
	Este menú nos permite administrar a los alumnos en la base de datos, nos genera información de un determinado estudiante, permitiendo su modificación.
	Esta opción nos genera datos de usuario registrado para poder eliminarlo o modificarlos
	Este menú es de mucha utilidad para la visualización de las notas de un determinado estudiante y si fuera necesario su modificación en la base de datos.
	Para la administración de los archivos subidos al servidor
	Este módulo permite subir, y borrar fotos a la galería.
	Permite administrar las noticias del sitio web.

Tabla A.1: Descripción del menú administrador

A.5 Menú estudiante

Esta opción mostrará un bloque de búsqueda de estudiante, la cual desplegará información del estudiante y permitirá modificarla o borrar el registro.

Nombre	Password	Modificar	Borrar
CABEZAS VALVERDE ERIK BLADIVIR	060322516-0	Modificar	Borrar

Figura A.5: Tabla de administración alumno

Notas.

Permite modificar las notas de los estudiantes que constan en la base de datos(figura A.5).

MATERIA	I TRIM	II TRIM	III TRIM	Modificar
Inglés	13.5	0	0	Modificar
Realidad Nacional	14.5	0	0	Modificar
Informática	16.5	0	0	Modificar
Teoría del Conoc. y Proyecto de Inv.	13.5	0	0	Modificar
Economía	17	0	0	Modificar
Cultura Física	18.5	0	0	Modificar
Química General	14	0	0	Modificar
Literatura General	15	0	0	Modificar
Matemáticas	15	0	0	Modificar
Anatomía-Biología	15	0	0	Modificar
Física	15	0	0	Modificar
Conducta	17.5	0	0	Modificar

Figura A.6: Tabla de administración nota.

A.6 Archivos.

Permite la administración de archivos para descarga ya sea del tipo general o para ley de transparencia que permite la página web, permitiéndonos escogerá que menú deseamos subir el archivo esto es del tipo general o ley de transparencia. Luego de haber llenado satisfactoriamente el formulario y de no exceder el archivo en dos megabytes nos muestra un mensaje de verificación.

Nota: para el usuario profesor se utiliza el mismo tipo de formulario para subir archivos.

NOMBRE:

descripcion:

TIPO: ▼

ARCHIVO:

Figura A.7: formulario de administración de archivos.

Galería. Permite subir fotos a la galería por álbumes, seleccionando la foto, con una descripción., además de la eliminación.

SUBIR FOTOS

Descripción

Album

 ▾

COMENTARIOS

TOTAL 4

Pág. : '11' '12' '13' '14'

Figura A.8: bloque administración galerías.

A.7 Noticias: Para llenar este formulario para subir noticias, se debe tener en consideración que existe un tamaño máximo soportado por el sistema web que es de dos megabytes. Fig A.8

TITULAR:

AUTOR:

FECHA:

IMAGEN:

CONTENIDO:

Figura A.8: bloque administración noticias.

BIBLIOGRAFÍA

BIBLIOGRAFÍA GENERAL

- FAIRLEY, Richard E. Ingeniería del Software. México: McGraw-Hill Interamericana, 1983. p. 265.
- GILMORE J. Beginning PHP and MySQL. 2ª ed. Philadelphia. EEUU: Paperback, 1998. 1450 p.
- JOYANES, L. y MARTÍNEZ, I. Estructura de Datos. Buenos Aires. Argentina: Macro, 2000. 620 p.
- KHRISTINE A, Dreamweaver MX 2004. Santa Cruz. Argentina: Anaya Multimedia, 2004. 608 p.
- MOLINARO, A. Curso de SQL. Santa Cruz. Argentina: Anaya Multimedia, 2006. 704 p.

BIBLIOGRAFÍA INTERNET

PROGRAMACIÓN WEB

- <http://www.webestilo.com/php/php05a.phtml>
(2008-11-10)
- <http://www.desarrolloweb.com/manuales/12/>
(2009-01-16)
- <http://dev.mysql.com/doc/refman/5.0/es/index.html>
(2009-01-16)
- <http://www.programatium.com/mysql.htm>
(2009-02-16)

BASE DE DATOS

- http://es.wikipedia.org/wiki/Base_de_datos
(2008-11-10)
- http://www.hipertexto.info/documentos/b_datos.htm
(2008-11-10)
- <http://www.monografias.com/trabajos11/basda/basda.shtml>
(2008-12-14)
- <http://www.mailxmail.com/curso-introduccion-access/que-es-base-datos>
(2008-12-14)

DISEÑO WEB

- <http://www.programacionweb.net/cursos/curso.php?num=18>
(2009-01-14)
- <http://es.wikipedia.org/wiki/DiseC3B1oweb>
(2009-02-14)
- <http://www.aulaclic.es/dreamweaver8/index.htm>
(2009-03-17)
- <http://www.webestilo.com/flash/>
(2009-03-20)