



Capítulo 3

Implementación del “Web-Site”

El presente capítulo abordará la problemática de la implementación. Es por tanto cuando nos enfrentamos directamente con el aspecto puramente tecnológico del proyecto. Pero antes de entrar directamente en materia, debemos conocer ciertos aspectos de la tecnología sobre la que se sustenta el mundo de Internet.

1. Introducción.

1.1 ¿Qué es Internet?

Internet no pertenece a ninguna compañía, simplemente, no pertenece a nadie, y se mueve mediante estándares que decide la propia comunidad de usuarios. Nació como una comunicación libre y espontánea entre redes de ordenadores; algo así como “¿por qué no conectamos mi red con la tuya?, yo me encargo de vigilar y mantener en funcionamiento mi sistema y tú el tuyo”. De esta forma tan simple, Internet ha crecido hasta llegar a millones de usuarios y convertirse en el sistema de comunicación más importante del mundo.

Internet no es más que una red de equipos y sistemas informáticos que se comunican entre sí para intercambiar datos. Es así de simple: Internet es un red de



ordenadores. Algunos ordenadores están conectados de forma permanente a Internet y otros lo hacen de forma temporal a través de la línea telefónica. Algunos son grandes equipos con microprocesadores avanzados y con sistemas operativos UNIX, mientras que otros, son PCs normales y corrientes que trabajan con el sistema operativo Windows. Esta es la idea más importante: Internet sólo es una red, es decir, un conjunto de ordenadores conectados físicamente entre sí mediante un cable. Lógicamente, si usted consigue conectarse a su vez a alguno de los ordenadores que forma parte de Internet (por ejemplo, a través del módem), entonces podrá acceder a cualquiera de los ordenadores de Internet. Eso sí, se trata de una red gigantesca formada por ciento de miles de equipos conectados por todo el mundo. Por ello, más que una red, Internet se ha de considerar una red de redes y para resaltar esa primacía es habitual denominar a Internet con el nombre de la Red (en mayúscula).

Todas las redes deben trabajar con un determinado protocolo de transferencia y utilizar un método de identificación de los ordenadores de la red. En el caso de Internet el protocolo utilizado es TCP/IP, y el método de identificación de los ordenadores es la dirección IP. Una vez establecida la comunicación con Internet, puede efectuar cualquiera de los servicios de Internet.

1.2 Arquitectura de protocolos TCP/IP.

El protocolo de transferencia de datos que utiliza Internet se llama TCP/IP. TCP/IP es un conjunto de normas y estándares entre ellos los dos protocolos que le dan nombre: TCP (Transmission Control Protocol) e IP (Internet Protocol), que se encarga de definir cómo se envían los datos y se establece la comunicación entre los componentes de una red. Desde el punto práctico, lo único que usted debe saber es precisamente eso: que el protocolo de Internet es TCP/IP y que todos los ordenadores que se conectan a Internet han de tener instalado y en funcionamiento el protocolo TCP/IP, pues en caso contrario no sabrían cómo enviar y recibir los datos. Pero no nos conformemos sólo con esto, y entremos un poco más en profundidad.



En realidad, no existe un modelo de protocolos TCP/IP “oficial”. Sin embargo, es útil caracterizar el conjunto de protocolos con un modelo de capas, como si tuviera cinco capas. Estas capas son:

- Capa de aplicación: proporciona una comunicación entre procesos o aplicaciones en computadores distintos.
- Capa de transporte o computador-a-computador: proporciona un servicio de transferencia de datos extremo-a-extremo. Esta capa puede incluir mecanismos de seguridad. Oculta los detalles de la red, o redes, subyacente a la capa de aplicación.
- Capa internet: relacionada con el encaminamiento de los datos del computador origen al destino a través de una o más redes conectadas por dispositivos de encaminamiento.
- Capa de acceso a la red: relacionada con la interfaz lógica entre un sistema final y una subred.
- Capa física: define las características del medio de transmisión, la tasa de señalización y el esquema de codificación de las señales.

En toda red es necesario identificar de forma unívoca y exclusiva a cada uno de los ordenadores de la red para poder enviar y/o solicitar datos a un equipo en particular.

En el caso de la red Internet se utiliza como identificación un número llamado dirección IP. Este número se representa mediante cuatro valores de 0 a 255 separados entre sí por un punto; por ejemplo, 172.16.1.48. Las direcciones IP siempre utilizan este formato especial, en realidad, una dirección IP es un único número de 32 bits y cada uno de los cuatro valores que se ven en su forma está representando un byte (8 bits) de ese número. Todos los ordenadores de Internet tienen una dirección IP que es única y exclusiva para ellos, es decir, no puede haber un ordenador en Internet sin dirección IP y no puede haber dos ordenadores con la misma dirección IP, que puede ser siempre la misma o, probablemente, su proveedor de Internet le asigne una diferente cada vez que se conecte.



1.3 Servicios de Internet.

Cuando un usuario se conecta a Internet, puede acceder a miles de equipos informáticos de todo el mundo. Ahora bien, ¿cómo interpretar la información de esos equipos? Para ello se utilizan los servicios Internet, que son estándares que defienden normas para representar y enviar la información que circula por Internet, de forma que el usuario pueda recuperarla e interpretarla correctamente. Desde el punto de vista del usuario, si desea utilizar un determinado servicio Internet simplemente debe trabajar con un programa que soporte el estándar y las especificaciones que indican cómo se ha de usar ese servicio. Por ejemplo, para utilizar el servicio el correo electrónico hay que trabajar con un programa que soporte el estándar SMTP/POP, que define el funcionamiento del correo electrónico Internet.

Un servicio Internet no es más que una norma o estándar aceptada por la comunidad e implantada en los programas. Por tanto, es habitual que surjan con frecuencia nuevos servicios a medida que aumentan las necesidades de los usuarios de Internet.

A continuación se comentan algunos de los principales servicios Internet, indicando en cada caso algún programa o herramienta que permite trabajar con dicho servicio.

Web. El servicio *World Wide Web* (WWW), abreviado normalmente como Web, es la oferta más atractiva que ofrece Internet actualmente. Se trata de un estándar para presentar y ver páginas de información que incluyen texto, imágenes, sonidos, animación, vídeos, etc. El programa cliente que lee las páginas Web recibe el nombre de navegador y su misión es muy sencilla: conectarse a un servidor Web para recibir sus datos y presentar en pantalla la página Web. Los dos navegadores más utilizados son Netscape e Internet Explorer, aunque antiguamente el navegador más popular era Mosaic.

Correo electrónico. El servicio Internet más utilizado es el correo electrónico. Todos los usuarios que tienen acceso a Internet poseen una cuenta de correo en el



formato nombre@compañía.organización (por ejemplo, pgomez@uid.es) y pueden transferir mensajes con cualquier otro usuario. Un equipo que funciona como servidor de correo se encarga de almacenar todos los mensajes. Usted, funcionando como cliente de correo, se conecta al servidor para recoger sus mensajes y enviar los que desee.

FTP. El servicio de FTP (*File Transfer Protocol*: Protocolo de Transferencia de Ficheros) permite transferir archivos entre dos equipos conectados a Internet. Existen programas especiales para conectarse a servidores FTP y enviar o recibir archivos, pero la mayoría de los navegadores, incluyendo Explorer, soportan el protocolo FTP desde dentro del propio navegador.

2. Modelo de comunicación.

Anteriormente, al describir alguno de los servicios de Internet, ha aparecido una característica que hasta el momento no habíamos mencionado para nada. Se trata del modelo de comunicación que utilizan todos estos servicios.

En todos los servicios de Internet siempre hay un ordenador que solicita el servicio (cliente) y un ordenador que ofrece el servicio (servidor); es decir, los servicios Internet siguen la arquitectura cliente-servidor. En teoría, cualquier ordenador puede funcionar como cliente y/o como servidor, incluso un ordenador puede estar funcionando en algunos momentos como cliente y en otros como servidor. Pero, en la práctica, lo más habitual es que existan ordenadores dedicados exclusivamente a ser servidores y otros, que se dediquen exclusivamente a ser clientes. Existen tres diferencias importantes entre clientes y servidores.

- **Diferentes programas.** El servidor y el cliente tienen instalados programas diferentes, aunque se conecten para usar el mismo servicio Internet. Por ejemplo, en el caso del servicio de las páginas Web, el servidor tiene instalado un programa (llamado servidor Web), que



permite enviar las páginas a los clientes que las soliciten, mientras que el cliente tiene instalado un programa que lee páginas Web (un navegador).

- **Diferentes tipos de conexión.** El servidor tiene que estar conectado a Internet de forma permanente, pues en caso contrario, un cliente intentaría acceder a él y no lo encontraría. Por el contrario, los clientes sólo han de estar conectados cuando piden información; es decir, un PC (cliente) conectándose a través de un módem puede desconectarse de Internet cuando ha recibido la información.
- **Diferentes ordenadores.** Un servidor ha de ser un equipo muy potente con una gran capacidad de almacenamiento y funciones de recuperación ante errores críticos como pérdida de datos o fallos en la corriente eléctrica, etc. Por el contrario, un cliente puede ser cualquier equipo que cumpla los requerimientos básicos para conectarse a Internet y ejecutar el programa cliente de un determinado servicio Internet (en general, hasta un 486 con 4 MB de memoria RAM y un módem, o una tarjeta de red).

El resumen de este apartado sobre cliente y servidor es muy sencillo, para usar un servidor Internet tiene que ejecutar un programa cliente que soporte el estándar de ese servicio y conectarse a un servidor que también implemente el estándar de ese servicio.

El servicio World Wide Web, es igual que cualquier otro servicio de Internet. Es decir, hay un ordenador cliente que pide información a un ordenador servidor utilizando un estándar de transferencia de datos. La información que se transfiere entre cliente y servidor recibe el nombre de páginas Web. Los elementos claves que se ven envueltos en la transmisión de páginas Web son los siguientes:

- **Cliente Web.** Es el ordenador que pide la información, acción que realiza a través de los programas Web, más conocidos como navegadores. Por tanto, todo ordenador que desee acceder al servicio



Web debe tener instalado un navegador, por ejemplo, Internet Explorer o Netscape Navigator.

- **Servidor Web.** Es el ordenador que ofrece la información. A veces estos servidores Web reciben el nombre de sitios Web (web-site), pero existe diferencia entre ambos conceptos: el servidor Web es un ordenador, mientras que el sitio Web es el conjunto virtual de páginas Web que almacena un servidor.

2.1 Protocolo de transferencia de hipertextos (HTTP).

El protocolo de transferencia de hipertextos (HTTP) es el protocolo base del “World Wide Web” y se puede utilizar en cualquier aplicación cliente-servidor que suponga la utilización de hipertextos. El nombre es más bien confuso ya que HTTP no es un protocolo para transferir hipertexto; en lugar de eso es un protocolo para transmitir información con la eficiencia necesaria para hacer que el hipertexto salte. Los datos transferidos por el protocolo pueden ser texto propiamente dicho, hipertexto, audio, imágenes o cualquier información accesible a través de Internet.

Para terminar con este apartado, describiremos de forma muy general el funcionamiento de este protocolo,.

HTTP es un protocolo cliente-servidor orientado a transacciones. El uso más común es entre un cliente web y un servidor web. Para proporcionar seguridad, hace uso de TCP. Sin embargo, HTTP es un protocolo “sin estados”: cada transacción se trata independientemente. Por consiguiente, una implementación típica creará una conexión nueva entre el cliente y el servidor con cada transacción y después la cerrará tan pronto como se complete la transacción, aunque la especificación no impone esta relación uno a uno entre la transacción y los tiempos de vida de la conexión.

La naturaleza de HTTP de ser un protocolo sin estados es la adecuada para su aplicación típica. Una sesión normal de un usuario con el cliente web supone obtener una secuencia de páginas y documentos web. La secuencia, de forma ideal, se obtiene



rápidamente y las localizaciones de las distintas páginas y documentos web pueden estar en una serie de servidores distribuidos mundialmente.

Otra característica importante de http es que es flexible en cuanto a los formatos que puede tratar. Cuando un cliente emite una solicitud a un servidor, puede incluir una lista de prioridades de formatos con los que puede operar, y el servidor responde con el formato adecuado. Por ejemplo, un cliente Lynx no puede operar con imágenes, por tanto el servidor no necesita transmitir ninguna imagen en las páginas web. Esta organización evita la transmisión de información innecesaria y proporciona la base para ampliar el conjunto de formatos de las especificaciones de las nuevas normalizaciones y de los propietarios.

3. Herramientas necesarias.

En esta parte del capítulo, tratamos de hacer una recopilación de todas aquellas herramientas que son necesarias para llevar a cabo la implementación del web-site que se ha diseñado.

3.1 Servidor de páginas Web Apache.

Este servidor, es actualmente el más ampliamente utilizado. Hablando históricamente, es un descendiente del httpd de NCSA, el primer servidor usado para la World Wide Web. Pero el crecimiento de este servidor no fue a la velocidad que querían muchos gestores de servidores Web. Así que un consorcio de esos gestores, usando httpd de NCSA como base, empezó a proporcionar su propio servidor Web, llamado Apache.

En una encuesta reciente de servidores Web en Internet, se vio que Apache era de los más populares y por tanto de los más utilizados. Nosotros también lo utilizaremos.

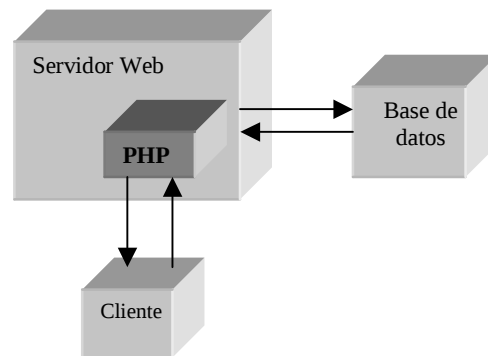


3.2 Plataforma para PHP.

Este servidor web Apache, tiene que estar compilado para incorporar las características PHP o admitir módulos dinámicos. Si nuestras necesidades no incluyen unas características muy especiales, quizá lo más cómodo es utilizar los binarios que incluyen las distribuciones y que ya vienen preparados para admitir las características más habituales.

PHP admite una gran variedad de módulos, pero las versiones precompiladas no los incorporan todos. Si necesita alguna funcionalidad concreta, es muy probable que tenga que compilar su propia versión de PHP.

Para evitar sorpresas, nosotros vamos a optar por esta última opción, nos hemos hecho con las últimas versiones del servidor Apache y de PHP4 en Internet, y las utilizaremos para instalarlas y configurarlas por nosotros mismos, incluyendo las opciones que nos hagan falta. En el primer apéndice de este documento, se puede encontrar los pasos seguidos para la instalación de Apache con PHP bajo Linux.





3.3 Sistema Gestor de Bases de Datos.

La funcionalidad con que PHP es capaz de consultar una base de datos le confiere una potencia excepcional a la hora de generar páginas web dinámicas. En la figura podemos ver la relación que se puede establecer entre PHP y un sistema gestor de bases de datos: nuestro programa puede realizar consultas y obtener los resultados para procesarlos. Este gestor de bases de datos puede estar ubicado en la misma máquina que ejecuta PHP o puede estar en otra máquina distinta y realizar las consultas a través de la red.

Como sistema gestor de bases de datos utilizaremos MySQL, debido a su licencia, a la posibilidad de obtener soporte comercial y su potencia. La gran mayoría de las aplicaciones desarrolladas con PHP con un gestor de bases de datos utilizan MySQL, un sistema gestor eficiente, suficiente para la mayoría de las aplicaciones que se desarrollan. Además, las últimas versiones que han surgido son muy prometedoras.

4. Programación en el lado servidor.

Este apartado pretende describir toda la programación que se ha llevado a cabo en el lado del servidor, programación necesaria para la implementación de los distintos servicios que se han diseñado.

A grandes rasgos, la programación en el lado servidor consiste en la implementación de algoritmos o procedimientos para realizar una cierta función, pero utilizando como interfaz una página web. Es decir, vamos a programar en lenguaje PHP (como se ha dicho anteriormente), que será interpretado posteriormente por el servidor web para generar código HTML de salida, que constituirá la página web que será la que devuelva el servidor al ordenador cliente.

Un programa PHP no se puede ver como un programa binario compilado que se ejecuta en la máquina cliente, tiene que verlo como un código interpretado en el



servidor que genera un código HTML, que a su vez tiene que ser interpretado en el cliente web del usuario que lanza el programa.

Hemos dicho que el objetivo de PHP es generar páginas HTML, por lo que su relación dentro de un programa, es muy estrecha. Tanto es así que el lenguaje PHP tiene que convivir en el mismo fichero con el lenguaje HTML y obliga a establecer unos criterios para tener los códigos respectivos convenientemente separados.

4.1 Implementación del servicio de autoevaluación.

Este servicio ya fue descrito y especificado en el capítulo anterior. Ahora pasaremos a describir como lo hemos implementado.

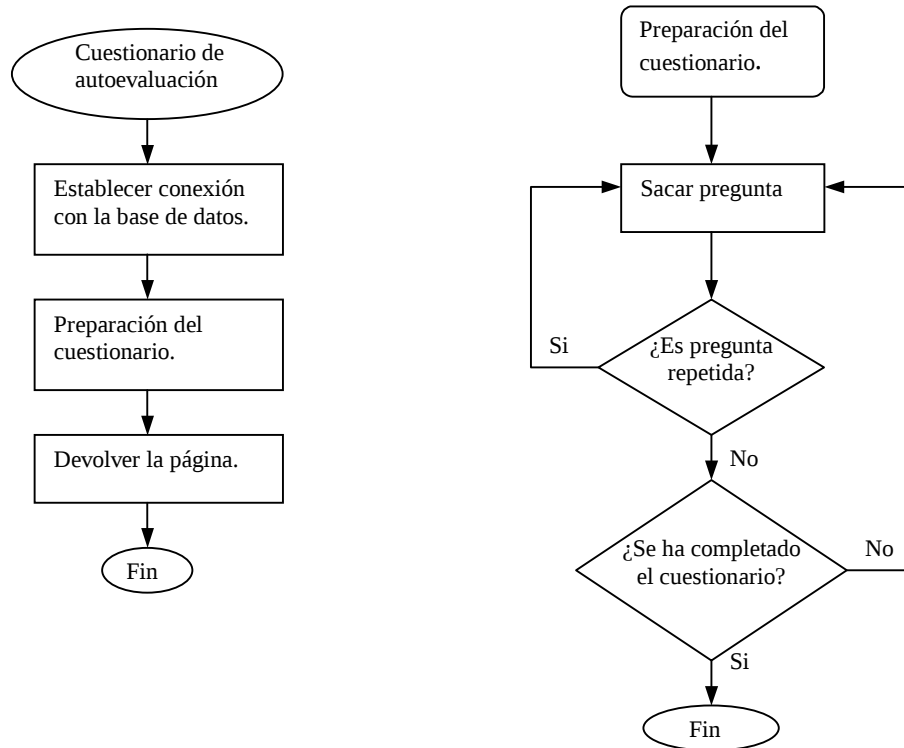
Para la implementación del servicio, lo primero que necesitamos es crear la base de datos “**cuestionario**”, que presenta el siguiente diseño:

Nombre: cuestionario	Nombre de campo:	Tipo:
Tabla: pregunta		
	ID (primary key)	Númérico
	Texto	Texto
	Solución	Carácter
	Tema	Texto
	Examen	Fecha
Tabla: opciones		
	ID (primary key)	Númérico
	Opciona	Texto
	Opcionb	Texto
	Opcionc	Texto
	Opciond	Texto



Esta base de datos contendrá toda la información necesaria para implementar el servicio. Guardará toda la información de las preguntas de los exámenes de otros años, junto con todas las opciones posibles, la respuesta correcta y cierta información adicional como puede ser el tema al que pertenece la pregunta o la fecha del examen en que apareció por primera vez.

A partir de aquí, nuestro programa tendrá que seguir la siguiente secuencia lógica que a continuación representamos utilizando unos diagramas de flujo de proceso:



En el diagrama de la izquierda se muestra la línea principal de ejecución del programa, desde que se hace la llamada al programa hasta que este devuelve la página de resultado. Esta página de resultado es la página que lleva el test de autoevaluación.

El diagrama de flujo de la derecha explica el proceso seguido por una cierta función que se ha definido para la constitución del test a partir de los datos extraídos de la base de datos.



Dicha función pregunta a la base de datos de forma aleatoria para recoger una pregunta con sus correspondientes opciones, y al mismo tiempo va configurando la apariencia que debe mostrar la página web. Si la pregunta que se retoma de la base de datos resulta que ya ha salido antes, no se acepta y se vuelve a preguntar por otra de nuevo. Así se va comprobando que no se repite ninguna pregunta en el test y que su formación es totalmente aleatoria. Cada vez que se invoca al programa, dará como resultado un nuevo test, con preguntas distintas.

A continuación se adjunta el código del programa:

```
<?php

/***** función botonradio_db *****/
/* $f conexión */
/* $tipo formato V=vertical ... */
/* $tabla1 tabla primera de consulta */
/* $tabla2 tabla segunda de consulta */
/*****/

function botonradio_db ($f, $tipo, $tabla1, $tabla2)
{
    if ($tipo=="V") $TIPO="&nbsp; <BR>";
    else $TIPO="&nbsp; &nbsp;";

    $cons = "SELECT pregunta.texto,pregunta.id, opciona, opcionb, opcionc, opciond FROM $tabla1,
    $tabla2 WHERE pregunta.id = opciones.id ";
    $res = mysql_query ($cons, $f) OR die ("Fallo en la respuesta.");
    echo "<form action=gueno.php method=post >";

    for ($i = 0; $i < 10; $i++)
    {
        $j = 0;
        $k = 0;

        while (!$k){
            $sale = mt_rand ("0", "110");
            $cadena[$i] = $sale;
            if ($i == 0) $k=1;
            else{
                for ($k=0; $k<$i; $k++)
                    if ($cadena[$k] == $sale)
                        $k = 0;
            }
        };

        $valor = mysql_result ($res, $sale, $j);
        $i=$i+1;
        echo "<p><b>$i. $valor</b>". $TIPO;
        $i=$i-1;
        $j = 1;
    }
}
```



```
$id = mysql_result ($res, $ale, $j);
$name = "radio".$id;
for ($j = 2; $j < 6; $j++)
{
    $valor = mysql_result ($res, $ale, $j);

    echo "<input type='radio' name='$name' value='$valor' >$valor". $TIPO;

}
/* echo "<input type='hidden' name='id' value='$id'>"; */
$array=$id." ".$array;
}
echo "<input type='hidden' name='ids' value='$array'>";
echo "<p><input type=submit value=Corregir> ";
echo "<input type=reset value=Borrar></p>";
echo "</form>";

}

/*****/
/*****/
/*****/
/*****/
/*****/

$servidor = "localhost";
$usuario = "root";
$contraseña = "*****";

$base = "cuestionario";

$id = mysql_connect($servidor, $usuario, $contraseña);
$conexion = mysql_select_db ($base, $id) OR die ("No puedo conectar con
la base de datos '$base'." );
print ("Conexión establecida con éxito.\n<br><br>\n");

echo "\n<HTML>\n<HEAD>";
echo "<META HTTP-EQUIV='Content-Type'CONTENT='text/html;charset=ISO-8859-1'>";
echo "</HEAD>\n<BODY>\n<br><br>";

botonradio_db ($id, "V", "pregunta", "opciones");

mysql_close ($id);
/*print ("Se ha cerrado con éxito.");*/
?>

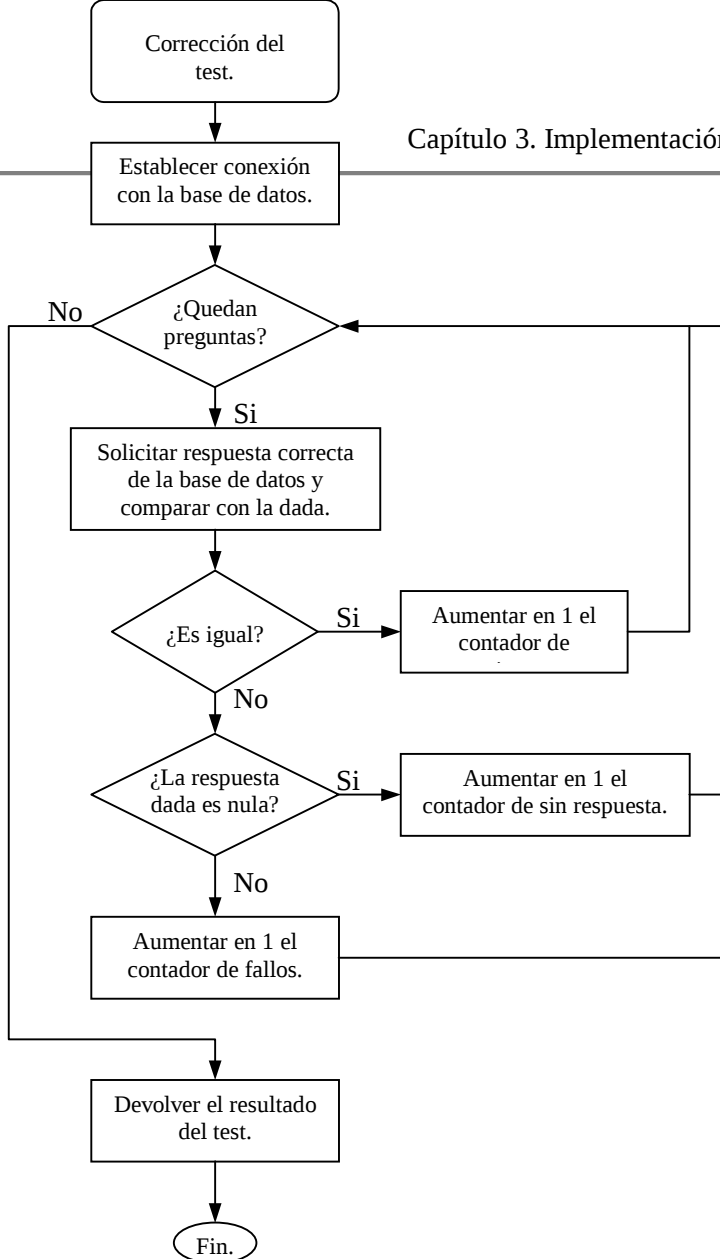
</BODY>
</HTML>
```

La segunda parte de la aplicación la constituye el programa que recibe el cuestionario una vez que se ha contestado y procede a su corrección. Dicho programa funciona de la siguiente forma:



“a partir de la identificación de la pregunta en cuestión, se obtiene de la base de datos cual es la respuesta correcta y posteriormente se compara con la que se ha dado, si las respuestas coinciden es que se ha contestado correctamente, si no coinciden es que se ha errado la pregunta. Si bien cabe la posibilidad de que la respuesta sea nula, en cuyo caso es que la pregunta se ha dejado sin contestar.”

Esto se puede ver mejor con ayuda del siguiente diagrama de flujo de la aplicación:



El anterior diagrama de flujo muestra la secuencia que sigue el programa que corrige el test de autoevaluación. Observamos que mientras quedan preguntas por corregir, se van obteniendo de la base de datos la solución y se va comparando con la respuesta. Al final de la ejecución, se devuelve una página web, que informa del resultado obtenido en la prueba.

A continuación se adjunta el código del programa:

```
<?php
```

```
/**
***** Intentamos la conexión con la base de datos.
*****
***/
```

```
$servidor = "localhost";
```





```
$usuario = "root";
$contraseña = "*****";

$base = "cuestionario";

$f = mysql_connect($servidor, $usuario, $contraseña);
$conexion = mysql_select_db ($base, $f) OR die ("No puedo conectar con
la base de datos \"$base\". ");
print ("Conexión establecida con éxito.\n<br><br>\n");

$contador=0;
$erroneas=0;
$sin=0;

$id=strtok($ids, ",");

while($id)
{
    $i = 0;
    $j = 0;

    $name = "radio".$id;

    $cons = "SELECT solucion FROM pregunta WHERE id=".$id;
    $res = mysql_query ($cons,$f) or die ("Fallo en la respuesta.");
    $valor = mysql_result($res, $i, $j);

    $opcion = "opcion".$valor;
    $cons = "SELECT ".$opcion." FROM opciones WHERE id=".$id;
    $res = mysql_query ($cons,$f) or die ("Fallo en la respuesta.");
    $valor = mysql_result ($res, $i, $j);

    if (strcmp(${$name},$valor) == 0) $contador=$contador+1;
    else if (!${$name}) $sin=$sin+1;
    else $erroneas=$erroneas+1;
    /*echo "${$name} <br>$valor<br>";*/

    $id=strtok(",");
}

print ("<br>Has acertado $contador preguntas.<br>");
print ("Has fallado $erroneas preguntas.<br>");
print ("Has dejado sin contestar $sin preguntas.<br>");
mysql_close($f);
?>
```

4.2 Implementación del servicio tablón de dudas.

Este servicio también fue descrito y especificado en el capítulo anterior. Ahora pasaremos a describir como lo hemos implementado.



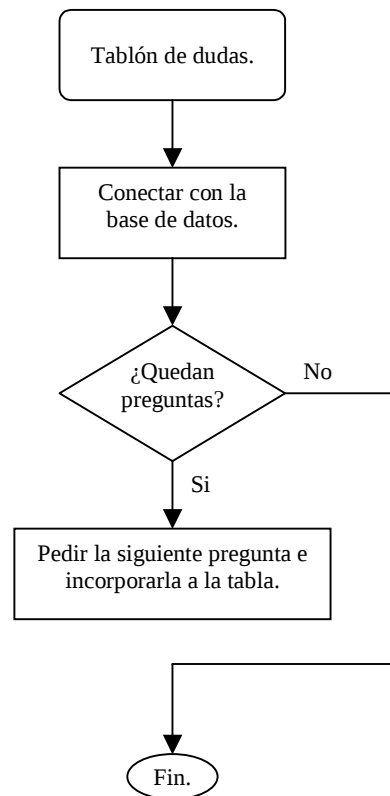
Para la implementación del servicio, también necesitamos crear una pequeña base de datos donde almacenar toda la información. Esta base de datos se llamará “**dudas**”, y presenta el siguiente diseño:

Nombre: dudas	Nombre de campo:	Tipo:
Tabla: dudas		
	ID (primary key)	Numérico
	Pregunta	Texto
	Respuesta	Texto
	Fecha	Fecha

Esta base de datos contendrá toda la información necesaria para implementar el servicio. El campo “pregunta” guardará el texto original de la pregunta que el alumno formuló al profesor y el campo “respuesta” la contestación ofrecida por éste último. También se ha incluido un campo fecha, para tener una idea de cuando se hizo esa pregunta por primera vez.

La primera parte de la aplicación la constituye un programa cuya misión será recoger de la base de datos todas las preguntas que existen almacenadas y prepararlas en una tabla para enviarlas al cliente que solicite el servicio. Esta tabla contendrá el enunciado original de las preguntas y una casilla de marcación para que el alumno pueda consultar de una sola vez más de una pregunta.

El siguiente diagrama de flujo muestra de una forma más esquemática lo que acabamos de explicar:



El anterior diagrama de flujo muestra la secuencia que sigue el programa. Podemos observar que mientras queden preguntas dentro de la base de datos se irán sacando una a una e incorporando en una tabla que posteriormente será la que se envíe al cliente.

A continuación se adjunta el código del programa:

```
<?php

/***** función tabladudas_db *****/
/* Muestra una tabla con todas las dudas almacenadas */
/* en la base de datos. */
/* $f conexión */
/*****/

function tabladudas_db ($f)
{

    $cons = "SELECT pregunta, fecha FROM dudas ";
    $res = mysql_query ($cons, $f) OR die ("Fallo en la respuesta.");

    /* Creamos un formulario donde establecemos una tabla con campos de texto */
```



```
echo "<form action=consultaduda.php method=post >";

?>
<center>
<table width="75%" border="1" cellspacing="1" cellpadding="1">
<?

$K = mysql_num_rows ($res);
for ($i=0; $i<$K; $i++){
    $j=0;
    $valor = mysql_result($res, $i, $j);
    $name = "caja".$i;
    echo "<tr><td>&nbsp;<input type='checkbox' name='\"$name\"' value='\"$valor\"'>$valor</td>";
    $j=1;
    $valor = mysql_result($res, $i, $j);
    echo "<td>&nbsp;<input type='checkbox' name='\"$name\"' value='\"$valor\"'>$valor</td></tr>";
}
echo "</table></center><br><br>";

echo "<p><input type=submit value=Consultar> ";
echo "<input type=reset value=Borrar></p>";
echo "</form>";
}

/***** Programa principal *****/

/*****
/* Estableciendo conexion con la base de datos. */
*****/

$servidor = "localhost";
$usuario = "root";
$contraseña = "*****";

$base = "dudas";

$id = mysql_connect($servidor, $usuario, $contraseña);
$conexion = mysql_select_db ($base, $id) OR die ("No puedo conectar con
la base de datos \"$base\". ");
print ("Conexión establecida con éxito.\n<br><br>\n");

echo "\n<HTML>\n<HEAD>";
echo "<META HTTP-EQUIV='\"Content-Type\"'CONTENT='\"text/html;charset=ISO-8859-1\"'>";
echo "</HEAD>\n<BODY>\n<br><br>";

tabladudas_db ($id);

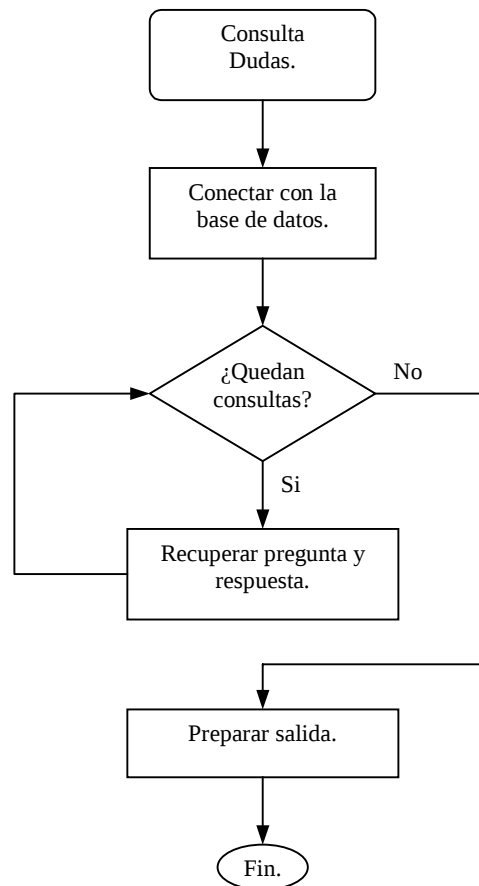
mysql_close ($id);
/* print ("Se ha cerrado con éxito."); */
?>
</BODY>
</HTML>
```

La segunda parte de la aplicación la constituye el programa que recibe el listado de preguntas que se quieren consultar. Dicho programa tendrá que acceder a la base de datos correspondiente y obtener las citadas preguntas junto a sus correspondientes



respuestas, preparar el formato de la página donde se mostrarán al cliente e incluirlas de forma adecuada.

Veamos con un diagrama de flujo la secuencia que sigue el programa:



A continuación se adjunta el código del programa:

```
<?php

/***** Programa principal *****/

/*****/
/* Estableciendo conexion con la base de datos. */
/*****/

$servidor = "localhost";
$usuario = "root";
$contraseña = "*****";
```



```
$base = "dudas";

Sid = mysql_connect($servidor, $usuario, $contraseña);
$conexion = mysql_select_db ($base, $id) OR die ("No puedo conectar con
la base de datos \"$base\". ");
/* print ("Conexión establecida con éxito.\n<br><br>\n"); */

echo "\n<HTML>\n<HEAD>";
echo "<META HTTP-EQUIV=\"Content-Type\"CONTENT=\"text/html;charset=ISO-8859-1\">";
echo "</HEAD>\n<BODY>\n<br><br>";

$cons = "SELECT id FROM dudas";
$res = mysql_query ($cons, $id) OR die ("Fallo en la respuesta.");
$sk = mysql_num_rows ($res);
for ($i=0; $i<$k; $i++){
    $name = "caja".$i;
    $kk=${$name};
    if (${ $name }){
        $cons = "SELECT respuesta FROM dudas WHERE pregunta='$kk'";
        $res = mysql_query ($cons, $id) OR die ("Fallo en la respuesta.");
        $valor = mysql_result($res,0,0);
        echo "${ $name }<br>";
        echo "$valor<br><br>";
    }
}

mysql_close ($id);
/* print ("Se ha cerrado con éxito."); */
?>

</BODY>
</HTML>
```

4.3 Implementación del servicio de encuesta de calidad.

Al igual que los anteriores, este servicio también fue descrito y especificado en el capítulo anterior. Ahora pasaremos a describir cómo lo hemos implementado.

Para la implementación de este servicio, se requiere la creación de una nueva base de datos que llamaremos “**encuesta**”. Dicha base de datos contendrá toda la información relativa a la encuesta de calidad que se quiere implementar. Para concretar, la información que se guardará en cada uno de los registros de la base de datos, será la correspondiente a realización concreta de la encuesta, es decir, se almacenará la puntuación que se ha dado a cada pregunta de la encuesta para una prueba concreta.



Los sucesivos registros de la base de datos serán las distintas realizaciones del cuestionario de calidad que realicen los alumnos.

Pasamos a describir la correspondiente base de datos:

Nombre: encuesta	Nombre de campo:	Tipo:
Tabla: bloque1		
	ID (primary key)	Numérico
	Valoracion	Numérico
	Pregunta1	Enum {1,2,3,4,5}
	Pregunta2	Enum {1,2,3,4,5}
		
	Pregunta7	Enum {1,2,3,4,5}
	Pregunta8	Enum {1,2,3,4,5}
Tabla bloque2		
	ID (primary key)	Numérico
	Valoracion	Numérico
	Pregunta9	Enum {1,2,3,4,5}
		
	Pregunta16	Enum {1,2,3,4,5}
Tabla: bloque3		
	ID (primary key)	Numérico
	Valoracion	Numérico
	Pregunta17	Enum {1,2,3,4,5}
		
	Pregunta24	Enum {1,2,3,4,5}
	Fecha	Fecha



Como se puede apreciar, la base de datos se ha dividido en tres tablas, cada una de las cuales se corresponde con cada uno de los bloques de ocho preguntas de que consta el cuestionario. Además, se incluye un campo “valoración”, para saber que valor da el alumno que contesta a cada bloque de preguntas; y también se incluye un campo “fecha” para tener una idea de cuando se contesto a dicha encuesta.

Cómo por la naturaleza del servicio, es una aplicación de uso restringido, ya que solo tienen acceso a contestar y rellenar la encuesta de calidad los alumnos que se encuentren matriculados en la asignatura, habrá que idear un método para controlar dicho acceso.

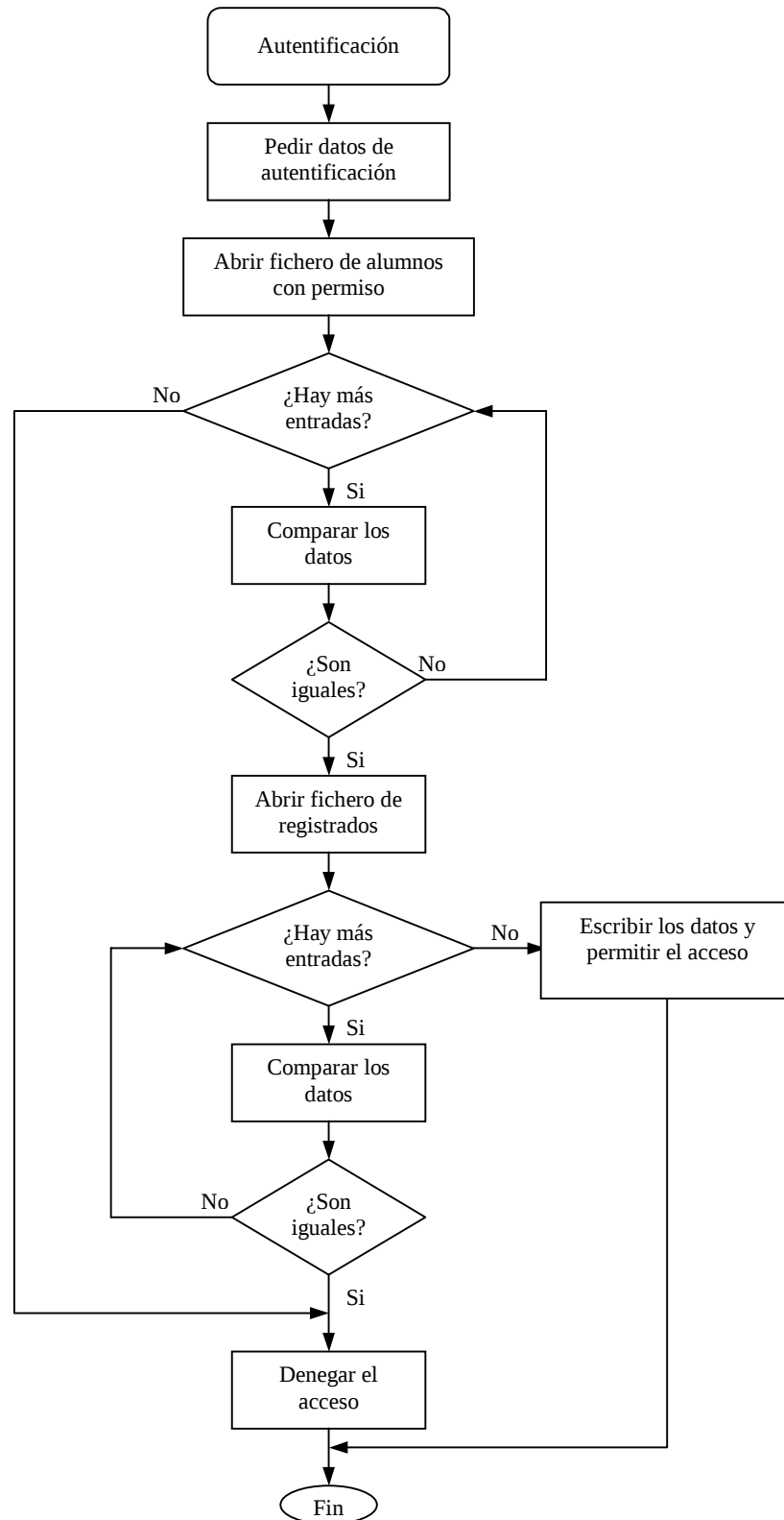
Para ello se ha optado por la siguiente solución. El profesor dispone de la información de todos los alumnos que se encuentran matriculados en la asignatura, por tanto esta información se volcará en un simple fichero de texto que tomaremos como referencia. En este fichero se encuentra D.N.I. del alumno, Apellidos, Nombre. A partir de esta información identificaremos si se trata de un alumno matriculado en la asignatura ó no lo es. Si no lo es, se le denegará el acceso, si lo es, habrá que revisar si dicho alumno ha contestado previamente a esta encuesta, porque de haberlo hecho anteriormente, ya no debe tener opción a volver a hacerla. Esto se va a implementar de esta forma para que un mismo alumno no tenga opción de contestar muchas encuestas de manera que se puedan enmascarar los resultados reales. Por tanto solo se permitirá una encuesta por alumno.

Para solucionar este problema lo que haremos es utilizar un nuevo fichero de texto, donde siguiendo el mismo formato que el anterior, iremos escribiendo en cada nueva línea, los datos del alumno que ha contestado a nuestra encuesta. De esta forma, con una segunda revisión de este nuevo fichero sabremos si el alumno que solicita contestar ya lo ha hecho antes, o por el contrario todavía no ha contestado. Una vez que conteste a la encuesta, habrá que incluirlo en este nuevo fichero.

Esta es la explicación del funcionamiento de la primera parte de la aplicación, un programa que será encargado de la validación del acceso al servicio. A continuación



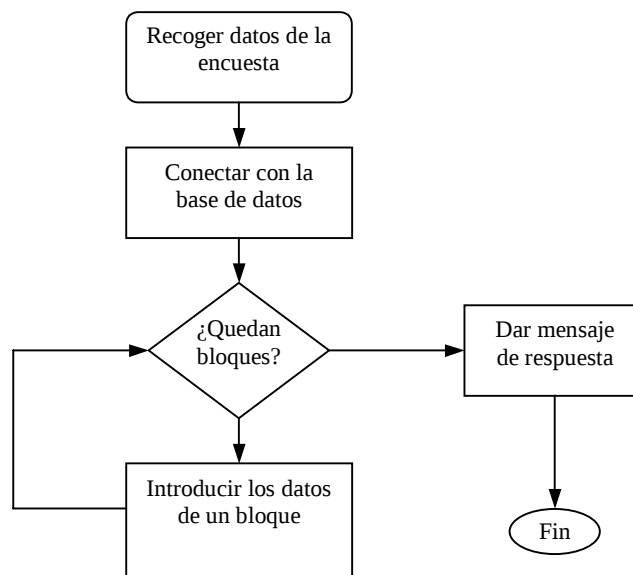
mostramos la secuencia que seguirá dicho programa basándonos en un diagrama de flujo:





Si después de esto se ha denegado el acceso, se muestra una pantalla informando de dicha negación. Si por el contrario se tiene acceso, se mostrará la página que contiene el cuestionario. Es cuestionario no pasa de ser más que un simple formulario puro de HTML.

Una vez que se tiene relleno el cuestionario, se envía para proceder a la captura de datos para un posterior análisis estadístico. El diagrama de flujo siguiente muestra el proceso que sigue la aplicación para almacenar los datos.



A continuación mostramos el código fuente de las dos partes de la aplicación:

```
<!doctype html public "-//w3c//dtd html 3.2//en">

<html>

<head>
<title>Acceso a encuesta.</title>
<meta name="GENERATOR" content="Arachnophilia 4.0">
<meta name="FORMATTER" content="Arachnophilia 4.0">
  <Script language="JavaScript">
<!--

function ApareceAyuda(){
  ayuda.style.visibility='visible';
  return true;
}

function DesapareceAyuda(){
  ayuda.style.visibility='hidden';
  return true;
```



```
}
//-->
</Script>
</head>

<body background="../../../images/fondo.jpg" text="#000000" link="#0000ff"
vlink="#ffffff" alink="#ff0000">

<table border="0" width="100%">
  <tr>
    <td valign="top">
      <p align="center">
      </td>
    <td> <b><font face="Arial" color="#000000"
size="4"><small>Complementos
de sistemas electrónicos digitales</small></font> </b></td>
  </tr>
</table>
<hr width="95%" align="center">
<p>

<h2 align="center">Encuesta de calidad</h2>

<table width="70%" align="center">
  <tr>
    <td>
      <b>Se recuerda al usuario que para poder acceder a la
realización de la encuesta debe estar matriculado en la asignatura de
"Complementos de Sistemas Electrónicos Digitales".
      </b>
    </td>
  </tr>
  <tr>
    <td>
      <b>A continuación se pedirán sus datos con el fin de
autenticarla, pero una vez que tenga acceso, no quedará registro
alguno de la persona que contesta a la encuesta, así que no se
preocupe y conteste con la mayor sinceridad que pueda.</b>
    </td>
  </tr>
  <tr>
    <td>
      <b>Muchas gracias.
      </b>
    </td>
  </tr>
</table>
<p>

<p align="center">

<input type="button" value="Acceder" onClick="ApareceAyuda( )">

<DIV Id="ayuda" STYLE="position:absolute;top:150px;left:150px;
width:400px;height:200px;text-align:justify;
background-color:#ccccff;visibility:hidden" >
```



Se va a proceder a su identificación como usuario con permiso para acceder a la encuesta.

<blockquote>

<p>

Por favor, rellene con sus datos:

[illegible]

</blockquote> </DIV>

</body>

</html>

```
<html>
<head>
<title>Encuesta.</title>
</head>
```

```
<body background="../images/fondo.jpg">
<table border="0" width="100%">
  <tr>
    <td valign="top" width="8%">
      <p align="center"> 
      </td>
    <td width="75%"> <font face="Arial" color="#000000" size="4">
      <b><small> Complementos de sistemas electrónicos
digitales</small></b></font></td>
    </tr>
```



```
</table>
<hr>

<?

$fp = fopen
("/usr/local/apache/htdocs/csed/encuesta/listadoalumnos.txt", "r");
$flag = 0;

while (!feof($fp)){
    $valor = fgets ($fp, 80);
    $dni2 = strtok($valor," ");
    $apellido2 = strtok(" ");

    /* print ("$dni, $apellido <br>"); */

    if ($dni == $dni2 && $apellido == $apellido2){
        $fp2 = fopen
("/usr/local/apache/htdocs/csed/encuesta/registro.txt", "r+");
        rewind($fp2);
        while (!feof($fp2)){
            $valor = fgets ($fp2, 80);
            $dni2 = strtok($valor," ");
            $apellido2 = strtok(" ");
            /* print("$dni2, $apellido2 <br>"); */
            if ($dni == $dni2){
                fclose ($fp2);
                fclose ($fp);
                die ("Lo sentimos, pero el usuario solo puede responder a
la encuesta una sola vez. Gracias.");
            }
        }
        fputs ($fp2, $dni." ");
        fputs ($fp2, $apellido."\n");
        $flag = 1;
        fclose ($fp2);
    }
}

if (feof($fp) && $flag == 0){
    fclose ($fp);
    die ("Lo sentimos, pero no es usted un usuario con permiso.
Gracias.");
}

?>

<h2 align="center">Encuesta de calidad.</h2>

<form action=tomadatos.php method=post >

<table border="1" width="95%" align="center" >
<tr>
<td> <td>Muy desacuerdo <td>Desacuerdo <td>Indiferente <td>De
acuerdo <td>Muy deacuerdo
<tr> <td width="60%"><b>ACTITUDES PERSONALES</b>
<td><td><td><td><td><input TYPE="text" align=rigth NAME="valor1"
SIZE="1" MAXLENGTH="1" VALUE="<? echo $valor1 ?>" >

<?

```





```
echo "<tr> <td>El profesor es educado y respetuoso con los alumnos ";
echo "<td><input type=\"radio\" name=\"pregunta1\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta1\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta1\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta1\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta1\" value=\"5\" >";

echo "<tr> <td>El profesor intenta motivar a los alumnos por la
asignatura ";
echo "<td><input type=\"radio\" name=\"pregunta2\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta2\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta2\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta2\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta2\" value=\"5\" >";

echo "<tr> <td>El profesor muestra interes en que los alumnos
comprendan las explicaciones ";
echo "<td><input type=\"radio\" name=\"pregunta3\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta3\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta3\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta3\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta3\" value=\"5\" >";

echo "<tr> <td>El profesor esta abierto a las sugerencias de los
alumnos ";
echo "<td><input type=\"radio\" name=\"pregunta4\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta4\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta4\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta4\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta4\" value=\"5\" >";

echo "<tr> <td>El profesor fomenta la participacion en clase ";
echo "<td><input type=\"radio\" name=\"pregunta5\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta5\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta5\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta5\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta5\" value=\"5\" >";

echo "<tr> <td>Las clases son amenas ";
echo "<td><input type=\"radio\" name=\"pregunta6\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta6\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta6\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta6\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta6\" value=\"5\" >";

echo "<tr> <td>El profesor cumple su horario de clase correctamente ";
echo "<td><input type=\"radio\" name=\"pregunta7\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta7\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta7\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta7\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta7\" value=\"5\" >";

echo "<tr> <td>El profesor atiende las tutorias correctamente ";
echo "<td><input type=\"radio\" name=\"pregunta8\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta8\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta8\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta8\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta8\" value=\"5\" >";
?>
```



```
<tr> <td width=\"60%\"><b>COMPETENCIA EXPOSITIVA</b>
<td><td><td><td><td><input TYPE=\"text\" align=right NAME=\"valor2\"
SIZE=\"1\" MAXLENGTH=\"1\" VALUE=\"<? echo $valor2 ?>\" >

<?
echo "<tr> <td>El profesor es claro en sus explicaciones ";
echo "<td><input type=\"radio\" name=\"pregunta9\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta9\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta9\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta9\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta9\" value=\"5\" >";

echo "<tr> <td>El profesor es ordenado en sus explicaciones ";
echo "<td><input type=\"radio\" name=\"pregunta10\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta10\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta10\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta10\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta10\" value=\"5\" >";

echo "<tr> <td>El ritmo que se sigue en las explicaciones se adecua al
ritmo de comprension del alumno ";
echo "<td><input type=\"radio\" name=\"pregunta11\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta11\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta11\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta11\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta11\" value=\"5\" >";

echo "<tr> <td>El profesor contesta debidamente a las dudas que se le
plantean ";
echo "<td><input type=\"radio\" name=\"pregunta12\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta12\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta12\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta12\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta12\" value=\"5\" >";

echo "<tr> <td>Las clases del profesor parecen estar bien preparadas
";
echo "<td><input type=\"radio\" name=\"pregunta13\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta13\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta13\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta13\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta13\" value=\"5\" >";

echo "<tr> <td>El profesor facilita la toma de apuntes ";
echo "<td><input type=\"radio\" name=\"pregunta14\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta14\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta14\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta14\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta14\" value=\"5\" >";

echo "<tr> <td>La asistencia a clase facilita la comprension del tema
";
echo "<td><input type=\"radio\" name=\"pregunta15\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta15\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta15\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta15\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta15\" value=\"5\" >";
```



```
echo "<tr> <td>El profesor utiliza con frecuencia ejemplos extraidos
de la realidad para la mejor comprension de los conceptos ";
echo "<td><input type=\"radio\" name=\"pregunta16\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta16\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta16\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta16\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta16\" value=\"5\" >";
?>

<tr> <td width=\"60%\"><b>ASPECTOS OBJETIVOS DE PREPARACION</b>
<td><td><td><td><td><input TYPE="text" align=rigth NAME="valor3"
SIZE="1" MAXLENGTH="1" VALUE="<? echo $valor3 ?>" >

<?
echo "<tr> <td>Durante el curso realizamos el numero suficiente de
casos practicos ";
echo "<td><input type=\"radio\" name=\"pregunta17\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta17\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta17\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta17\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta17\" value=\"5\" >";

echo "<tr> <td>Se dedica a la resolucio de cada caso practico el
tiempo que efectivamente se requiere ";
echo "<td><input type=\"radio\" name=\"pregunta18\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta18\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta18\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta18\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta18\" value=\"5\" >";

echo "<tr> <td>Durante el curso realizamos el numero suficiente de
problemas ";
echo "<td><input type=\"radio\" name=\"pregunta19\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta19\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta19\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta19\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta19\" value=\"5\" >";

echo "<tr> <td>Se dedica a la resolucio de problemas el tiempo que
efectivamente se requiere ";
echo "<td><input type=\"radio\" name=\"pregunta20\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta20\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta20\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta20\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta20\" value=\"5\" >";

echo "<tr> <td>La amplitud del temario de la asignatura es acorde con
la duracion del curso ";
echo "<td><input type=\"radio\" name=\"pregunta21\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta21\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta21\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta21\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta21\" value=\"5\" >";

echo "<tr> <td>El profesor utiliza correctamente los medios
audiovisuales disponibles como herramientas de apoyo a la docencia ";
echo "<td><input type=\"radio\" name=\"pregunta22\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta22\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta22\" value=\"3\" >";
```



```
echo "<td><input type=\"radio\" name=\"pregunta22\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta22\" value=\"5\" >";

echo "<tr> <td>La bibliografia recomendada es accesible y comprensible
";
echo "<td><input type=\"radio\" name=\"pregunta23\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta23\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta23\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta23\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta23\" value=\"5\" >";

echo "<tr> <td>Los exámenes se ajustan a los contenidos explicados
durante el curso ";
echo "<td><input type=\"radio\" name=\"pregunta24\" value=\"1\" >";
echo "<td><input type=\"radio\" name=\"pregunta24\" value=\"2\" >";
echo "<td><input type=\"radio\" name=\"pregunta24\" value=\"3\" >";
echo "<td><input type=\"radio\" name=\"pregunta24\" value=\"4\" >";
echo "<td><input type=\"radio\" name=\"pregunta24\" value=\"5\" >";

echo "</table>";
echo "<p><input type=submit value=Enviar> ";
echo "<input type=reset value=Borrar></p>";
echo "</form>";
?>
```

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

```
<html>
<head>
<title>Encuesta.</title>
<style>
BODY
{
scrollbar-3d-light-color: #FF9933; scrollbar-arrow-color: Black;
scrollbar-base-color: Black; scrollbar-dark-shadow-color: #367CC2;
scrollbar-face-color: #5894D0; scrollbar-highlight-color:
ThreedLightShadow; scrollbar-shadow-color: InactiveCaption
}
A:link {
COLOR: black; FONT-FAMILY: times new roman; FONT-SIZE: 11pt; TEXT-
DECORATION: none
}
TD {
COLOR: black; FONT-FAMILY: times new roman; FONT-SIZE: 11pt
}
A {
COLOR: black; FONT-FAMILY: times new roman; FONT-SIZE: 11pt
}
A:hover {
BACKGROUND-COLOR: #99ccff; COLOR: blue; FONT-FAMILY: times new roman;
TEXT-DECORATION: none
}
A:active {
COLOR: #0E0B76; FONT-FAMILY: times new roman; TEXT-DECORATION: none
}
A:visited {
```



```
FONT-FAMILY: times new roman; TEXT-DECORATION: none
}
</style>
</head>

<body background="../images/fondo.jpg">
<table border="0" width="100%">
  <tr>
    <td valign="top" width="8%">
      <p align="center"> 
    </td>
    <td width="75%"> <font face="Arial" color="#006666" size="4">
      <b><small> Complementos de sistemas electrónicos
digitales</small></b></font></td>
  </tr>
</table>
<hr>

<?

/*****/
/* PROGRAMA> tomadatos.php */
/*****/

/*****/
/* Conectamos con la base de datos */
/*****/

$servidor = "localhost";
$usuario = "root";
$contrasena = "j0nj0n";

$base = "encuesta";

$f = mysql_connect ($sevidor, $usuario, $contrasena);
$conexion = mysql_select_db ($base, $f) or die ("No puedo conectar con
la base de datos \"$base\". ");

/* print ("Conexion establecida con exito. \n <br><br> \n"); */

/* Creamos la sentencia sql */

$sql = "INSERT INTO bloque1 VALUES ('', '$valor1', '$pregunta1',
'$pregunta2', '$pregunta3', '$pregunta4', '$pregunta5', '$pregunta6',
'$pregunta7', '$pregunta8')";
$res = mysql_query ($sql, $f) or die ("Fallo en la toma de datos 1.");

/* Creamos la sentencia sql */

$sql = "INSERT INTO bloque2 VALUES ('', '$valor2', '$pregunta9',
'$pregunta10', '$pregunta11', '$pregunta12', '$pregunta13',
'$pregunta14',
'$pregunta15', '$pregunta16')";
$res = mysql_query ($sql, $f) or die ("Fallo en la toma de datos 2.");

/* Creamos la sentencia sql */

$fecha = date("Y-m-d");
```



```
$sql = "INSERT INTO bloque3 VALUES ('', '$valor3', '$pregunta17',  
'$pregunta18', '$pregunta19', '$pregunta20', '$pregunta21',  
'$pregunta22',  
'$pregunta23', '$pregunta24', '$fecha')";  
$res = mysql_query ($sql, $f) or die ("Fallo en la toma de datos 3.");  
  
/* print ("Se ha concluido la toma de datos."); */  
mysql_close ($f);  
  
?>  
  
<p>  
  
<center><h1>Gracias por su colaboración.</h1></center>  
  
<p>  
  
<table width="70%" align="center">  
  <tr>  
    <td>  
      <b> Los datos que usted nos ha proporcionado han sido  
convenientemente almacenados para un posterior tratamiento estadístico  
sin que haya quedado constancia de su identificación. Muchas  
gracias.</b>  
    </td>  
  </tr>  
</table><p>  
  
<p>  
  
<p align="center">  
  <a href="../index.html"> <b>Volver al inicio</b></a>  
</body>  
  
</html>
```

5. Programación en el lado cliente.

En este apartado echaremos un vistazo a un lenguaje de programación que hemos utilizado para mejorar mucho la calidad de nuestro servidor Web. Nos estamos refiriendo al lenguaje JavaScript. Dicho lenguaje, desarrollado por Netscape, mejora los formularios HTML, genera código HTML “sobre la marcha” según se carga un documento y proporciona mecanismos para controlar el modo en que se presentan los documentos Web al usuario.



Por otra parte también teníamos la opción de usar Java, que desarrollado originalmente por Sun Microsystems, es un lenguaje mucho más robusto, permite que una página Web dibuje gráficos (no sólo mostrar imágenes), que proporcione una interfaz gráfica de usuario, que haga nuevas conexiones a la red, y hacer casi cualquier otra cosa imaginable. De hecho, Java fue diseñado originalmente para otros propósitos a parte de las comunicaciones Web.

Uno de los principales beneficios que proporcionan tanto Java como JavaScript es una funcionalidad adicional del lado del navegador en lugar del lado del servidor. Al dar al navegador más capacidades, el servidor Web no se sobrecarga con programas lentos.

JavaScript y Java se complementan muy bien. JavaScript maneja la presentación de documentos Web y les proporciona capacidades extra mientras que Java permite a los usuarios recibir programas sofisticados en un navegador para ser ejecutados inmediatamente.

El código JavaScript se integra directamente en el código HTML. De hecho, literalmente extiende la sintaxis HTML, especialmente para formularios.

Algunos de los usos más comunes de JavaScript en las páginas Web son:

- Dirigir conexiones a ventanas y marcos en particular.
- Crear funciones de validación para verificar la entrada de datos en campos de formulario.
- Generar código HTML mientras se carga una página.
- Modificar la conducta de campos de formulario, como, por ejemplo, moverse a una página en particular cuando un usuario hace clic en un botón.

JavaScript es un lenguaje de programación orientado a objetos. Separa los elementos de un documento HTML en unidades distintas, llamadas objetos. Los objetos



en un documento HTML incluyen la entrada en un formulario, un formulario en sí mismo, un marco, o una imagen. De hecho, un documento HTML en sí mismo es también un objeto.

El tema de la programación orientada a objetos puede ser muy complicada, pero para JavaScript, un objeto es simplemente una entidad que tiene propiedades que la definen (como variables) y métodos para modificarlo.

Aunque a lo largo de todo el trabajo en muchas de las páginas creadas se pueda hacer uso de alguna que otra función de JavaScript, a continuación nos centraremos en explicar los bloques de código que a nuestro parecer son más interesante, desde el punto de vista funcional y por el efecto visual que se consigue.

5.1 Implementación del módulo explorador de temario.

El módulo explorador de temario fue descrito en el capítulo anterior, pero vamos a volver a recordar brevemente cual era su función.

Se ha ideado un sistema dinámico de carpetas que se abren y cierran a petición del usuario, sistema similar al explorador de Windows, y que muestran de una forma muy organizada y esquemática todo el temario.

El funcionamiento del módulo no es muy complicado. Cuando pulsamos con el ratón sobre una carpeta, el sistema identifica que es una entrada madre de otras subcarpetas o por el contrario no lo es. En caso afirmativo, hace un cambio de icono mostrando la carpeta abierta y despliega todas las entradas hijas con su conveniente indexación.

Para conseguir esta funcionalidad ha sido necesario la implementación de varias funciones de JavaScript que dividan el trabajo a realizar en cada momento. A continuación se muestra este código:



```
<SCRIPT LANGUAGE="JavaScript">

// **funciones que establecen objetos y entorno**
// constructor del array basico
function makeArray(n) {
    this.length = n
    return this
}

// constructor de objeto para cada entrada
// (see object-building calls, below)
function dbRecord(mother,display,URL,indent){
    this.mother = mother // is this item a parent?
    this.display = display // text to display
    this.URL = URL // link tied to text
    this.indent = indent // how deeply nested?
    return this
}

// create object containing outline content and attributes
// To adapt outline for your use, modify this table. Make sure
// that the size of the array (db[i]) is reflected in the call
// to makeArray(i). See the dbRecord() function, above, for the
// meaning of the four parameters in each array entry.
var db = new makeArray(39)
db[1] = new dbRecord(true, "TEMA 1. INTRODUCCIÓN A LOS PROCESADORES DIGITALES DE
SEÑAL (DSPs). CONCEPTOS BÁSICOS.", "tema1/tema1.html", 0)
db[2] = new dbRecord(false, "1 Introducción e historia.", "tema1/punto1.html", 1)
db[3] = new dbRecord(false, "2 Principales aplicaciones de los DSPs.", "tema1/punto2.html", 1)
db[4] = new dbRecord(false, "3 Estructura interna basica de un DSP.", "tema1/punto3.html", 1)
db[5] = new dbRecord(false, "4 Principales fabricantes de DSPs.", "tema1/punto4.html", 1)
db[6] = new dbRecord(true, "TEMA 2. FAMILIA DE DSPs TMS320C3x DE TEXAS INSTRUMENTS.
CONCEPTOS BÁSICOS DEL HARDWARE.", "tema2/tema2.html", 0)
db[7] = new dbRecord(false, "1 Introduccion.", "tema2/punto1.html", 1)
db[8] = new dbRecord(false, "2 Descripcion general del sistema.", "tema2/punto2-1parte.html", 1)
db[9] = new dbRecord(true, "3 Unidad central de control de procesos (CPU).", "tema2/punto3.html", 1)
db[10] = new dbRecord(false, "3.1 Registros internos de la CPU.", "", 2)
db[11] = new dbRecord(false, "4 Operaciones internas del bus.", "", 1)
db[12] = new dbRecord(true, "5 Organizacion de la memoria.", "", 1)
db[13] = new dbRecord(false, "5.1 Reset y Vectores de Interrupcion.", "", 2)
db[14] = new dbRecord(false, "5.2 Periferico Memoria Cache.", "", 2)
db[15] = new dbRecord(false, "5.3 Arranque programado en el TMS320C31: Boot-Loader.", "", 2)
db[16] = new dbRecord(true, "6 Operaciones externas del bus.", "", 1)
db[17] = new dbRecord(false, "6.1 Estructura Master-Slave en un sistema microprocesador.", "", 2)
db[18] = new dbRecord(false, "6.2 Diagramas de tiempos y estados de espera.", "", 2)
db[19] = new dbRecord(false, "6.3 Tiempo de ejecucion y tiempo de acceso.", "", 2)
db[20] = new dbRecord(false, "6.4 Peculiaridades en el acceso a perifericos.", "", 2)
db[21] = new dbRecord(true, "TEMA 3. FAMILIA TMS320C3x. DESCRIPCION DEL
SOFTWARE.", "tema3/tema3.html", 0)
db[22] = new dbRecord(false, "1 Introducción.", "tema3/punto1.html", 1)
db[23] = new dbRecord(false, "2 Formato de datos.", "tema3/punto2.htm", 1)
db[24] = new dbRecord(false, "3 Modos de direccionamiento.", "tema3/punto3.html", 1)
db[25] = new dbRecord(false, "4 Pila.", "tema3/punto4.html", 1)
db[26] = new dbRecord(false, "5 Sumario de instrucciones.", "tema3/punto5.htm", 1)
db[27] = new dbRecord(true, "TEMA 4. FAMILIA TMS320C3x: PERIFERICOS
INTERNOS.", "tema4/tema4.html", 0)
db[28] = new dbRecord(false, "1 Introducción.", "tema4/punto1.html", 1)
db[29] = new dbRecord(false, "2 Entradas-salidas digitales: XF0, XF1.", "tema4/punto2.html", 1)
db[30] = new dbRecord(true, "3 Temporizadores.", "tema4/punto3.html", 1)
```



```
db[31] = new dbRecord(false, "3.1 Registros de control y configuración.", "tema4/punto3-1.html", 2)
db[32] = new dbRecord(false, "3.2 Estructura interna y configuración del periférico temporizador.", "tema4/punto3-2.html", 2)
db[33] = new dbRecord(true, "4 Puertos serie sincronos.", "tema4/punto4.html", 1)
db[34] = new dbRecord(false, "4.1 Registros de control y configuración.", "tema4/punto4-1.html", 2)
db[35] = new dbRecord(false, "4.2 Estructura interna y configuración del periférico puerto serie.", "tema4/punto4-2.html", 2)
db[36] = new dbRecord(false, "4.3 Modos de operación: Diagramas de tiempo.", "tema4/punto4-3.html", 2)
db[37] = new dbRecord(true, "5 Controlador DMA.", "tema4/punto5.html", 1)
db[38] = new dbRecord(false, "5.1 Registros de control y configuración.", "tema4/punto5-1.html", 2)
db[39] = new dbRecord(false, "5.2 Modos de operación: Sincronización de los eventos DMA.", "tema4/punto5-2.html", 2)
```

```
// ** functions that get and set persistent cookie data **
// set cookie data
function setCurrState(setting) {
    document.cookie = "currState=" + escape(setting)
}

// retrieve cookie data
function getCurrState() {
    var label = "currState="
    var labelLen = label.length
    var cLen = document.cookie.length
    var i = 0
    while (i < cLen) {
        var j = i + labelLen
        if (document.cookie.substring(i, j) == label) {
            var cEnd = document.cookie.indexOf(";", j)
            if (cEnd == -1) {
                cEnd = document.cookie.length
            }
            return unescape(document.cookie.substring(j, cEnd))
        }
    }
    return ""
}

// **function that updates persistent storage of state**
// toggles an outline mother entry, storing new value in the cookie
function toggle(n) {
    if (n != 0) {
        var newString = ""
        var currState = getCurrState() // of whole outline
        var expanded = currState.substring(n-1, n) // of clicked item
        newString += currState.substring(0, n-1)
        newString += expanded ^ 1 // Bitwise XOR clicked item
        newString += currState.substring(n, currState.length)
        setCurrState(newString) // write new state back to cookie
    }
}
```

```
// **functions used in assembling updated outline**
// returns the proper GIF file name for each entry's control
function getGIF(n) {
    var mom = db[n].mother // is entry a parent?
    var expanded = getCurrState().substring(n-1, n) // of clicked item
```



```
        if (!mom) {
            return "images/image1.gif"
        } else {
            if (expanded == 1) {
                return "images/folder-1.gif"
            }
        }
        return "images/folder-0.gif"
    }
}

// returns the proper status line text based on the icon style
function getGIFStatus(n) {
    var mom = db[n].mother // is entry a parent
    var expanded = getCurrState().substring(n-1,n) // of rolled item
    if (!mom) {
        return "No hay más entradas."
    } else {
        if (expanded == 1) {
            return "Pulsa para cerrar la carpeta."
        }
    }
    return "Pulsa para abrir la carpeta."
}

// returns padded spaces (in multiples of 3) for indenting
function pad(n) {
    var result = ""
    for (var i = 1; i <= n; i++) {
        result += "   "
    }
    return result
}

// initialize 'current state' storage field
if (getCurrState() == "" || getCurrState().length != db.length) {
    initState = ""
    for (i = 1; i <= db.length; i++) {
        initState += "0"
    }
    setCurrState(initState)
}

// see if user is running a Mac browser for special case handling
function isMac() {
    return (navigator.userAgent.indexOf("Macintosh") >= 0) ? true : false
}
// end -->
</SCRIPT>
... ..
<SCRIPT LANGUAGE="JavaScript">
<!-- start
// build new outline based on the values of the cookie
// and data points in the outline data array.
// This fires each time the user clicks on a control,
// because the HREF for each one reloads the current document.
var prevIndentDisplayed = 0
var showMyDaughter = 0

var newOutline = "<PRE><H4>" // let padded spaces make indents
```



```

// cycle through each entry in the outline array
for (var i = 1; i <= db.length; i++) {
    var theGIF = getGIF(i) // get the image
    var theGIFStatus = getGIFStatus(i) // get the status message
    var currIndent = db[i].indent // get the indent level
    var expanded = getCurrState().substring(i-1,i) // current state
    // display entry only if it meets one of three criteria
    if (currIndent == 0 || currIndent <= prevIndentDisplayed || (showMyDaughter == 1 &&
(currIndent - prevIndentDisplayed == 1))) {
        newOutline += pad(currIndent)
        if (isMac()) {
            newOutline += "<A HREF=# onMouseOver=\"window.parent.status=\" +
theGIFStatus + \"\";return true;\" onClick=\"toggle(\" + i + \");var timeoutID =
setTimeout('history.go(0)',300)\"><IMG SRC=\"" + theGIF + "\" BORDER=0></A>"
        } else {
            newOutline += "<A HREF=\"javascript:history.go(0)\"
onMouseOver=\"window.parent.status=\" + theGIFStatus + \"\";return true;\" onClick=\"toggle(\" + i +
\")\"><IMG SRC=\"" + theGIF + "\" BORDER=0></A>"
        }
        newOutline += " <A HREF=\"" + db[i].URL + "\"
onMouseOver=\"window.parent.status='Acceder a: " + db[i].display + "\";return true;\"><font
face='Arial' color='#000099'><small>" + db[i].display + "</small></font></A><BR>"
        prevIndentDisplayed = currIndent
        showMyDaughter = expanded
    }
}
newOutline += "</H4></PRE>"
document.write(newOutline)

// end -->
</SCRIPT>

```

5.2 Implementación del módulo sistema de ayuda.

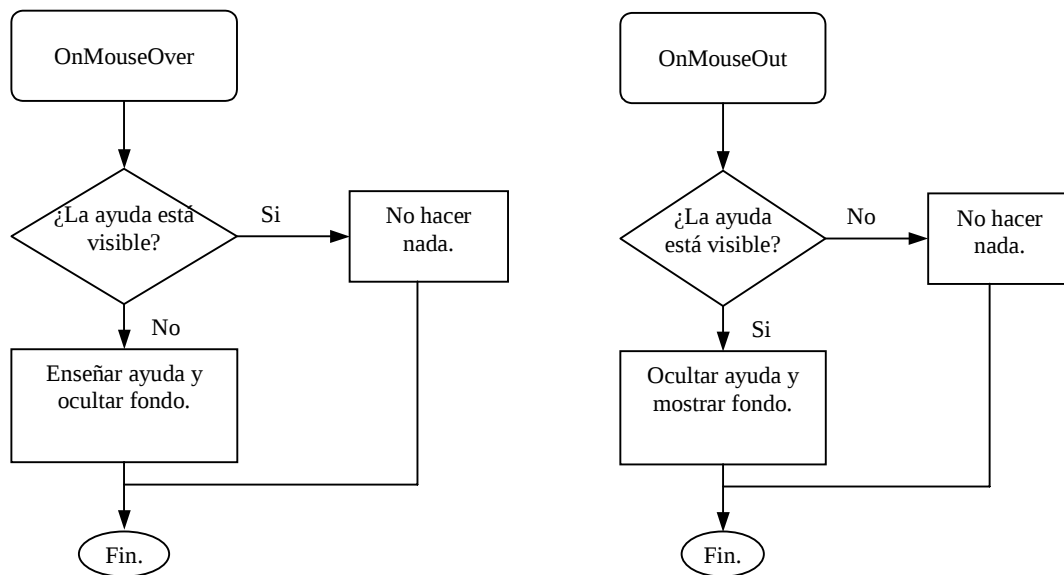
Al igual que el anterior, el módulo funcional del sistema de ayuda fue descrito con anterioridad en el capítulo 2. No obstante, volveremos a recordar de forma muy general cual era su cometido.

El objetivo fundamental del sistema de ayuda es presentar distintas ventanas que muestren información adicional para que sirva de ayuda a la navegación. Estas ventanas aparecerán a petición del usuario, situando el ratón sobre unos iconos, y acto seguido aparecerá una zona nueva en la pantalla, presentando la información relacionada a lo



que se puede encontrar si toma ese itinerario de navegación. La información presenta un adelanto en modo descriptivo de lo que el usuario se encontrará en las páginas que corresponden a ese acceso.

La secuencia de ejecución es la misma para todas y cada una de la peticiones de ayuda, y se muestra claramente en los siguientes diagramas de flujo:



A continuación mostramos el código JavaScript que implementa esta funcionalidad:

```
<Script language="JavaScript">
<!--

function ApareceAyuda(){
    ayuda.style.visibility='visible';
    return true;
}

function CambiaAyuda(){
    if(ayuda.style.visibility=='visible'){
        ayuda.style.visibility='hidden';
    }
    else{
        ayuda.style.visibility='visible';
    }
    return true;
}
```



```
function DesapareceAyuda(){
    ayuda.style.visibility='hidden';
    return true;
}

//-->
</Script>

... ..

    <td ><input type="image" src="images/esivol.gif" width="31" height="31"
onMouseOver="ApareceAyuda( )" onMouseOut="DesapareceAyuda( )" ></td>
    <td > <a href="info.html"> <b> Información general de
    la asignatura. </b> </a> </td>

... ..

<DIV Id="ayuda" STYLE="position:absolute;top:205px;left:305px;
    width:440px;height:300px;text-align:justify;
    background-color:#cccc00;visibility:hidden" >
    <Center><B><U>Ayuda</U></B></Center><BR>
    Este texto sería el correspondiente a la siguiente ayuda, la ayuda2 y aparece cuando el usuario lo
    solicita
</DIV>
```

5.3 Implementación del módulo de control de animación.

El objetivo fundamental de este módulo es, como su propio nombre indica, el poder controlar de alguna forma el mostrado de una animación.

A lo largo de toda la exposición hemos aludido ya un par de veces, al hecho de que una de las características importantes que se querían conseguir era modelar mediante algún tipo de animación, el funcionamiento concreto de algún aspecto estudiado, con objeto de facilitar su comprensión y hacerla lo más intuitiva posible.

Dicho trabajo exige, que además de crear la propia animación, se pueda disponer de mecanismos que permitan controlar su visualización, para así proporcionar que se pueda seguir una correcta exposición secuencial paso a paso, sobre el funcionamiento que se está modelando.

En nuestro caso concreto, describiremos y analizaremos el modelado de la estructura *pipeline*, por ser el más representativo.



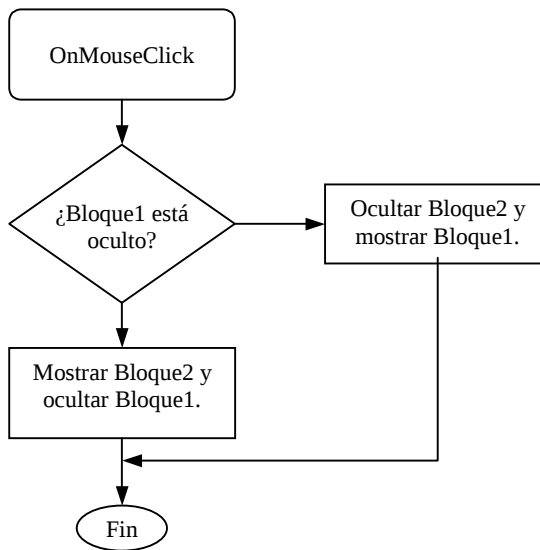
La estructura pipeline en la ejecución de instrucciones, consiste en la ejecución en paralelo de los niveles Fetch, Decode, Read y Execute de varias instrucciones básicas, y si es posible, en el mismo intervalo de tiempo. Esto se esquematiza en la tabla siguiente:

Ciclo	Fetch	Decode	Read	Execute
$k-3$	w	-	-	-
$k-2$	x	w	-	-
$k-1$	y	x	w	-
k	z	y	x	w
$k+1$	-	z	y	x
$k+2$	-	-	z	y
$k+3$	-	-	-	z

Solapamiento perfecto
de instrucciones
(*Overlapping*)

Para modelar este funcionamiento, hemos optado por la realización de dos bloques animados y presentar un modelo en dos pasos. El primer bloque modela la entrada de cuatro instrucciones distintas, representadas por bloques de colores distintos, y modela como van pasando poco a poco por cada uno de los cuatro niveles de ejecución de microinstrucciones de la CPU (modelada mediante un cerebro). Este bloque termina precisamente con el *Overlapping* o solapamiento perfecto, cuando los cuatro niveles están ocupados con las cuatro instrucciones al mismo tiempo. El segundo bloque simplemente modela como se termina la ejecución de cada instrucción pasando por el resto de estados y liberando completamente a la CPU (esto se modela viendo que las instrucciones desaparecen y el cerebro queda como al principio).

Desde el punto de vista técnico, para controlar esta animación en dos pasos, requerimos un proceso que sea capaz de intercambiar los dos bloques de animación en el momento que se pulse un botón por parte del usuario. El programa JavaScript que controla el funcionamiento seguirá el siguiente diagrama de flujo:



A continuación mostramos el código JavaScript que implementa esta funcionalidad:

```
<script language="JavaScript">
<!--

function CambiaMostrar(){

    if(prueba1.style.visibility=='hidden'){
        prueba2.style.visibility='hidden';
        prueba1.style.visibility='visible';
        Boton.value="Paso siguiente";
    }
    else if(prueba1.style.visibility=='visible'){
        prueba2.style.visibility='visible';
        prueba1.style.visibility='hidden';
        Boton.value="Ver animación";
    }

    return true;
}

//-->

</script>

... ....

    <p ALIGN="center">
    <p ALIGN="center"><input type="button" name="Boton" value="Ver animación"
onClick="CambiaMostrar( )">

... ..
```



```
<DIV Id="prueba1" STYLE="position:absolute;top:291px;left:191px;
visibility:hidden">
  <p> </p>
</DIV>

<DIV Id="prueba2" STYLE="position:absolute;top:291px;left:191px;
visibility:hidden" >
  <p> </p>
</DIV>

... ..
```

6. Consideraciones de seguridad.

La seguridad de un servidor Web puede dividirse en dos categorías, seguridad del servidor y seguridad general. Para la seguridad del servidor, vamos a echar un vistazo a algunos de los puntos vulnerables más comunes de la seguridad Web y a tratar algunos consejos de configuración que se deberían considerar.

6.1 Seguridad del servidor Web.

Los servidores Web de Internet se crean porque una compañía, universidad, o cualquier otro tipo de organización tiene información útil que quiere hacer disponible para otros, en nuestro caso el Departamento de Tecnología Electrónica de la Universidad de Sevilla. Un usuario malicioso podría intentar modificar páginas Web o ver páginas que no son para un público general. El usuario puede que no esté interesado en el servidor Web sino que lo que quiera sea usarlo como base de lanzamiento para alcanzar otros ordenadores dentro de la organización.

Es importante tener un servidor Web que esté configurado apropiadamente. Pero es igual de importante comprender qué otros servicios de Internet está ofreciendo su ordenador porque esos servicios pueden ser incluso más fáciles de comprometer.

Consejos para una configuración segura del servidor.



- Ejecute el servidor sin privilegios de root. Siempre debe asumir que existe algún potencial de ataque, o por un fallo de configuración o por un punto vulnerable en el mismo servidor, por lo que siempre será mejor ejecutar el servidor como un usuario con privilegios limitados.
- Use un firewall para Intranet si su servidor se va a usar sólo por un dominio seleccionado de usuarios.
- Instale parches. De tiempo en tiempo, se descubren nuevos puntos vulnerables en servidores Web. Es importante que los trate tan pronto como sea posible.

Consejos para un árbol de documentos seguro.

- Superponga servidores FTP y Web con cuidado. La mayor parte de servidores de FTP tienen un área para dejar ficheros, y algunos servidores Web (como Apache) pueden configurarse para ejecutar cualquier programa con una extensión en particular, como por ejemplo .php. Esta configuración podría permitir potencialmente a un usuario dejar un fichero a través de FTP anónimo y ejecutarlo a través del servidor Web.
- Diseñe los Scripts PHP cuidadosamente.
- Mantenga ocultos a los intérpretes y los shells.
- Centralice los mecanismos de control de acceso.

6.2 Seguridad general.

Aunque su servidor Web pueda ser seguro, podemos encontrar otros caminos que comprometan la integridad del sistema.

Los sistemas operativos actuales, particularmente UNIX, proporcionan una gran cantidad de servicios de Internet aparte de servidores Web, que incluyen Telnet, rlogin,



mail, ftp, y numerosos otros. Cada uno de estos servicios proporciona a otros sistemas de Internet información sobre su servidor, junto con una avenida por la que colarse.

La mejor aproximación para ofrecer un servidor Web público es usar un sistema “sacrificable”, uno que proporciona los menos servicios posibles y que no se usa para otros propósitos. Ciertamente evite usar un único sistema para servidor Web y como sistema de almacenamiento de ficheros privados de usuario.

Apague tantos servicios como sea posible. Si las páginas Web siempre se actualizan desde la consola, apague los servicios Telnet y rlogin. Otros como Finger tampoco son necesarios generalmente.

En general, cuantos más servicios se encuentren apagados, más seguro es su sistema.