

Práctica 2 – Adaptación de la web

ÍNDICE

1. Contenido de la práctica.

1.1. Ficheros de la práctica.

1.2. Objetivos alcanzados del guión.

2. Especificaciones.

2.1. PHP.

2.2. Formularios.

2.2.1. Usuarios.

2.2.2. Items.

2.3. MySQL, tablas.

2.4. JavaScript.

1. Contenido de la práctica.

1.1. Ficheros de la práctica.

La práctica contiene los mismos ficheros que la práctica evaluable I, habiéndose sustituido los ficheros con extensión ‘.html’ por ficheros con extensión .php por lo general. Contiene además los respectivos ficheros ‘.css’ con el diseño y ciertos archivos con extensión ‘.inc’ que incluyen fragmentos de código comunes a la mayoría de ficheros con formularios, los cuales se concretarán más adelante.

Por otra parte, la estructura y la conexión entre los ficheros se mantiene, incluyendo en dicha conexión los nuevos ficheros que hayan sido necesarios. Todos los ficheros se encuentran en el directorio ‘pe2’ de Betatun a excepción de las imágenes, las cuales se encuentran en un subdirectorío del anterior denominado ‘Imágenes’ tal y como en la práctica anterior se indicó.

Por último, en el mismo directorio que el resto de ficheros de la práctica existe un PDF, denominado ‘como_se_hizo.pdf’, que se corresponde con la memoria de la práctica.

1.2. Objetivos alcanzados por el gui3n.

La práctica cumple casi con totalidad todos los objetivos planteados por el gui3n. Es posible darse de alta como usuario, registrándose así en la base de datos de la web, así como iniciar sesi3n (si se ha dado de alta anteriormente) y mantener la misma hasta que se desee.

Como usuario adem3s, se permite la modificaci3n de todos los datos personales del perfil desde el nombre de usuario hasta la contraseña o el correo electr3nico si así se desea.

Si se es administrador (usuario dado ya de alta en la base de datos bajo el nombre de usuario ‘Admin’), se podr3 crear, modificar y eliminar secciones de la base de datos actualiz3ndose la barra de navegaci3n de las secciones cuando se realice alg3n cambio. Adem3s, se podr3n aadir, eliminar y modificar ítems (desde ‘Admin’). Es posible adem3s la eliminaci3n de usuarios desde la administraci3n. Adem3s, desde el ámbito de usuario no se podr3 acceder a la administraci3n.

Por otra parte, todos los formularios est3n procesados por su propio fichero ‘.php’, los cuales se especificar3n m3s adelante y validados por un script realizado en JavaScript tal y como se indica (uno por formulario concretamente).

Si en alg3n ítem se pasa el rat3n por encima de su imagen, aparecer3 un texto emergente que mostrar3 el nombre y la secci3n de dicho ítem.

Sin embargo, no es posible aadir una imagen a un ítem.

2. Especificaciones

2.2. PHP.

En éste apartado se destacarán los fragmentos escritos en lenguaje PHP incrustados en ciertas partes del código, ya que los formularios se especificarán posteriormente.

A lo largo de los ficheros ‘.php’ que contiene la práctica se pueden encontrar ciertos fragmentos de código en PHP que sirven como ayuda de cara al empleo tanto de los formularios como de otro tipo de funcionalidades.

En todos los ficheros en los que se puede iniciar sesión o en los que se debe haber iniciado sesión con anterioridad se encuentra un fragmento PHP que contiene una serie de condicionales cuya funcionalidad es la siguiente: Si no hay una sesión iniciada, se mostrará el formulario para iniciar sesión así como el enlace a la pestaña de registro de usuarios para darse de alta. En caso contrario, si el usuario de la sesión actual no es administrador, se mostrará el nombre del usuario, así como un enlace para cerrar sesión y otro para editar el perfil. Si el usuario es administrador, se mostrará un enlace para cerrar sesión y otro para acceder a la pestaña de administración. De ésta forma, solo el usuario ‘Admin’ podrá acceder a dicha sección de la web.

```
<?php
if(!empty($_SESSION['Usuario'])){
    echo '<h2 id="u"> ' . $_SESSION['Usuario'] . ' </h2>';
    <a class="enlace" href="procesar_logout.php" id="u2"> Cerrar Sesión &nbsp;   </a>';

    if($_SESSION['Usuario']==$Admin){
        echo '<a class="enlace" href="administracion.php" id="u3"> Administración </a>';
    }else{
        echo '<a class="enlace" href="modificarperfil.php" id="u3"> Editar perfil </a>';
    }
}else{
    echo '<form class="iniciar_sesion" name="iniciar_sesion" action="procesar_login.php" onsubmit="return validarFormulario()" method="post">
    <fieldset> <!-- Inicacion de sesión / Registro de usuarios -->
    <label for="usr"><b> Usuario: </b></label>
    <input type="text" name="usr" id="usr" placeholder="Username" />
    <label for="psw"><b> Contraseña: </b></label>
    <input type="password" name="psw" id="psw" placeholder="Password" />

    <input type="submit" id="b1" value="Enviar"/>
    </fieldset>
    </form>

    <a href="altausuario.php"> Registro de usuarios </a>';
}
?>
```

Al principio de estos ficheros (y al principio de determinados ficheros), se encuentra la llamada a la función para que la sesión se mantenga iniciada.

```
<?php
session_start();
$Admin="Admin";
?>
```

Con respecto a la práctica anterior, el código que mostraba la barra de navegación con las secciones ha sido sustituido por un fragmento PHP que llama a una función que extrae las secciones de la base de datos y las muestra en forma de lista.

```
<?php
    require_once('Seccion.class.inc');

    $arrayS=array();
    $Secciones=Seccion::obtenerSecciones($arrayS);
?>
```

```
/*
 * Función para obtener las secciones de la web
 */
public static function obtenerSecciones($secciones){
    $consulta="SELECT * FROM Secciones";
    $conexion=parent::conectar();
    $resultados=$conexion->query($consulta);

    foreach($resultados as $fila){
        echo '<nav class="BarNav">
            <ul id="bn">
                <li> <a href="seccion1-1.php" target="_blank">'.$fila["Nombre"].' </li>
            </ul>
        </nav>';
    }

    parent::desconectar($conexion);
}
```

Existen además dos ficheros, 'configuracion.inc' y 'datosObject.class.inc' que contienen una serie de variables y funciones globales a todos los ficheros con formularios. De la clase 'DataObject' heredan el resto de clases de los formularios las funciones para obtener un campo de la base de datos, así como conectarse y desconectarse de la misma.

El resto de fragmentos de código en PHP incrustados en código HTML están relacionados con los formularios. Existe además un fichero 'procesarFormulario.php' que comprueba la validez de los datos extraídos de la base de datos.

2.2. Formularios.

La práctica contiene un formulario para iniciar sesión, el cual aparece tal y como se ha explicado anteriormente, un formulario para darse de alta como nuevo usuario, un formulario para modificar los datos personales de un perfil, un formulario para añadir un ítem y un formulario para modificar los datos de un ítem.

El resto de formularios, como los destinados a eliminar usuarios, ítems o secciones, se encuentran en la pestaña de administración directamente.

2.2.1. Usuarios.

Para los formularios destinados a la gestión de usuarios, tanto por parte de los mismos como por parte de la administración, existe una clase denominada 'Usuario' contenida en el fichero 'Usuario.class.inc' que incluye todas las funciones para insertar, modificar, eliminar o validar el inicio de sesión de los usuarios en función del contenido de la base de datos. En dicha base de datos se encuentra la tabla 'Usuarios' que contiene los datos asociados a uno.

El formulario para darse de alta se encuentra en el fichero 'altausuario.php' y el formulario para modificar el perfil se encuentra en el fichero 'modificarperfil.php'. Cada uno de estos ficheros cuenta con un fichero extra para procesar los formularios ('procesar_altausuario.php' y 'procesar_modificarperfil.php' respectivamente). En estos ficheros, se extrae la información de los formularios y se realizan las llamadas a las funciones pertinentes para interactuar tanto con la base de datos como con los ficheros correspondientes.

En relación al inicio y al cierre de sesión existen los ficheros 'procesar_login.php' y 'procesar_logout.php' cuya función es realizar las llamadas a las funciones necesarias para la gestión de la sesión. La sesión abierta podrá, como se indica anteriormente, mantenerse abierta hasta que el usuario decida cerrar la misma.

2.2.2. Items.

En cuanto a los ítems, la estructura de ficheros es exactamente igual a la estructura de ficheros de los usuarios mostrada en el apartado anterior, conteniendo evidentemente los formularios y ficheros correspondientes a la gestión de ítems.

Sin embargo, el formulario para modificar un ítem no puede extraer el ítem a modificar de la sesión actual, tal y como sucede con el usuario, por lo que existe un campo en el apartado de administración que envía el nombre del ítem a modificar desde dicho apartado al fichero 'modificaritem.php' donde se almacena en una variable. Ésta variable se envía a través de un 'input' de tipo 'hidden', para que no se muestre e interfiera con la visualización del formulario, al fichero 'procesar_modificaritem.php' donde se reciben los nuevos datos y se realizan las llamadas necesarias a las funciones de la clase ítem (dichero 'Item.class.inc').

2.3. MySQL, tablas.

Para el desarrollo de la práctica existen, en la base de datos asociada a mi usuario de Betatun, un total de tres tablas. La destinada al almacenamiento de las secciones ('Secciones'), la destinada a la gestión de usuarios ('Usuarios') y, por último, la encargada de contener los ítems ('Items').

La tabla de secciones contiene únicamente una columna para los nombres de las mismas.

Las otras dos tablas contienen un número de columnas mayor ya que se corresponden con los datos de los usuarios y los ítems.

La conexión y desconexión de la base de datos se realiza, como se ha concretado en otro apartado, mediante las funciones de la clase 'DataObject'.

Las consultas se realizan desde las funciones de las clases 'Seccion', 'Usuario' e 'Item' según corresponda. En la siguiente imagen se muestra un ejemplo de una función que realiza la inserción de un usuario en la base de datos. La estructura de la función es similar en el resto de ellas.

Las funciones de la clase 'Process_form' como la de la imagen ('campoCorrecto(\$valor)') se encuentran en el fichero 'procesarFormulario.php' mencionado anteriormente.

```
/*  
 * Función para insertar un usuario en la tabla 'Usuarios'  
 */  
public static function insertarUsuario($datosU){  
    $campo=new Process_form();  
    $nom=$datosU;  
    $conexion=parent::conectar();  
  
    try{  
        foreach($nom as $valor){  
            $campo->campoCorrecto($valor);  
        }  
    }catch(Exception $e){  
        echo "</br> Error, ".$e->getMessage()."</br>";  
    }  
  
    $consultaSQL="INSERT INTO Usuarios VALUES ('$datosU[0]', '$datosU[1]', '$datosU[2]',  
                                                '$datosU[3]', '$datosU[4]', '$datosU[5]',  
                                                '$datosU[6]', '$datosU[7]', '$datosU[8]',  
                                                '$datosU[9]', '$datosU[10]')";  
  
    try{  
        $conexion->query($consultaSQL);  
        parent::desconectar($conexion);  
    }catch(PDOException $e){  
        parent::desconectar($conexion);  
        echo "Error de consulta: ".$e->getMessage()."</br>";  
    }  
}
```

2.4. JavaScript.

La corrección de la totalidad de los formularios de la práctica está validada mediante este lenguaje. En la imagen siguiente se muestra el script correspondiente a la comprobación de la corrección del formulario de inicio de sesión así como su llamada desde el propio formulario. El resto de comprobaciones de los formularios sigue una estructura equivalente adaptada a los campos de cada uno.

```
<form class="iniciar_sesion" name="iniciar_sesion" action="procesar_login.php" onsubmit="return validarFormulario()"
```

```
<script>
function validarFormulario(){
    let required=["usr", "psw"];

    for(campo of required){
        if(document.forms["iniciar_sesion"][campo].value==""){
            alert("El campo "+campo+" es obligatorio");
            return false;
        }
    }
}
</script>
```

Se almacenan los campos obligatorios en un array y se procede a realizar las comprobaciones necesarias.

Por otra parte, en el momento en el que el cursor del ratón se posicione encima de la imagen de un ítem, se mostrará debajo de la imagen un texto emergente con el nombre y la sección del ítem.

```
<p id="imgT"> <?php echo $nombreI ?> - <?php echo $seccion ?> </p>

<script>
    let img=document.getElementById('img');
    let name=document.getElementById('name');
    let sec=document.getElementById('sec');

    document.getElementById('imgT').style.display='none';

    img.onmouseover=function(){
        document.getElementById('imgT').style.display='block';
        document.getElementById('imgT').style.color='white';
    }

    img.onmouseout=function(){
        document.getElementById('imgT').style.display='none';
    }
</script>
```

Nombre: Arturo Alonso Carbonero
DNI: 75936665-A
Grupo: 3º - A