

Práctica evaluable II

Autor: Antonio Guzmán Martín

Cómo se hizo la práctica

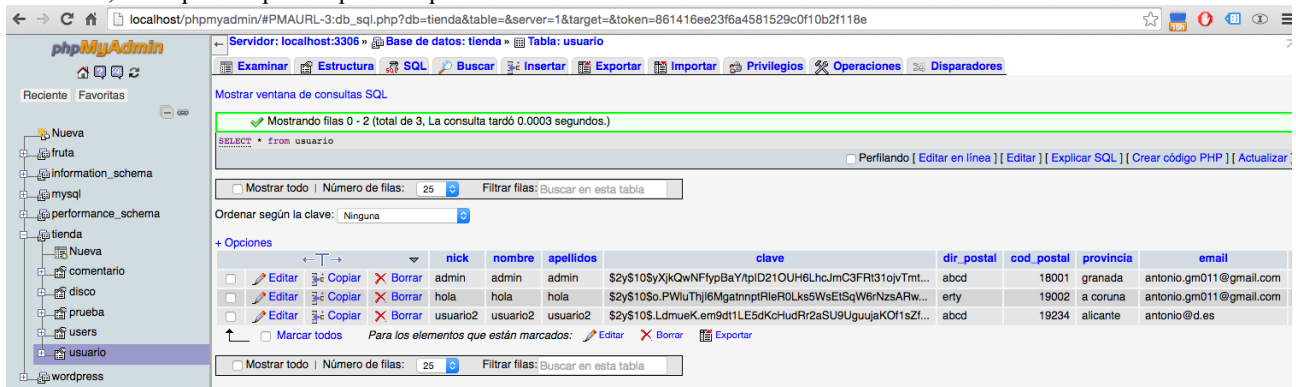
1. Dar de alta a los usuarios en la base de datos

Para dar acceso a los usuarios he necesitado los archivos suscribirse.php, sesion.php, home.php, class.usuario.php y dbtable.sql.

Dbtable.sql: Contiene el código sql necesario para almacenar cada uno de los campos introducidos en el formulario (nombre, apellidos, email...)

Class.usuario.php: Es la clase que permite registrar, iniciar sesión, comprobar que están en línea y desconectar a los usuarios.

El constructor de dicha clase crea una conexión con la base de datos para ejecutar consultas. **Registrar** por ejemplo realizará un INSERT en la base de datos cuando el formulario suscribirse en suscribirse.php se haya enviado. Esta función tiene la capacidad de cifrar contraseñas mediante password_hash y password_verify sin embargo en bahia no es posible probar esta cualidad puesto que la versión php que se corre es la 5.4.1 y estas funciones funcionan a partir de las 5.5, se ha podido probar por otra parte en un servidor local de forma satisfactoria.



Se usa la función bind_param para vincular el parametro con una variable dada, antes de ejecutar la consulta con execute().

```
$stmt = $this->connect->prepare("INSERT INTO usuario(nick,nombre,apellidos,email,clave,dir_postal,cod_postal,provincia,dni,visa,observaciones,envio)
VALUES(:unick, :unombre, :uapellidos, :uemail, :upass, :udir_postal, :ucod_postal, :uprovincia, :udni, :uvisa, :uobservaciones, :uenvio);");

$stmt->bindParam(":unick", $unick);
$stmt->bindParam(":unombre", $unombre);
$stmt->bindParam(":uapellidos", $uapellidos);
$stmt->bindParam(":uemail", $uemail);
$stmt->bindParam(":upass", $upass); // $stmt->bindParam(":upass", $password_hashed);
$stmt->bindParam(":udir_postal", $udir_postal);
$stmt->bindParam(":ucod_postal", $ucod_postal);
$stmt->bindParam(":uprovincia", $uprovincia);
$stmt->bindParam(":udni", $udni);
$stmt->bindParam(":uvisa", $uvisa);
$stmt->bindParam(":uobservaciones", $uobservaciones);
$stmt->bindParam(":uenvio", $uenvio);
```

Tras ejecutar la consulta se inserta al user en la BD.

Función doLogin inicia la sesión de un usuario existente comprobando si la contraseña almacenada y la suministrada en index.php son iguales. Si el resultado es positivo inicia sesión

La **función doLogout** hace un session_destroy() para acabar con la sesión del usuario.

Suscribirse.php: Permite el registro de nuevos usuarios, siempre que el usuario no este ya logueado

```
if($usuario->is_loggedin()!="")//si esta logueado
{
    $usuario->redirect('home.php');
}
```

El formulario tiene validación con javascript "onsubmit", si todos los campos son rellenados correctamente se procesará el formulario siempre y cuando el usuario ya no exista, ya que antes se hace una consulta a la base de datos para comprobar que este no existe.

Si el usuario no existe entonces se registra y con \$_GET establecemos que el usuario se a registrado correctamente y le pedimos que inicie sesión en index.php con un link a este archivo.

```
303     <?php
304         if(isset($error))
305         {
306             foreach($error as $error)
307             {
308                 ?>
309                 <?php echo $error; ?>
310                 <?php
311             }
312         }
313     else if(isset($_GET['joined']))
314     { //se ha registrado el usuario muestro el boton de iniciar sesion
315     ?>
316         <section class="registered">
317         <p>Registrado correctamente</p> <a href='index.php'>LOGIN</a>
318         </section>
319     <?php
320     }
321     ?>
```

Sesion.php: Lo único que hace es crear un usuario, comprobar si está logueado y en caso de que no lo este evita que entre al home.php, que es donde se encuentra la página inicial con el menú personal de cada tipo de usuario.

```
1  <?php
2
3  session_start();
4
5  require_once 'class.usuario.php';
6  $session = new USUARIO();
7  $nick = $_SESSION['usuario_sesion'];
8
9
10
11  if(!$session->is_loggedin() &&
12  ($_SERVER['SCRIPT_NAME']=='/~x75929747/tiendamusicaII/home.php' || $_SERVER['SCRIPT_NAME']=='/~x75929747/tiendamusicaII/formulario-disco.php'))
13  {
14      $session->redirect('/~x75929747/tiendamusicaII/index.php');//se evita con este archivo acceder a home.php sin iniciar sesión
15  }
16  else if($_SERVER['SCRIPT_NAME']=='/~x75929747/tiendamusicaII/formulario-disco.php' && $nick!='admin'){
17      $session->redirect('/~x75929747/tiendamusicaII/index.php');
18  }
19
```

2. Incorporar comentarios a los discos.

Se hace posible con los archivos comentarios.sql y class.comentario.php.

En la página del disco lo que se ha comprobado es que para que el usuario pueda comentar primero debe comprobarse que está logueado, después se asegura que se ha pulsado el botón enviar y a continuación se realiza la consulta a la base de datos para insertar dicho comentario.

```
$auth_usuario = new USUARIO();//usuario autenticado
if($auth_usuario->is_loggedin()!="")//si no esta logueado
{
    $nick = $_SESSION['usuario_sesion'];
    $stmt = $auth_usuario->runQuery("SELECT * FROM usuario WHERE nick=:nick");
    $stmt->execute(array(":nick"=>$nick));
    $userRow=$stmt->fetch(PDO::FETCH_ASSOC);
    $new_comment= new COMENTARIO;
    if(isset($_POST['enviar']))
    {
        $ctexto = strip_tags($_POST['comentarios']);
        $timezone = date_default_timezone_get();
        date_default_timezone_set($timezone);
        $cfecha = date('Y/m/d', time());
        $chora = date('H:i:s',time());
        $cgenero="edm";
        $new_comment->makeComment($nick,$cfecha,$chora,$cgenero,$ctexto);
        $auth_usuario->redirect("armin_van_buuren_embrace_20151029.php");//tiene el objetivo de desactivar el POST_enviar al recargar
    }
}
```

Se comprueba con código javascript que el campo comentario no queda vacío.

Class.comentario.php: Dos funciones a remarcar ShowComments, donde se hace el insert del comentario y countComments que permitirá llevar una cuenta de los comentarios realizados en la página.

Comentarios.sql: Los comentarios se han identificado por el nick del usuario, hora del comentario y fecha. Además está el campo género que permite conocer en que disco se está comentando y obviamente texto para almacenar los

comentarios.

3. Añadir discos a la página

Para añadir los discos a la página son necesarios: formulario-disco.php, class.disco.php y disco.sql

Formulario-disco.php: Es una nueva página que simplemente contendrá un formulario con corrección javascript de los campos nombre-disco, autor y género. Esta página solo podrá ser accedida por admin (ver sesion.php). Los formularios se corrigen con javascript.

Class.disco.php: La función que permitirá insertar un nuevo disco será newAlbum , que hace un insert de la misma forma que en las otras clases ya comentadas.

La función showAlbums mostrará los albums en cada página según el género que le corresponda, mediante la operación select.

```
43 public function showAlbums($agenero){
44
45     $stmt = $this->connect->prepare("SELECT * FROM DISCO WHERE genero=:agenero");//queremos todos los albums
46     $stmt->execute(array(':agenero'=>$agenero));//esto ejecuta la consulta teniendo en cuenta el genero deseado
47     $rows=$stmt->fetchAll();//devolver el resultado como un array indexado por nombre de columna
48     return $rows;
49 }
50
51
52 }
```

Disco.sql: Tiene los campos nombre-disco,autor y género y se identificará por nombre-disco y género.

4. Ventana emergente al pasar el ratón.

Para conseguir este efecto, se ha usado código javascript y las funciones window.open y window.close()

```
function show_list_album()
{
    miVentana=window.open("", "miVentana", "width=200,height=300");
    miVentana.document.write("<p>Embrace</p>");
    miVentana.document.write("<p>Another you</p>");
    miVentana.document.write("<p>Strong ones</p>");
    miVentana.document.write("<p>Make it right</p>");
    miVentana.document.write("<p>Face of summer</p>");
    miVentana.document.write("<p>Hands to heaven</p>");
    miVentana.document.write("<p style='color:red;font-weight:bold;'>Click en imagen para cerrar</p>");
}

function hide_list_album()
{
    miVentana.close();
}
```