

# Ejercicios\_UT6\_1\_Victor\_Valdes\_Cobos

Mi aplicación consiste en un gestor de videoclubs sencillo para practicar POO y PDO en php.

La base de datos es gestionada en PHPMyAdmin y tiene la misma estructura que la dada en el enunciado, con el detalle de que hay una tabla Log sin relación con ninguna otra tabla donde se guarda toda la información sobre inicios de sesión.

## Tecnologías y herramientas utilizadas)

8.2.0 Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.2.0

IDE: NETBEANS

## Diseño)

El diseño principal consiste en tablas y en ocasiones una tabla secundaria que en realidad es una caja div flex que ocupa todas las columnas de ancho.

El login consiste en una mezcla de dos presets bootstrap como explico posteriormente.

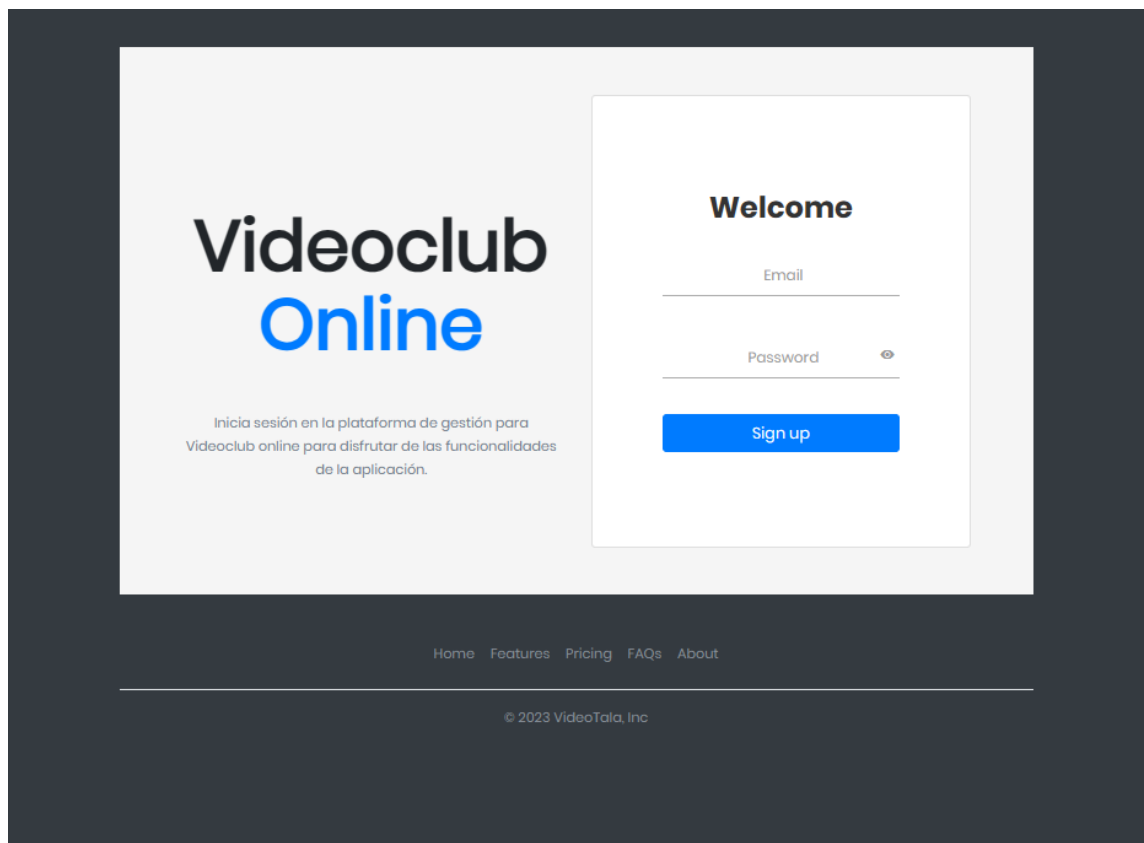
Hay un footer con información de la página web y la información de la tecnología usada y un header que solo contiene el botón de cerrar sesión.

## IMPLEMENTACIÓN)

La aplicación cumple todos los puntos opcionales y obligatorios del enunciado.

Explicación punto por punto:

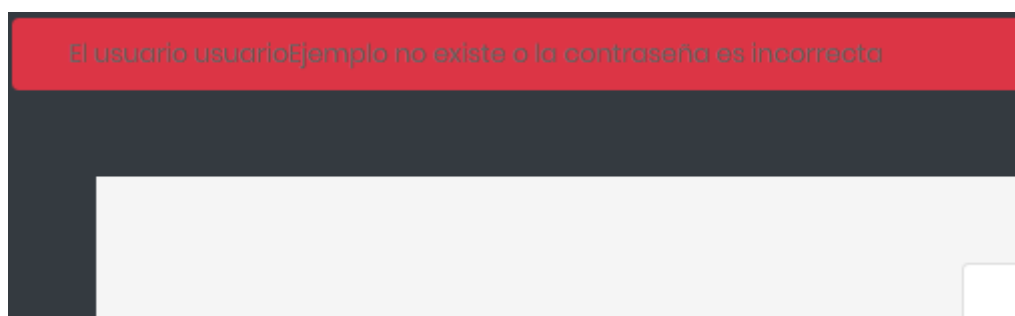
### 1) Login



El estilo del login consiste en una mezcla entre dos templates basados en Bootstrap.

A)

Cuando no existe el usuario en la base de datos, se genera un mensaje en pantalla que avisa de ello:



B)

Cuando el usuario ha iniciado sesión, se tiene en cuenta su rol:

Solo cuando el usuario y la contraseña son correctos, se instancia usuario; cuando esto ocurre aprovecho la variable superglobal sesión para guardar el rol de forma encriptada.

```
56 private function actualizarSesion(&$sesion_aux,$encriptar) {  
57     //Encriptado  
58     if($encriptar===true) $this->rol = password_hash($this->rol, PASSWORD_BCRYPT);  
59     $sesion_aux['rol'] = $this->rol;  
60 }
```
















Y de esta forma compruebo el rol:

```
$sesRolAdmin = isset($_SESSION["rol"]) && password_verify('1', $_SESSION['rol']);
```

Uso password\_hash() ya que según ciertas páginas en internet, es una forma segura de encriptar información ya que, el algoritmo PASSWORD\_BCRYPT se va actualizando con el tiempo y mediante password\_verify() verificaré su valor original sin importar futuros cambios en la forma de encriptar información con password\_hash().

## 2) Funciones del usuario administrador:

### A) Ver una lista con todas las películas junto a los controles 'Modificar' y 'Borrar':

id	título	genero	pais	año	cartel	5	6	7
1	El Señor de los Anillos	Fantasia	Estados Unidos	2001		Eliminar	Modificar	
<div> <div>Elijah Wood</div>  </div> <div> <div>Scarlett Johansson</div>  </div> <div> <div>Brad Pitt</div>  </div> <div> <div>Jennifer Lawrence</div>  </div>								
2	Titania	Romance	Estados Unidos	1997		Eliminar	Modificar	
<div> <div>Kate Winslet</div>  </div> <div> <div>Scarlett Johansson</div>  </div> <div> <div>Brad Pitt</div>  </div> <div> <div>Leonardo DiCaprio</div>  </div> <div> <div>Emma Stone</div>  </div>								
3	Pulp Fiction	Crimen	Estados Unidos	1994		Eliminar	Modificar	
<div> <div>John Travolta</div>  </div> <div> <div>Jennifer Lawrence</div>  </div> <div> <div>Emma Stone</div>  </div>								

## B) Funcionalidad borrar:

La aplicación permite borrar múltiples películas simultáneamente:

Esto es lo que pasa cuando elimino una película:

Los botones vienen determinados por su propio atributo name, este se compone de dos partes: 'eliminarPelícula\_' y el id de la misma:

```
name="eliminarPelículaId_2">
```

Esto facilita mucho el proceso ya que el propio botón indica lo que tiene que ocurrir. El botón refresca la página con su nombre en el array POST (Cabe destacar que toda la página homePage contiene solo un formulario independientemente del rol del usuario). Una función regida por un switch interpreta el array POST:

```
255 function gestionarFuncionalidad() {  
256     switch (FuncionalidadPelículas($funcionalidadID)) {  
257         case "anadirPelícula":  
258             anadirPelícula($ _POST["maxIDPelículas"]);  
259             break;  
260         case "modificarPelícula":  
261             modificarPelícula($funcionalidadID["id"]);  
262             break;  
263         case "eliminarPelícula":  
264             eliminarPelículas($funcionalidadID);  
265             break;  
266         case "enviarMail":  
267             enviarMail(direccionMail($funcionalidadID["id"]), $funcionalidadID["mensaje"]);  
268             break;  
269         default:  
270             break;  
271     }  
272 }
```

Gestionar funcionalidad interpreta POST y controla la funcionalidad, esta función solo es llamada en este caso:

```
if ($ _SERVER["REQUEST_METHOD"]=="POST" && comprobarLogin($tabla)) {  
    isset($ _POST["miUsuario"]) ? gestionarFuncionalidad() : crearInstanciaUsuario($miUsuario, $tabla, $ _SESSION);  
}
```

Cuando el método del servidor es POST y el usuario y contraseña son correctos; y si existe la instancia usr en POST.

Dentro de gestionarFuncionalidad() la función más importante es funcionalidadPelículas() que contiene la mayor parte de la lógica. Se encarga de interpretar POST de la forma más óptima posible. Por orden de prioridad recorriendo el array POST trabaja para estos casos: Cuando hay que enviar un mail, cuando se añade una película, cuando se modifica una película, y por último cuando se elimina.

Cuando se va a eliminar una película, funcionalidadID se convierte en un array de dos posiciones que valdrá el string eliminarPelícula y el id de la película.

Cuando se han seleccionado más películas, la búsqueda realizada en post ocurre de la siguiente forma: mientras se va recorriendo el array se van guardando los ids de los selectores seleccionados y, hasta que no se encuentra un valor eliminarPelícula, el algoritmo no se decide a borrar, puede ser que se vaya a modificar una película aunque haya selectores seleccionados,

a partir de este momento la búsqueda pasa a 'modo eliminar' y solo busca otros posibles selectores seleccionados después de haber encontrado el valor eliminar.

(Los selectores al igual que los botones contienen una nomenclatura significativa en su nombre que contienen la funcionalidad y el id de la película seleccionada)

```
function funcionalidadPelículas(&$funcionalidadID) {
    // Get all keys from $_POST.
    $clavePost = array_keys($_POST);
    $arrSelectorIDs = array();
    $enProcesoModificar = false;
    for ($i = 0; $i < count($clavePost); $i++) {
        $key = $clavePost[$i];
        // Check if the key contains "selectorMovieId".
        if (strpos($key, "enviarMailId") === 0) {
            $funcionalidadID = array("id" => obtenerID($key), "mensaje" => $_POST["mensajeMail"]);
            return "enviarMail";
        }
        if (strpos($key, "selectorPelículaId") === 0) {
            // Get the ID and add it to the array of selector IDs.
            $id = obtenerID($key);
            array_push($arrSelectorIDs, $id);
        }
        if (strpos($key, "modificar") === 0) {
            if(!$enProcesoModificar) $enProcesoModificar = true;
        }
        // Check other functionalities based on $_POST keys.
        if (strpos($key, "anadirPelícula") === 0) {
            // Return the "anadirPelícula" functionality.
            $funcionalidadID = array("funcion" => "anadirPelícula");
            return "anadirPelícula";
        } else if (strpos($key, "eliminarActor") === 0) {
            $id = obtenerID($key);
            $funcionalidadID = array("funcion" => "eliminarActor", "id" => $id);
            return "eliminarActor";
        } else if (strpos($key, "modificarPelículaId") === 0) {
            // Return the "modificarPelícula" functionality.
            $id = obtenerID($key);
            $funcionalidadID = array("funcion" => "modificarPelícula", "id" => $id);
            return "modificarPelícula";
        } else if (strpos($key, "eliminarPelículaId") === 0) {
            //Devuelve array(eliminarPelículaId,idPelícula)
            //Si me encuentro en la ultima posicion y nos e han encontrado selectores...
            //Return eliminar solo con ese id
            // Iterate over the remaining keys to collect selector IDs.
            while ($i < count($clavePost) && $enProcesoModificar === false) {
                $key = $clavePost[$i];
                if (strpos($key, "selectorPelículaId") === 0 || strpos($key, "eliminarPelículaId") === 0) {
                    $id = obtenerID($key);
                    array_push($arrSelectorIDs, $id);
                }
                $i++;
                if (strpos($key, "modificar") === 0) {
                    if(!$enProcesoModificar) $enProcesoModificar = true;
                }
            }
            // Return the "eliminarPelícula" functionality.
            $funcionalidadID = array("funcion" => "eliminarPelícula", "id" => $arrSelectorIDs);
            return $enProcesoModificar === false ? "eliminarPelícula" : '';
        }
    }
}
```

Esta función devolvería la funcionalidad eliminarPelícula con la variable puntero correspondiente y llamaría a la función eliminarPelículas() la cual solo interpreta funcionalidadID, esta función a su vez llama a eliminarDatos, que es la función que modifica la base de datos satisfactoriamente.

### C) Añadir película

Cuando se acciona el botón de añadir película aparece un modal para introducir los datos necesarios.

Estos datos pasan a la variable POST, gestionarFuncionalidad interpreta Post y llama a la función anadirPelícula que a su vez llama a la función insertar que modifica la base de datos:

```
function anadirPelícula($maxIDPelículas) {
    insertar("películas", array("id" => $maxIDPelículas + 1, "titulo" => $_POST["nuevaPelícula_Titulo"], "genero" => $_POST["nuevaPelícula_Genero"]));
}
```

El id se genera automáticamente siendo este una unidad mayor al máximo id en la tabla. Esta variable es demasiado absurda como para hacer solo una consulta para ello, así que viene dada por un parámetro e/s de la función crearInstanciasPelícula():

```
if (!isset($arrPelículas)) {
    $arrPelículas = crearInstanciasPelícula($tabla, $maxIDPelícula);
}
```

#### D) Modificar una película

Consta de dos pasos, uno consiste en cambiar cada parámetro del elemento en la tabla en un input, excepto el id, y el segundo paso consiste en aplicar los valores introducidos y modificar la base de datos:

El primer paso es muy parecido a eliminar película, el propio botón en su nombre contiene la función y el id.

Cuando se modifica una película también se puede cancelar el proceso.

gestionarFuncionalidad llama a modificarPelícula y como todavía nos encontramos en el primer paso solo hace esto:

```
} else {
    $_POST["mostrarInputsPelícula_ID"] = $id;
}
```

Inicia esta instancia en POST.

Esta instancia es interpretada por la función que imprime la tabla y esta llama a la función que imprime inputs en la fila que corresponde al ID:

```
foreach ($arrAtributos as $nombre => $valor) {
    if (isset($_POST["mostrarInputsPelícula_ID"]) && $_POST["mostrarInputsPelícula_ID"] == $arrAtributos["id"]) {
        $html .= entornoTd(entornoInputsModificarPelícula(imprimirAtributosInput($nombre, $valor, $id), $valor));
    } else {
        $html .= entornoTd(imprimirAtributosMostrar($nombre, $valor, $id));
    }
}
```

gestionarFuncionalidad interpreta los datos pasados por el formulario en la tabla y llama a modificarPelícula, que llama a modificarTabla que es la función que modifica la base de datos.

### 3) Funciones del usuario normal

#### A) Mostrar una lista de películas

Debido a que guardo el rol en la variable superglobal SESSION, esta tarea ha sido fácil de hacer.

```
$esRolAdmin = isset($_SESSION["rol"]) && password_verify('1', $_SESSION['rol']);
echo $esRolAdmin
? entornoFormularioPrincipal(anadirListaPazo(imprimirTablaPeliculas($arrPeliculas,
: imprimirTablaPeliculas($arrPeliculas, $arrActores, $tablaActuan, $arrActoresPazo,
```

Debido a que solamente se ha de mostrar una lista, no necesito formulario para la tabla películas ni necesito la lista de actores en paro. Además en imprimirTablaPeliculas hay un parámetro que imprime o no los botones modificar y modificar (\$tablaNoFuncional):

```
195 * Devuelve el html de la tabla correspondiente, $arrTablaExtra supone un div debajo de cada
196 * fila de la tabla principal, relacionada por $arrTablaRelacion. $arrActoresPazo en esencia
197 * es un parámetro e/s que devuelve los id's de la tabla extra no usados en la tabla.
198 *
199 * @param array $arrPeliculas
200 * @param array $arrTablaExtra
201 * @param array $arrTablaRelacion
202 * @param array $arrActoresPazo
203 * @param type $tablaNoFuncional
204 * @param boolean $enviarMail
205 * @return string
206 */
function imprimirTablaPeliculas($arrPeliculas, $arrTablaExtra = null, $arrTablaRelacion = null,
```

## B) FORMULARIO PARA ESCRIBIR MAIL A ADMINISTRADOR A ELEGIR

```
92 if (!$esRolAdmin) {
93     $tabla=extraerTablas("SELECT * FROM USUARIOS WHERE ROL = 1",true);
94     $arrAdmins=crearInstanciasAdminsAux($tabla);
95     echo entornoFormulario(inputsFormularioMailAdmin($arrAdmins),$miUsuario);
96 }
```

Primero creo las instancias usuario para los administradores obteniendo la tabla de la base de datos. CrearInstanciasAdminsAux() instancia usuarios según la tabla obtenida desde la BD, estos usuarios solo tendrán rellenos los atributos usuario e id:

```
177 function crearInstanciasAdminsAux($tabla) {
178     $arrAdmins = array();
179     $arrAux = array();
180     foreach ($tabla as $admin) {
181         $usuarioAux = new Usuario(
182             $admin["id"],
183             $admin["username"],
184             null,
185             null,
186             $arrAux,
187             false
188         );
189         array_push($arrAdmins, $usuarioAux);
190     }
191     return $arrAdmins;
192 }
```

Esto es posible gracias a que el constructor de usuario tiene ciertos parámetros iniciados a null. Esto ahorrará espacio en memoria y aumentará la seguridad, ya que veo totalmente innecesario guardar datos como la contraseña de los admins cuando no son necesarios.

Para imprimir la tabla donde selecciono el admin, reutilizo imprimirTablaPeliculas, ya que esta función es altamente reutilizable y escalable:



```
39 function inputsFormularioMailAdmin($tablaAdmins) {
40     return imprimirTablaPeliculas($tablaAdmins, null, null, $arrActoresPare, true, true) .
41         '<div class="form-group">
42             <label for="exampleFormControlTextarea1">Example textarea</label>
43             <textarea name="mensajeMail" class="form-control" id="exampleFormControlTextarea1" rows="5">
44             </div>';
45 }
```

La función solo necesita de una tabla admin y tendrá controles para enviar el mail.

La tabla es rodeada de un formulario que además contendrá un label y un textarea para escribir el mensaje.

Imprimir tabla tiene como true el parámetro enviarmail, lo que hace posible esto:

```
242 if ($enviarMail)
    $html .= imprimirControlMail($arrAtributos["id"]);
```

Que a su vez imprime un botón que añade una instancia POST con el id del admin a enviar mensaje.

Una vez clicko en el botón, el formulario refresca la página, lo que permite que gestionarFuncionalidad, al igual que en todos los demás casos, interprete el array POST y llame a la siguiente función:

```
279 case "enviarMail":
280     enviarMail(direccionMail($funcionalidadID["id"]), $funcionalidadID["mensaje"]);
281     break;
```

direccionMail devuelve la dirección mail del id del usuario y enviar mail envia el mensaje a la dirección de mail indicada:

```
function enviarMail($mailReceptor, $mensaje) {
52
53     //HAGO ESTO PARA QUE SE CARGUEN LAS DEPENDENCIAS DE PHP MAILER SOLO CUANDO HAYA QUE ENVIAR UN MAIL
54     //DE CASO CONTRARIO SE CARGARÍAN CADA VEZ QUE SE USA UNA FUNCIÓN
55     // Ruta al archivo PHP que deseas incluir
56     $rutaPHPMailer = $_SERVER['DOCUMENT_ROOT'] . '/Ejercicios_UT6_1_Victor_Valdes_Cobos/libraries/funciones/PHPMailer.php';
57     // Verificar si el archivo existe antes de incluirlo
58     if (file_exists($rutaPHPMailer)) {
59         $_SESSION["mailReceptor"]=$mailReceptor;
60         $_SESSION["mensaje"]=$mensaje;
61         // Incluir el archivo
62         include $rutaPHPMailer;
63         unset($_SESSION["mailReceptor"]);
64         unset($_SESSION["mensaje"]);
65     } else {
66         // Manejar el caso en que el archivo no existe
67         echo 'El archivo no existe.';
68     }
69 }
```

El mail receptor y el mensaje son guardados en el array SESSION para su posterior uso en el archivo que uso para ejecutar el código de PHPMailer.

Si no funciona enviar mail, descargar la carpeta vendor con PHPMailer dentro desde composer y añadir los archivos descargados a raíz/vendor, reemplazar si se da la opción. (Cuando reinicio mi pc, el archivo autoload se elimina solo y tengo que hacer este paso, creo que eso es lo que provoca el error.)

### Funciones generales de la aplicación )

## Gestión de sesiones y funcionalidad OPCIONAL Logs)

```
<div class="card-body py-3 px-md-3">
  <form method="POST" class="mt-4 p-4 d-flex flex-column" action='pages/homePage.php'>
```

Cuando inicio sesión, automáticamente voy a homepage y hago la comprobación ahí.

Esto puede desembocar ciertas opciones:

```
6 session_start();
7 if (isset($_POST["miUsuario"])) {
8   $miUsuario = unserialize(base64_decode($_POST["miUsuario"]));
9   $miUsuario->actualizarPost($_POST);
10  $miUsuario->actualizarSesion($_SESSION, false);
11 } else{
12   $_POST["password"]=hash('sha256', $_POST['password']);
13 }
14 if ($_SERVER["REQUEST_METHOD"]=="POST" && comprobarLogin($tabla)) {
15   isset($_POST["miUsuario"]) ? gestionarFuncionalidad() : crearInstanciaUsuario($miUsuario, $tabla, $_SESSION);
16   unset($password,$tabla);
17   if (isset($_POST["miUsuario"])) unset($_POST["miUsuario"]);
18   if (isset($_POST["password"])) unset($_POST["password"]);
19   if (isset($_POST["usr"])) unset($_POST["usr"]);
20 } else{
21   crearInstanciaLogError($_POST["usr"]);
22   isset($_POST["usr"]) ? $url = '../index.php?formError=true&usr=' . urlencode($_POST["usr"]) . '&mensajeNoExisteUser=true' : $url = '../index.php?formError=true';
23   $_SESSION=[];
24   session_destroy();
25   header('Location: '.$url);
26   exit();
27 }
28 ?>
```

En homepage se gestiona (POR ORDEN DE IF'S):

Si la instancia de POST llamada mi usuario ya está inicializada, quiere decir que ya estábamos en homepage anteriormente y, por tanto, había una instancia Usuario propietaria de la sesión ya creada.

Si el método es POST y el login es satisfactorio (es decir venimos del login y nos queremos loguear). Y si no (else): Se destruye la sesión devolviendo el nombre del usuario o una bandera que indica error si corresponde y se redirecciona al login.

Este es el botón para cerrar sesión ( se encuentra en header.php ):

```
echo '<li class="nav-item"><a class="nav-link" href="homepage.php?logOut=true">
  <p class="mx-2 text-danger">Cerrar sesión</p>
</a></li>';
```

Simplemente redirecciona a homepage (la página donde nos encontremos) y la gestión de homePage hace el resto, al no haber enviado nada por post, se cierra sesión automáticamente.

POO en la sesión: Creo el usuario de la sesión llamado \$miUsuario en homePage, veo innecesario hacerlo en index.

Cuando creo una instancia usuario ocurre esto:

```

24 public function __construct($id, $username=null, $password=null, $rol=null, &$sesion_aux, $reg
25     $this->id = $id;
26     $this->username = $username;
27     $this->password = $password;
28     $this->rol = $rol;
29     if ($registrarInstancia) {
30         new Log($id, $username, $error = false) && $this->actualizarSesion($sesion_aux, true);
31     }
32     //$this->actualizarPost($_POST);
33     //$this->iniciarCookieSesion();

```

Siempre que instancio una sesión actualizo la variable SESSION e instancio un Log

Log:

```

9 public function __construct($id, $username, $error) {
10     $fechaHora = date('Y-m-d H:i:s');
11     if ($error) {
12         $this->texto = "[ $fechaHora ] ERROR: Intento de inicio de sesión fallido para el usuario { $username }.";
13     } else {
14         $this->texto = "[ $fechaHora ] El usuario { $username } [ { $id } ] ha iniciado sesión correctamente.";
15     }
16     //Si no existe la tabla la crea, además inserta
17     !comprobarLog() ? crearTabla("log", array("texto" => "varchar(255)")) : null;
18     insertar("Log", array("texto" => $this->texto));
19     $this->crearCookieSesion();
20     //Aquí debería llamar al destructor
21     $this->__destruct();
22 }

```

El constructor se ocupa de todo, ya que la instancia log no tiene ninguna utilidad en el programa en su duración, cuando se crea la instancia, se destruye, pero en este profeso ya se ha creado un registro en la base de datos.

Cuando el logueo ha sido erróneo:

```

20 } else{
    crearInstanciaLogError($_POST["usr"]);

```

Llamo a la función crear instancia log Error que llama a la función crear instancia log con error:

```

104 function crearInstanciaLogError($username) {
105     new Log(null, $username, true);
106 }

```

El constructor de Log crea la tabla Log si esta no existe.

```

texto
[2023-12-21 00:18:55] ERROR: Intento de inicio de ...
[2023-12-21 00:19:00] ERROR: Intento de inicio de ...
[2023-12-21 00:19:01] ERROR: Intento de inicio de ...
[2023-12-21 00:19:04] ERROR: Intento de inicio de ...
[2023-12-21 00:21:38] ERROR: Intento de inicio de ...
[2023-12-21 00:22:54] ERROR: Intento de inicio de ...
[2023-12-21 00:23:05] ERROR: Intento de inicio de ...
[2023-12-21 00:23:12] ERROR: Intento de inicio de ...
[2023-12-21 00:25:32] El usuario admin [1] ha inic...
[2023-12-21 00:42:18] El usuario admin [1] ha inic...
[2023-12-21 00:47:39] El usuario admin [1] ha inic...
[2023-12-21 00:48:16] ERROR: Intento de inicio de ...

```

La sesión puede ser cerrada clickando en cerrar sesión, cuando cerramos sesión ocurre esto:

```

21 } else(
22     crearInstanciaLogError($_POST["usr"]);
23     isset($_POST["usr"]) ? $url = '../index.php?formError=true&usr=' . urlencode($_POST["usr"]) . '&mensajeN...' : $url = '../index.php?formError=true&usr=' . urlencode($_POST["usr"]) . '&mensajeN...';
24     $_SESSION=[];
25     session_destroy();
26     setcookie("ultCone", null, time() - 3600, 'localhost');
27     header('Location: '.$url);
28     exit();
29 }

```

## Cookies)

```

5 setcookie("ultCone", date("Y-m-d H:i:s"),time() + 30, 'localhost');

```

Esta es la cookie que indica la ultima conexión, esta cookie se inicia en index para luego mostrarse en homepage, esta cookie también se actualiza cada vez que creo una instancia Usuario y cada vez que refresco la página homePage.

La cooke es destruida cuando el método del servidor cargando homePage no es post, el login con los parámetros dados en index es erróneo o cuando clicko en cerrar sesión.

## RETO) Actores en paro

La función imprimir Tabla Películas sabe qué actores imprime, por tanto sabe qué actores no imprime, esto es devuelto mediante un parámetro e/s y a posteriori es usado por una función anadirListaParo, esta función siempre va a depender de la función imprimirListaPelículas, de ahí la palabra añadir: Nunca se va a dar el caso de que quiera imprimir elementos de una tabla que no tengan relación con otra, si la otra no existe.

```
function anadirListaPare($sinnerHtml = null, $arrActoresPareID, $arrActores) {
256     if ($sinnerHtml === null)
257         $sinnerHtml = '';
258     if (isset($arrActoresPareID) && count($arrActoresPareID) > 0) {
259         $sinnerHtml .= '<h1>Actores en pare</h1>';
260         $arrActoresPare = filtraTablaId($arrActores, $arrActoresPareID, true);
261         return $sinnerHtml .= imprimirTablaPeliculas($arrActoresPare, null, null, $arrActoresPareID, true);
262     } else {
263         $sinnerHtml .= '<h1>Actores en pare</h1>';
264         return $sinnerHtml .= imprimirTablaPeliculas($arrActores, null, null, $arrActoresPareID, true);
265     }
266 }
```

He intentado reutilizar todas las funciones posibles, uso imprimirTablaPeliculas también para imprimir actores en pare, estos actores en pare supondrán la tabla principal, y no habrá ninguna otra tabla incrustada.

Filtro la tabla de actores que ya extraje en una consulta con los ID's de los actores en pare, con esto obtengo las instancias necesarias del objeto Actor para usarlas en la función imprimirTablaPeliculas.

Actores en pare			
id	nombre	apellidos	fotografia
11	Tom	Hanks	tom_hanks.jpg
12	Meryl	Streep	meryl_streep.jpg
13	Denzel	Washington	denzel_washington.jpg
14	Natalie	Portman	natalie_portman.jpg
15	Will	Smith	will_smith.jpg

### W3C)

homePage.php admin:

#### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

#### Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by **text input** ▼

Enter the URL for an HTML, CSS, or SVG document

[Check](#)

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

Document checking completed. No errors or warnings to show.

homePage.php no admin:

## Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

### Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by  ☐ css

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    <title>Iniciar Sesión - Videoclub</title>

    <!-- Agrega el enlace al archivo CSS de Bootstrap -->
  </-->
```

[Check](#)

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

Document checking completed. No errors or warnings to show.

Index.php:

## Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

### Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by  ☐ css

```
<!--
=====
<script src="vendor/daterangepicker/moment.min.js"></script>
<script src="vendor/daterangepicker/daterangepicker.js"></script>
<!--
=====
<script src="vendor/countdowntime/countdowntime.js"></script>
<!--
=====
<script src="js/main.js"></script>

</body>
</html>
```

[Check](#)

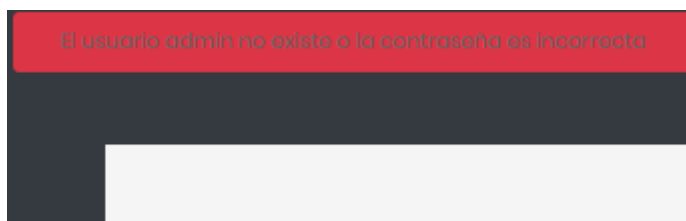
Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

1. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
[From line 46, column 47; to line 46, column 52](#)  
Videoclub `<br />`

Document checking completed. No errors or warnings to show.

### PRUEBAS)



Usuario no disponible

Añadir película

Título
Prueba
Género
Prueba
País
Prueba
Año
20088
Cartel
.jpg

El año no es válido

Control de cada parámetro al añadir película

3 Pulp Fiction Crimen | 1994 pulp\_fiction Cancelar Modificar

! Rellene este campo.

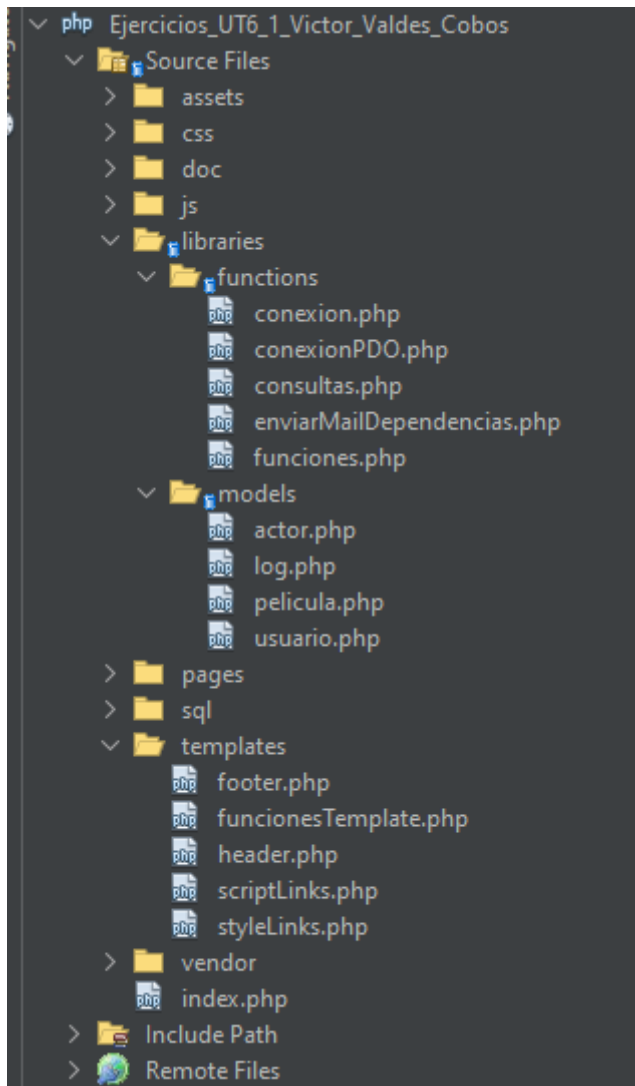
John Travolta Jennifer Lawrence Emma Stone

Prueba para modificar, no acepta valor vacío

### Organización y notas)

Organización:





Las funciones se dividen en funcionesTemplates.php, que pueden depender de otras funciones en el archivo funciones.php y devuelven código html principalmente, y funciones.php, que no dependen de funcionesTemplates.php.

He intentado refactorizar lo máximo posible, aún así podría refactorizar más a cambio de invertir todavía más tiempo en el trabajo:

imprimirTablaPeliculas() debería llamarse imprimirTablaPrincipal(), ya que es usada para cuatro tablas distintas, creo que las demás funciones tienen un nombre dentro de lo que cabe correcto, este error ha sido porque al principio del proyecto no pensé que iba a usar esta función para tantos puntos.

El orden de los parámetros en varias funciones es incorrecto, ya que las variables opcionales han de estar en la parte derecha de las funciones, esto es algo que desconocía antes de empezar y la causa de gran parte de los warnings del código.

#### OTRAS NOTAS:

En homepage.php:

```
21 } else{  
22     crearInstanciaLogError($_POST["usr"]);  
23     isset($_POST["usr"]) ? $url = '../index.php?formError=true&usr=' . urlencode($_POST["usr"]) . '&mensajeN=  
24     $_SESSION=[];  
25     session_destroy();  
26     setcookie("ultCone", null, time() - 3600, 'localhost');  
27     header('Location: '.$url);  
28     exit();  
29 }
```

Uso urlencode porque he leído que es un extra de seguridad para mandar parámetros por get y evitar un 'Man in the middle'. Debido a que creo el usuario en homePage y aprovecho su constructor para realizar todo lo necesario para empezar la sesión de forma correcta, si el inicio de sesión es erróneo tengo que redireccionar a index.php, para mandar parámetros en un redireccionamiento con header solo se hacerlo con el método GET. Hago todo esto para guardar el usuario introducido en caso login erróneo.

En la función funcionalidadPelículas uso esta función nativa:

```
(strpos($key, "eliminarPelículaId") === 0) {
```

Me ayuda a encontrar coincidencias entre un string y otro, si devuelve 0 estricto quiere decir que hay coincidencia, esto me ayuda a interpretar el array POST.

Todas las clases tienen los mismos atributos que en la base de datos, y la tabla actúan simplemente la guardo en un array.

En el directorio sql tengo un archivo llamado consultas de interés donde guardo dos consultas:

Si las peticiones a base de datos fuesen un problema creo que se podrían extraer casi todos los datos necesarios con dos o tres consultas, aun así estimo que con la aplicación que he hecho hago entorno a cinco consultas por cada refresco.

La segunda consulta es simplemente una consulta para vaciar la base de datos.

Para hacer la tabla donde muestro las películas totalmente escalables, tengo una función en todos los objetos menos en log llamada toArray, que hace lo que el nombre indica, gracias a esto no hay ningún tipo de máximo en la página.

```
35 private function toArray() {  
36     return get_object_vars($this);  
37 }
```

Usar arrays para imprimir atributos puede suponer un mayor uso de la memoria local en un pequeño espacio de tiempo para una gran mejoría en cuanto a escalabilidad bajo mi punto de vista, para mí no tiene sentido ir escribiendo a mano cada atributo.

He experimentado errores con los estilos por posibles errores de caché o cookies y no era capaz de solicitarlo hasta que hice esto:

Añadiendo una nueva versión a la carga del archivo de alguna forma deja de cargar a caché del archivo:

[CSS - No se actualiza CSS \(lawebdelprogramador.com\)](http://lawebdelprogramador.com)

```
21 <link rel="stylesheet" type="text/css" href="css/util.css?1.0" media="all" >
22 <link rel="stylesheet" type="text/css" href="css/main.css?1.0" media="all" >
23 <!--=====
```

Añadir modales cuando voy a eliminar una película sería una buena opción, pero no me ha dado tiempo a implementarlo.

```
//Interpreto el rol del usuario usando password_verify
$esRolAdmin = isset($_SESSION["rol"]) && password_verify('1', $_SESSION['rol']);
```

Encriptar un número de un solo dígito no sé si es muy seguro de todas formas, ya que solo hay dos estados posibles para esta variable, 0 y 1.

Pienso que devolver html en funciones que trabajan con él es muy cómodo, ya que este código html lo puedo manipular en otra función:

```
echo entornoFormularioPrincipal(
    anadirListaParo(
        imprimirTablaPeliculas($arrPeliculas, $arrActores, $tablaActuan, $arrActoresParo),
        $arrActoresParo,
        $arrActores
    ),
    $miUsuario,
    $maxIDPelicula
);
```

Esto me permite reutilizar muchas funciones.

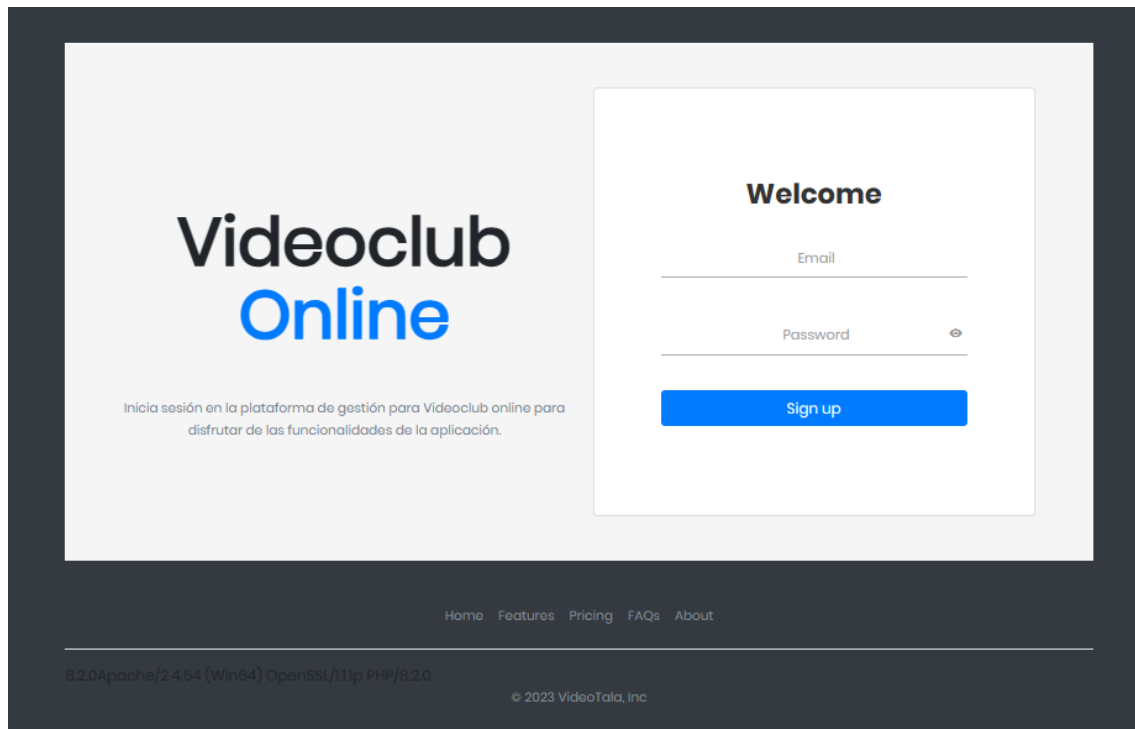
Faltan mejores pruebas a la hora de modificar película.

## (HOW-TO)

### Paso 1)

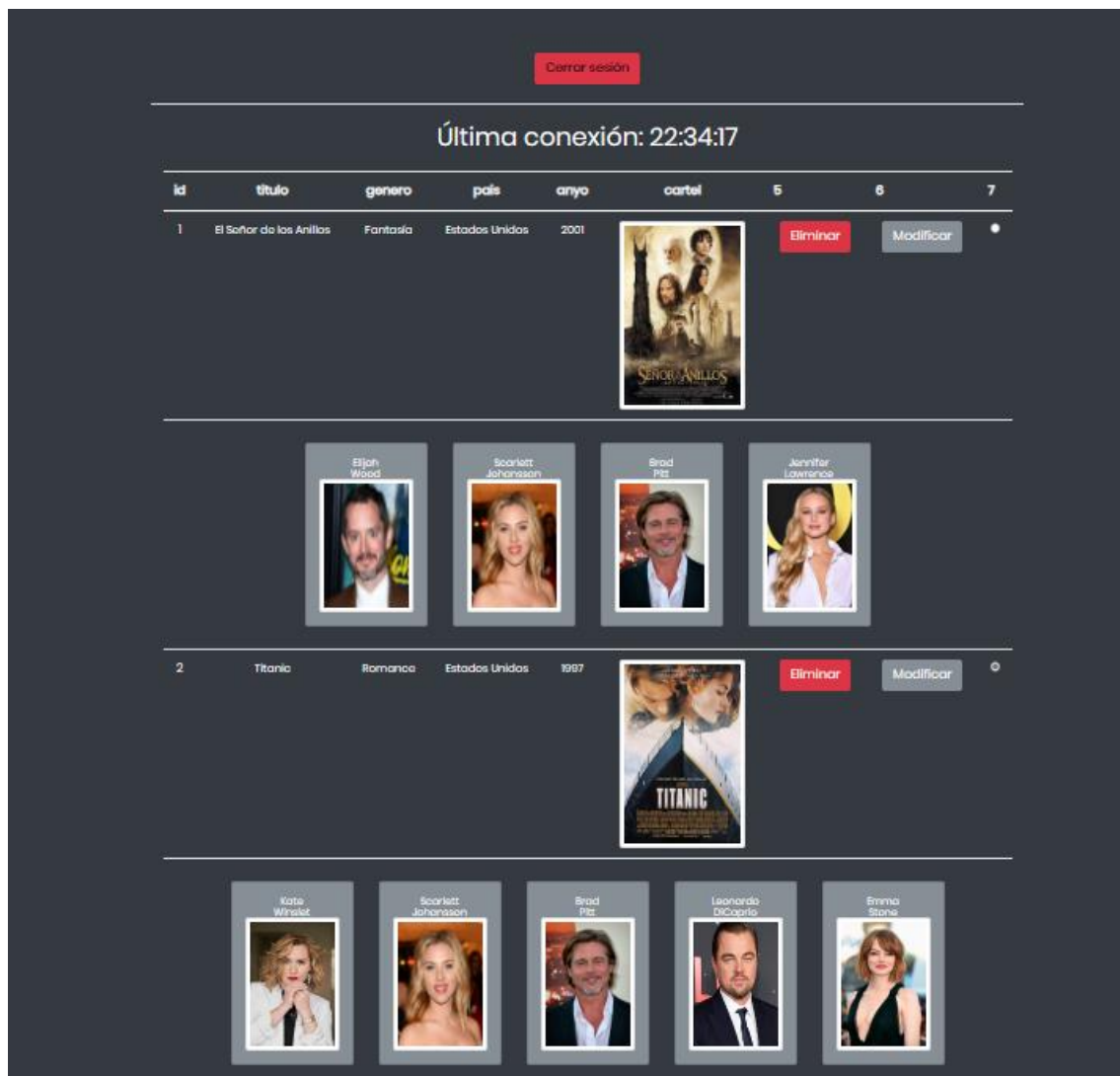
Iniciar sesión

Simplemente introducir las credenciales y clicar en Sign up



## Paso 2) Funcionalidades

Admin:



### Modificar)

Una vez clicado en modificar:

2 Titanic Romance Estados Unid 1997 titanic.jpg Cancelar Modificar

Cancelar para cancelar modificación y modificar de nuevo para aceptar los cambios, todos los valores han de estar rellenos


### Eliminar)

Rellenar los checks de las películas que se van a eliminar y clicar en el botón de eliminar, ¡cuidado! La película que se encuentre en la fila del botón que hemos clicado también se eliminará:


Cerrar sesión

---


Última conexión: 2023-12-29 22:41:08

id	título	genero	pais	año	cartel	4	5	6
1	El Señor de los Anillos	Fantasia	Estados Unidos	2001		<span style="background-color: red; color: white; padding: 2px 5px; border-radius: 3px;">Eliminar</span>	<span style="background-color: #ccc; color: #000; padding: 2px 5px; border-radius: 3px;">Modificar</span>	•


Elijah Wood




Scarlett Johansson




Brad Pitt



Jennifer Lawrence



---

3	Pulp Fiction	Crimen	Estados Unidos	1994		<span style="background-color: red; color: white; padding: 2px 5px; border-radius: 3px;">Eliminar</span>	<span style="background-color: #ccc; color: #000; padding: 2px 5px; border-radius: 3px;">Modificar</span>	•
---	--------------	--------	----------------	------	--	--	---	---

Si se clicla en el botón eliminar de la película que se encuentra arriba se borrarán las dos películas.  
No es necesario activar los checks para eliminar.

**AÑADIR PELÍCULA)**

### Añadir película ×

Título

Título

Género

Género

País

País

Año

Año

Cartel

Cartel

Cerrar

Enviar

Todos los parámetros han de completarse, el año ha de tener 4 cifras y el cartel ha de terminar en .jpg, en caso contrario se mostrará un error pero se guardarán los valores introducidos anteriormente.

NO ADMIN)

Enviar un mail:

id	username	
1	admin	<button>Enviar</button>
3	delegado	<button>Enviar</button>

Algún problema? contacta con nosotros

hola

Simplemente escribir un mensaje y seleccionar enviar al admin que se quiera enviar el mensaje.

Mensaje desde un cliente de videoClubTalavera

 **videoclubtalavera@gmail.com** 22:53 (hace 0 minutos)  
para delegado ▼  
hola

### PASO 3)

Cerrar sesión: Simplemente click en cerrar sesión.

Víctor Valdés Cobos.