Examen Desarrollo Web en Entorno Servidor

WatchVault es el nombre de mi aplicación. Un proyecto en el que un cliente al crear una cuenta podrá almacenar su peliculas y series con un estado: **vista, viendo, pendiente**.
Los usuario solo pueden acceder a la funcionalidad principal de nuestra aplicación si ha creado una cuenta.

En el documento se mostarán algunos ejemplos de todos los requisitos que han sido cubiertos y una pequeña explicación de forma técnica del ciclo de uso de la aplicación

Pablo Bartolomé Gallardo

Requisitos

- Aplicación que se conecte a una base de datos, hacer CRUD
- Formularios 3-4
- Uso de arrays asociativos
- Uso de cookies y sesiones
- Uso de get y post
- Entrada al sistema
- Validaciones cliente y servidor
- Mejoras:
  - Mas de una tabla
  - Mas formularios
  - MVC

Pablo Bartolomé Gallardo

Índice

Sobre el proyecto

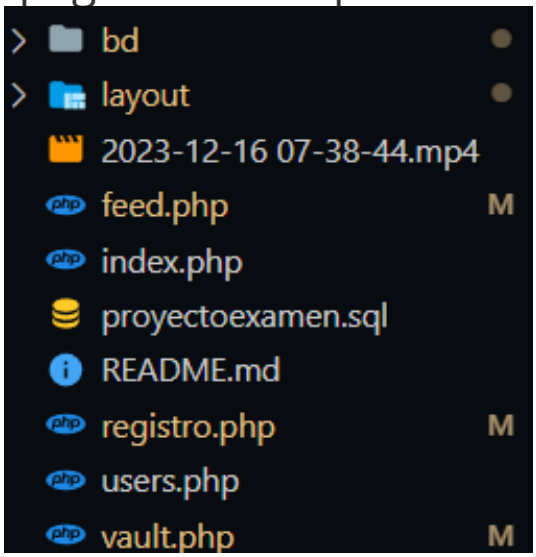
WatchVault se ha creado usando php y una base de datos mysql. He utilizando también la programación funcional para poder crear toda la lógica de mi aplicación.

Hay algo que se debe tener en cuenta para poder levantar el proyecto y visualizarlo de forma correcta.

Se debe abrir con cualquier editor o IDE de código que tenga acceso a una terminal y dentro de dicha terminal utilizar el siguiente comando en la raiz del proyecto:
"**php -S localhost:4455**"

¿Porqué usar este modo?
Este modo permite al usuario, en este caso al profesor ver todas las funcionalidades de forma correcta ya que uso rutas absolutas. Así que siempre que el usuario de la web pase de una web a otra no habrá problema alguno.

El proyecto está dividido de la siguiente manera:
en la carpeta BD está la conexión y todas las funcionalidades.
En la carpeta Layout tiene el menu y el footer de la web y las demás son las páginas de la aplicación



Conexión

```php
try {
    $conexion = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
    $conexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo '<p class="text-green-400">BD Conectada</p>';
} catch (PDOException $e) {
    // echo "Error de conexión: " . $e->getMessage();
    echo '<p class="text-red-400">Error de conexión</p>';
}
```

La conexión a la base de datos la he realizado en PDO. He usado esta forma porque al estar orientada a objetos puedo crear un TryCatch en las funciones que uso y capturar el error para mostrarselo al usuario. En la conexión de la base de datos vemos que estoy mostrando un párrafo en HTML dependiendo de si la conexión ha sido correcta o no

Formularios

He creado varios formularios, cada uno relacionado con una funcionalidad. Ejemplo:

Formulario para eliminar

El formulario NO actualiza. Lo que hace es cambiar una variable a true para mostrar el formulario correcta para modificar los datos **- vault.php**

```php
<form action="" method="POST">
    <button type="submit" name="openmodalmodificar" class="text-base mr-2" title="Modificar">
    <input type="hidden" name="idAudiovisual" value="' . $audiovisual['id'] . '">
    <input type="hidden" name="idUsuario" value="' . $_SESSION['id'] . '">
    <input type="hidden" name="tipo" value="' . $audiovisual['tipo'] . '">
    <input type="hidden" name="nombre" value="' . htmlspecialchars($audiovisual['nombre']) . '">
    <input type="hidden" name="descripcion" value="' . htmlspecialchars($audiovisual['descripcion']) . '">
    <input type="hidden" name="estado" value="' . $audiovisual['estado'] . '">
        <svg
            xmlns="http://www.w3.org/2000/svg"
            width="30"
            height="30"
            viewBox="0 0 24 24"
        >
            <path
                fill="#9ca3af"
                d="M3 21v-4.25L16.2 3.575q.3-.275.663-.425t.762-.15q.4 0 .775.15t.65.45L20.425 5q.3.275.438
            />
        </svg>
    </button>
    <button type="submit" name="eliminaraudiovisual" class="text-base mr-2" title="Eliminar">
        <input type="hidden" name="idAudiovisual" value="' . $audiovisual['id'] . '">
        <input type="hidden" name="idUsuario" value="' . $_SESSION['id'] . '">
```

Formulario para mostrar el modal - vault.php

```php
<form action="" method="POST">
    <button type="submit" name="openmodalaudiovisual" class="crear text-base bg-white rounded-full mt-3">
        <input type="hidden" name="accion" value="agregar">

        <svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" viewBox="0 0 24 24">
            <path fill="#000000" d="M11 13H5v-2h6V5h2v6h6v2h-6v6Z" />
        </svg>
    </button>
</form>
```

Formulario de FILTROS - vault.php

```php
<form action="" method="POST" class="flex flex-row gap-2 justify-around">
    <select name="tipofiltro" class="px-1 rounded font-bold text-white bg-gray-800">
        <option value="" <?php echo ($tipoFiltro === null) ? 'selected' : ''; ?> disabled>-Tipo-</option>
        <option value="serie" <?php echo ($tipoFiltro === 'serie') ? 'selected' : ''; ?>>Serie</option>
        <option value="pelicula" <?php echo ($tipoFiltro === 'pelicula') ? 'selected' : ''; ?>>Pelicula</option>
    </select>

    <select name="estadofiltro" class="px-1 rounded font-bold text-white bg-gray-800">
        <option value="" <?php echo ($estadofiltro === null) ? 'selected' : ''; ?> disabled>-Estado-</option>
        <option value="vista" <?php echo ($estadofiltro === 'vista') ? 'selected' : ''; ?> class="hover:bg-red-400">Vista</option>
        <option value="viendo" <?php echo ($estadofiltro === 'viendo') ? 'selected' : ''; ?> class="hover:bg-red-400">Viendo</opti
        <option value="pendiente" <?php echo ($estadofiltro === 'pendiente') ? 'selected' : ''; ?> class="hover:bg-red-400">Pendie
    </select>
    <button type="submit" name="busquedafiltro"><svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" viewBox="0 0 256 25
        <path fill="#9ca3af" d="M168 112a56 56 0 1 1-56-56a56 56 0 0 1 56 56Zm61.66 117.66a8 8 0 0 1-11.32 01-50.06-50.07a88
    </svg></button>
    <button type="submit" name="resetfiltro"><svg xmlns="http://www.w3.org/2000/svg" width="30" height="30" viewBox="0 0 32 32">
        <path fill="#9ca3af" d="M18 28A12 12 0 1 0 6 16v6.2l-3.6-3.6L1 2016 6l6-6l-1.4-1.4L8 22.2V16a10 10 0 1 1 10 10Z" />
    </svg></button>
</form>
```

Formulario para crear audiovisual

Este formulario se muestra al cliente cuando una variable sea TRUE y así pueda el usuario rellenar los datos. Del mismo modo existe también el formulario de modificar. **- vault.php**

```php
<form action="" method="POST">
    <div class="flex flex-col justify-center gap-3">
        <label for="" class="text-gray-300 font-bold">Tipo:</label>
        <div class="flex flex-row gap-4">
            <div class="flex flex-row items-center gap-2">
                <label for="">Pelicula</label>
                <input type="radio" name="tipo" value="pelicula" class="bg-gray-400 text-gray-800">
            </div>
            <div class="flex flex-row items-center  gap-2">
                <label for="">Serie</label>
                <input type="radio" name="tipo" value="serie" class="bg-gray-400 text-gray-800">
            </div>
        </div>
        <div class="flex flex-row gap-2">
            <label for="" class="text-gray-300 font-bold">Nombre</label>
            <input type="text" name="nombre" class="bg-gray-400 text-gray-800">
        </div>
        <div class="flex flex-row gap-2">
            <label for="" class="text-gray-300 font-bold">Descripción</label>
            <textarea name="descripcion" id="" cols="25" rows="4"  class="bg-gray-400 text-gray-800"></textarea>
        </div>
        <div class="flex flex-row gap-2">
            <label for="" class="text-gray-300 font-bold">Estado:</label>
            <select name="estado" id="" class="bg-gray-400">
                <option value="" disabled selected>-Estado-</option>
                <option value="vista">Vista</option>
                <option value="viendo">Viendo</option>
                <option value="pendiente">Pendiente</option>
            </select>
        </div>
        <button type="submit" name="agregaraudiovisual" class="bg-gray-600 rounded-lg hover:bg-gray-500 hover:">Agregar</button>
```

Formulario para modificar - vault.php

```php
<form action="" method="POST">
    <div class="flex flex-col justify-center gap-3">
        <label for="" class="text-gray-300 font-bold">Tipo:</label>
        <div class="flex flex-row gap-4">
            <div class="flex flex-row items-center gap-2">
                <label for="">Pelicula</label>
                <input type="radio" name="tipo" value="pelicula" ' . ($tipo  === 'pelicula' ? 'checked' : '') . ' class="bg-gray-400
            </div>
            <div class="flex flex-row items-center  gap-2">
                <label for="">Serie</label>
                <input type="radio" name="tipo" value="serie" ' . ($tipo  === 'serie' ? 'checked' : '') . ' class="bg-gray-400 text-
            </div>
        </div>
        <div class="flex flex-row gap-2">
            <label for="" class="text-gray-300 font-bold">Nombre</label>
            <input type="text" name="nombre" value="' . $nombre . '" class="bg-gray-400 text-gray-800">
        </div>
        <div class="flex flex-row gap-2">
            <label for="" class="text-gray-300 font-bold">Descripción</label>
            <textarea name="descripcion" id=""  cols="25" rows="4" class="bg-gray-400 text-gray-800">' . $descripcion . '</textarea>
        </div>
        <div class="flex flex-row gap-2">
            <label for="" class="text-gray-300 font-bold">Estado:</label>
            <select name="estado" id="" class="bg-gray-400">
                <option value="" disabled selected>-Estado-</option>
                <option value="vista" ' . ($estado === 'vista' ? 'selected' : '') . '>Vista</option>
                <option value="viendo" ' . ($estado === 'viendo' ? 'selected' : '') . '>Viendo</option>
                <option value="pendiente" ' . ($estado === 'pendiente' ? 'selected' : '') . '>Pendiente</option>
            </select>
        </div>
        <button type="submit" name="modificaraudiovisual" class="bg-gray-600 rounded-lg hover:bg-gray-500 hover:">Modificar</button>
        <input type="hidden" name="idAudiovisual" value="' . $idAudiovisual . '">
```

Formulario para cambiar estilo y controlar la Cookie - header.php

```php
<form action="" method="POST">
    <!--Uso el resultado de la cookie para el titulo Modo Pablo o Modo Sol -->
    <button type="submit" name="cambiarsestilo" title="<?= ($_COOKIE['estilo'] == 'sol') ? 'Modo Pablo' : 'Modo Sol' ?>">
        <?= ($_COOKIE['estilo'] == 'sol') ? '<svg xmlns="http://www.w3.org/2000/svg" width="40" height="40" viewBox="0 0 25
    </button>
</form>
```

Arrays, Peticiones, Cookies y Sesiones

He usado 3 arrays asociativos super globales:
$_POST, $_GET, $_SESSION
Que podemos ver como ejemplo en las siguientes imágenes:

Podemos ver que pregunto si la petición $_POST llamada registro usuario existe, si la condición es verdadera guardo los datos del formulario en variables y más tarde ejecuto una función **- bd.php**

```php
//Registro Usuario
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['registrousuario'])) {
    $nombre = $_POST['nombre'];
    $email = $_POST['email'];
    $password = $_POST['password'];

    registrarUsuario($conexion, $nombre, $email, $password);
}

function registrarUsuario($conexion, $nombre, $email, $password)
{
```

Aquí vemos el primer punto donde usamos la petición $_POST en el formulario **- registro.php**

```php
<form action='' method='POST'>
<div class='flex flex-col justify-center gap-
<label for='' class='text-gray-300 font-bold
<input type='text' name='nombre' class='mb-2

<label for='' class='text-gray-300 font-bold
<input type="mail" name='email' class='mb-2 

<label for='' class='text-gray-300 font-bold
<input type='password' name='password' class=

<button type='submit' name='registrousuario'
</div>
</form>
```

En la petición **$_GET** pregunto si el valor del array asociativo es true o false en un registro para mostrar un párrafo HTML el cual ha sido declarado en un TryCatch cuando ocurre un error en la manipulación de la BD:

- registro.php

```php
if (isset($_GET['errorformregistro']) && $_GET['errorformregistro'] === 'true') {
    echo "<p class='text-red-500 font-bold text-center text-base'>Campos vacios</p>";
}
```

En esta función por ejemplo utilizo varios **$_GET** almacenando su clave y valor para poder mostrar en una página un mensaje dependiendo del valor del **$_GET - bd.php**

```php
if (empty($nombre) || empty($email) || empty($password)) {
    /*si hay un campo vacio enviamos al usuario a la página r
    con el array asociativo GET lleno con ese dato para valor
    header('Location: registro.php?errorformregistro=true');
    exit;
} else {
    try {
        $query = "INSERT INTO usuarios (nombre , email , cont
        $statement = $conexion->prepare($query);
        $statement->bindParam(':nombre', $nombre, PDO::PARAM_
        $statement->bindParam(':email', $email, PDO::PARAM_ST
        $statement->bindParam(':contrasena', $password, PDO::
        $statement->execute();
        header('Location: registro.php?successregistro=true');
    } catch (PDOException $e) {
        header('Location: registro.php?errordb=true');
    }
}
```

En este caso mi array superglobal **$_SESSION** lo inicio cuando la base de datos me responde un dato existente comparando usuario y contraseña de alguien que quiere hacer un registro. Si la base de datos me responde su propio dato lleno mi **$_SESSION** con un ID y con su nombre de usuario para mostrarlo en mi web **- bd.php**

```php
if ($usuario) {
    // Usuario autenticado, establecer la sesión
    session_start();
    $_SESSION['usuario'] = $usuario['nombre'];
    $_SESSION['id'] = $usuario['id'];

    header('Location: ../index.php?success=true');
    exit;
} else {
    // Credenciales inválidas, mostrar mensaje de error
    header('Location: ../index.php?errorcred=true');
    exit;
}
```

Aquí pregunto si **$_SESSION['usuario']** está vacio, si no lo está el dato que muestra es el valor de **$_SESSION**. Si no, muestro el nombre **Invitado**
- header.php

```php
<p class="font-bold <?php echo isset($_SESSION['usuario']) ? 'text-green-400' : ''; ?>">
    <!--Aquí verifico que está la sesión iniciada o no para añadir el nombre del usuario -->
    <?php echo isset($_SESSION['usuario']) ? $_SESSION['usuario'] : 'Invitado'; ?>
</p>
```

Aquí uso el resultado de **$_SESSION['id']** para usar el ID del usuario loggeado para introducir datos en relación al usuario y su **ID**
- bd.php

```php
try {
    $query = "INSERT INTO audiovisual (idUsuario, tipo, nombre, descripcion,estado) VALUES (:idU
    $statement = $conexion->prepare($query);
    $statement->bindParam(':idUsuario', $_SESSION['id'], PDO::PARAM_INT);
```

He usado las **Cookies** solo para fines visuales por los siguientes motivos:
1- El usuario podría configurar el navegador sin uso de cookie
2- No quiero usar datos de mi DB ya que es algo inseguro

Lo promero que debemos ver es nuestro documento **styles.css**

```css
body.sol{
  background-color: #180726;
}
tr.sol{
  background-color: #561a45;
}
```

Tenemos unos estilos declarados para la clase "**SOL**"
Al comienzo de mi archivo bd.php pregunto **NO** existe un valor en la clave **$_COOKIE['estilo']** si es verdadero, creo una cookie con esa clave y con el valor '**PABLO**' **- bd.php**

```php
if (!isset($_COOKIE['estilo'])) {
    setcookie('estilo', 'pablo');
}
```

De esa forma pregunto por el valor de que tiene **$_COOKIE['estilo']**. Si el valor es SOL, añade **class="sol"** si es falso, no hace nada aunque podría añadir la clase 'pablo'. Al no existir estilos en mi CSS no habría diferencia. También podemos ver un metodo POST con un envio a "**cambiarestilo**" **- header.php**

```php
<body class="<?= ($_COOKIE['estilo'] == 'sol') ? 'sol' : '' ?>">
    <div class="flex flex-row justify-end items-center mt-5 mr-10">
        <div>
            <form action="" method="POST">
                <!--Uso el resultado de la cookie para el titulo Modo Pablo o Modo Sol -->
                <button type="submit" name="cambiarsestilo" title="<?= ($_COOKIE['estilo'] == 'sol') ? 'Modo Pablo' : 'Modo Sol' ?>">
                    <?= ($_COOKIE['estilo'] == 'sol') ? '<svg xmlns="http://www.w3.org/2000/svg" width="40" height="40" viewBox="0 0 2
                </button>
            </form>
```

En la petición POST si el estilo resultado de la cookie es 'pablo' cambia a 'sol' para añadir los estilos y alreves. Con una duración de 5000ms **- bd.php**

```php
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['cambiarsestilo'])) {
    $nuevoEstilo = ($_COOKIE['estilo'] == 'pablo') ? 'sol' : 'pablo';
    setcookie('estilo', $nuevoEstilo, time() + 5000);
    header("Location: index.php");
}
```

Ciclo de vida

Lo primero que podemos ver al entrar en la aplicación es si se ha realizado la conexión a la base de datos de forma exitosa y o no. También se mostrará un menú dinámico que cambiará dependiendo de si existe un usuario logueado o no. En cuyo caso no lo esté aparecerá como Invitado. Si el usuario está logueado la web aparecerán con 2 distintivos notables:

1- Tendrá habilitada la opción de **Vault** la cual se encuentra en el menú de color verde
2- Tiene en el menú su nombre haciendo referencia que existe una persistencia de datos

El usuario no logueado puede acceder a todas las páginas de la web menos **Vault.** Pudiendo acceder también a la página **users.php y feed.php**

En **feed.php** podrá ver todos los comentarios de los usuarios pero al no estar logueado no podrá añadir ningún comentario.

Cuando el usuario se haya registrado podrá crear comentarios en la página **feed.php**

Cuando el usuario se quiera registrar debe ir a Login para abrir el formulario de acceso y acceder al botón de registro.

Cuando haya creado el registro podrá hacer login
Una vez logueado podrá guardar sus propias peliculas y series en un formulario que a parece dentro de la página **vault.php** pulsando el botón **+** Podrá añadir peliculas o series son su nombre, una descripción a elegir y el estado de su pelicula / serie.

El usuario también puede filtrar su contenido si solo quisiera ver sus peliculas o series o incluso filtrar por el estado de las mismas. Ambos usuarios, logueados o no. Pueden cambiar el diseño de la aplicación web accediendo al botón derecho superior en forma de sol de color morado. Cuando seleccione ese botón automáticamente la web cambiará el diseño y se le llevará al menú de inicio de la aplicación.

Mejoras

Siendo consciente de la magnitud de la aplicación. Para poder crear una escalabilidad correcta se debería aplicar un **modelo-vista-controlador**. Ya que dicho modelo es ágil para aplicaciones medianas y grandes. Pero para proyectos pequeños crear un modelo desde 0 dificulta mucho el desarrollo. Se habla teniendo en cuenta que se crearía lo que se conoce como un "**monolito**". Por lo tanto he decidido no aplicarlo por falta de tiempo y no hacer sobreesfuerzo en la aplicación.
Esto implica que si quisiera aplicarlo. No dudaría en utilizar un framework a ser posible con **ORM** el cual me ayude a gestionar la base de datos con sus respectivos modelos y no lanzar consultas preparadas, si no que el propio **ORM** pueda gestionarlas de forma más seguras, además de poder añadir seguridad en el retorno de funciones que muestren datos completos o simplemente en un hasheo de contraseñas mas robusto y actual. Ya que los métodos de hasheo de php no están actualizados ya que usan términos y técnicas de hasheo antiguos. Es mejor utilizar librerías externas que están actualizadas.

Web y registro

La creación de la aplicación se ha creado con la intención de registrar todo el posible proceso de la misma. Se ha creado desde las fechas mostradas en los **commits** de git para su futura puesta a producción.
Por lo tanto la web ha sido subida a producción en y está en el repositorio de forma pública en las fechas viables al examen para que el profesor tenga acceso a las mismas haciendo click.

Repositorio GitHub:
https://github.com/barthxc/Examen-Php-Proyecto

Web en Producción:
https://watchvaultexamen.000webhostapp.com