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.

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

Explicación punto por punto:

1. **Login**

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

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

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

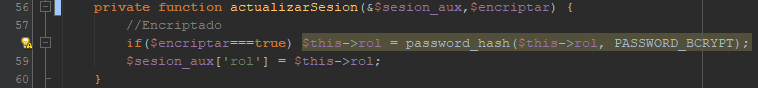
Forma

Descripción generada automáticamente con confianza baja

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.



Y de esta forma compruebo el 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().

1. **Funciones del usuario administrador:**
2. **Ver una lista con todas las películas junto a los controles ‘Modificar’ y ‘Borrar’:**

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

1. **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: ‘eliminarPelicula\_’ y el id de la misma:



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:

Texto

Descripción generada automáticamente

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



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 funcionalidadPeliculas() 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 eliminarPelicula 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)

Texto

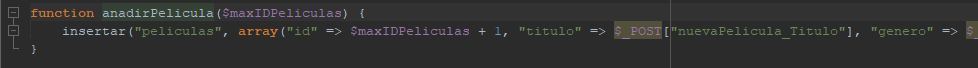
Descripción generada automáticamente

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

1. **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 anadirPelicula que a su vez llama a la función insertar que modifica la base de datos:



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, asique viene dada por un parámetro e/s de la función crearInstanciasPelicula():



1. **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

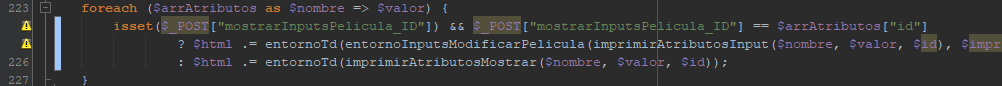
gestionarFuncionalidad llama a modificarPelícula y como todavía nos encontramos en el primer paso solo hace esto:

Texto

Descripción generada automáticamente

Inicia esta instancia en POST.

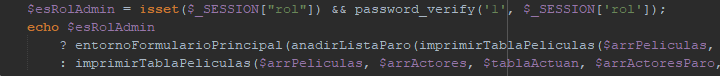
Esta instancia es interpretada por la función que imprime la tabla y esta llama a la función que imprime inputs en la fila que corresponde al ID:



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

1. **Funciones del usuario normal**
2. **Mostrar una lista de películas**

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

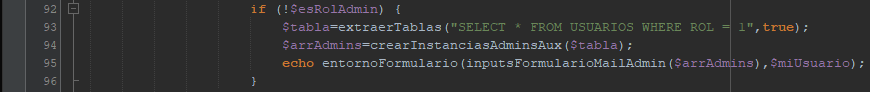


Debido a que solamente se ha de mostrar una lista, no necesito formulario para la tabla películas ni necesito la lista de actores en paro. Además en imprimirTablaPelículas hay un parámetro que imprime o no los botones modificar y modificar ($tablaNoFuncional):

Texto

Descripción generada automáticamente

1. **FORMULARIO PARA ESCRIBIR MAIL A ADMINISTRADOR A ELEGIR**



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:

Texto

Descripción generada automáticamente

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:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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:



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:



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

Texto

Descripción generada automáticamente

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)



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

Esto puede desembocar ciertas opciones:

Texto

Descripción generada automáticamente

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 ):



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:

Texto

Descripción generada automáticamente

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

**Log:**

Texto

Descripción generada automáticamente

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 erroneo:



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

Texto

Descripción generada automáticamente

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

Tabla

Descripción generada automáticamente

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

Texto

Descripción generada automáticamente

**Cookies)**



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 imprimirListaPeliculas, 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.

Captura de pantalla de un celular

Descripción generada automáticamente

He intentado reutilizar todas las funciones posibles, uso imprimirTablaPeliculas también para imprimir actores en paro, estos actores en paro 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 par, con esto obtengo las instancias necesarias del objeto Actor para usarlas en la función imprimirTablaPeliculas.

Captura de pantalla de un videojuego

Descripción generada automáticamente

Explicación y notas)

Organización:

Texto

Descripción generada automáticamente

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 deconocía antes de empezar y la causa de gran parte de los warnings del código.

**OTRAS NOTAS:**

En homepage.php:

Texto

Descripción generada automáticamente

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 cpnstructor 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 funcionalidadPeliculas uso esta función nativa: 

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.

Texto

Descripción generada automáticamente

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 solicionarlo 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)](https://www.lawebdelprogramador.com/foros/CSS/1643487-No-se-actualiza-CSS.html)

