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:

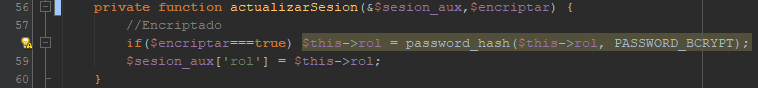
Forma

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

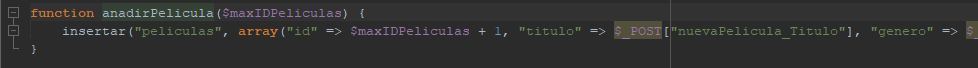
Descripción generada automáticamente

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

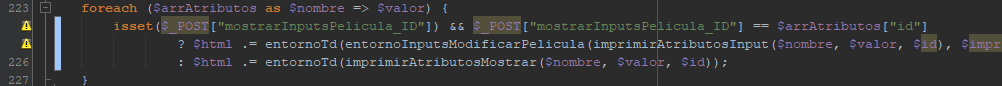
gestionarFuncionalidad llama a modificarPelícula y como todavía nos encontramos en el primer paso solo hace esto:

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.