

Práctica 2.

Programación Web dinámica de un sistema de venta de DVDs

ORGANIZACIÓN DEL PROYECTO

> Listado de html's contenidos en carpeta 'templates':

- index.html
- pelicula.html
- sesion.html
- registro.html
- micuenta.html
- carrito.html
- privacidad.html

> Información de las películas:

- catalogo.json

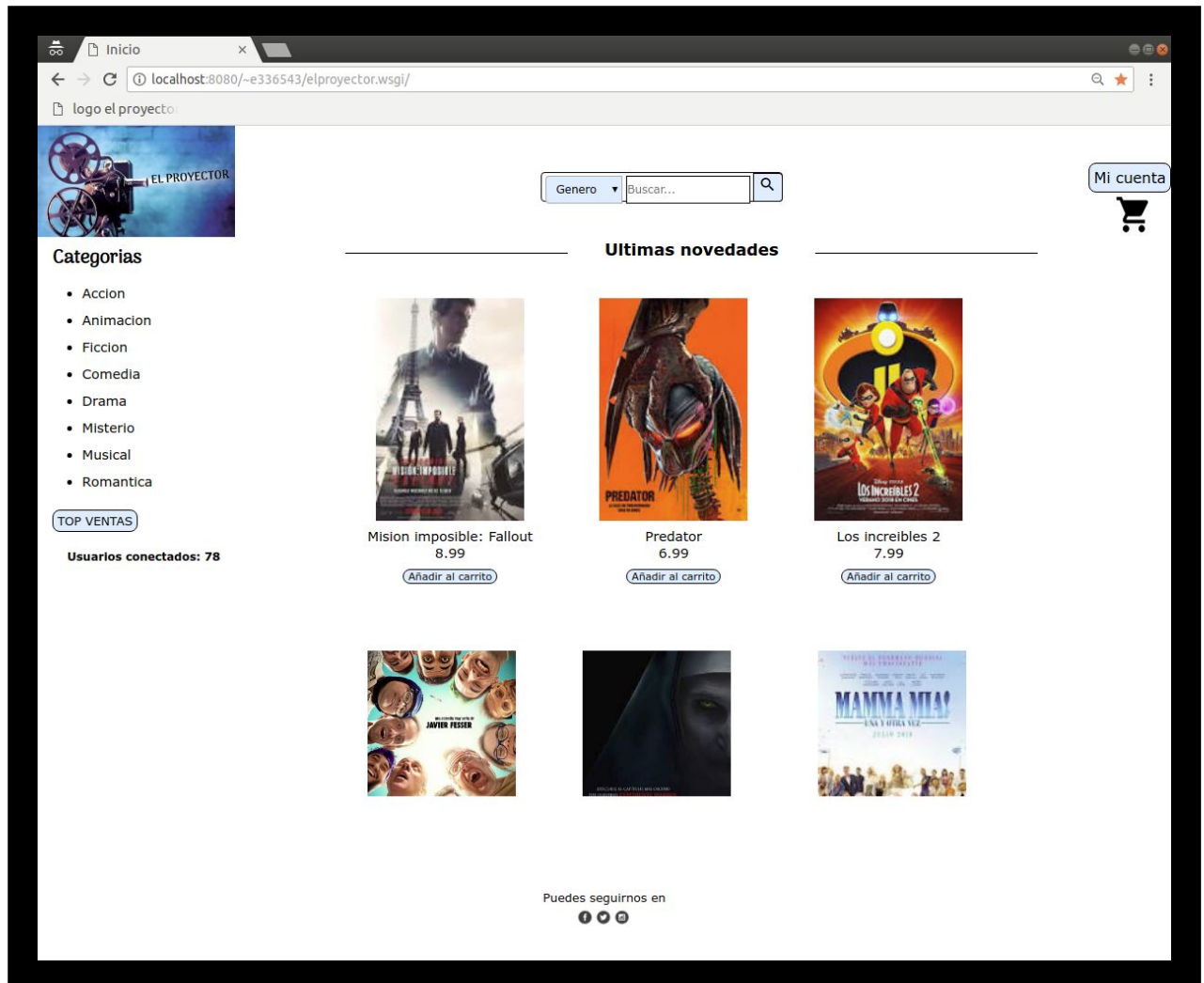
> Información de usuario en carpeta 'usuarios': existe una carpeta con cada nombre de usuario registrado (email hasta @):

- datos.dat : información relativa a email, contraseña, saldo...
- historial.dat: información relativa a las compras

> Administrador de peticiones al servidor:

- manage.py : todas las rutas y funcionalidades que se solicitan durante la ejecución.

PÁGINA PRINCIPAL



url = <http://localhost:8080/~usuario/elproyector.wsgi/>

html = index.html'

Desde la página principal se puede acceder a la información de cada película mostrada, así como ver las películas correspondientes a un género, las últimas novedades y el top ventas. También se permite buscar una película introduciendo su nombre y el género al que pertenece. En el caso de introducir únicamente el género, actuará igual que el menú de categorías y en el caso de introducir únicamente un nombre, buscará la película con dicho título. Dicha funcionalidad se ha logrado gracias al uso de Flask y los datos cargados a partir de catalogo.json.

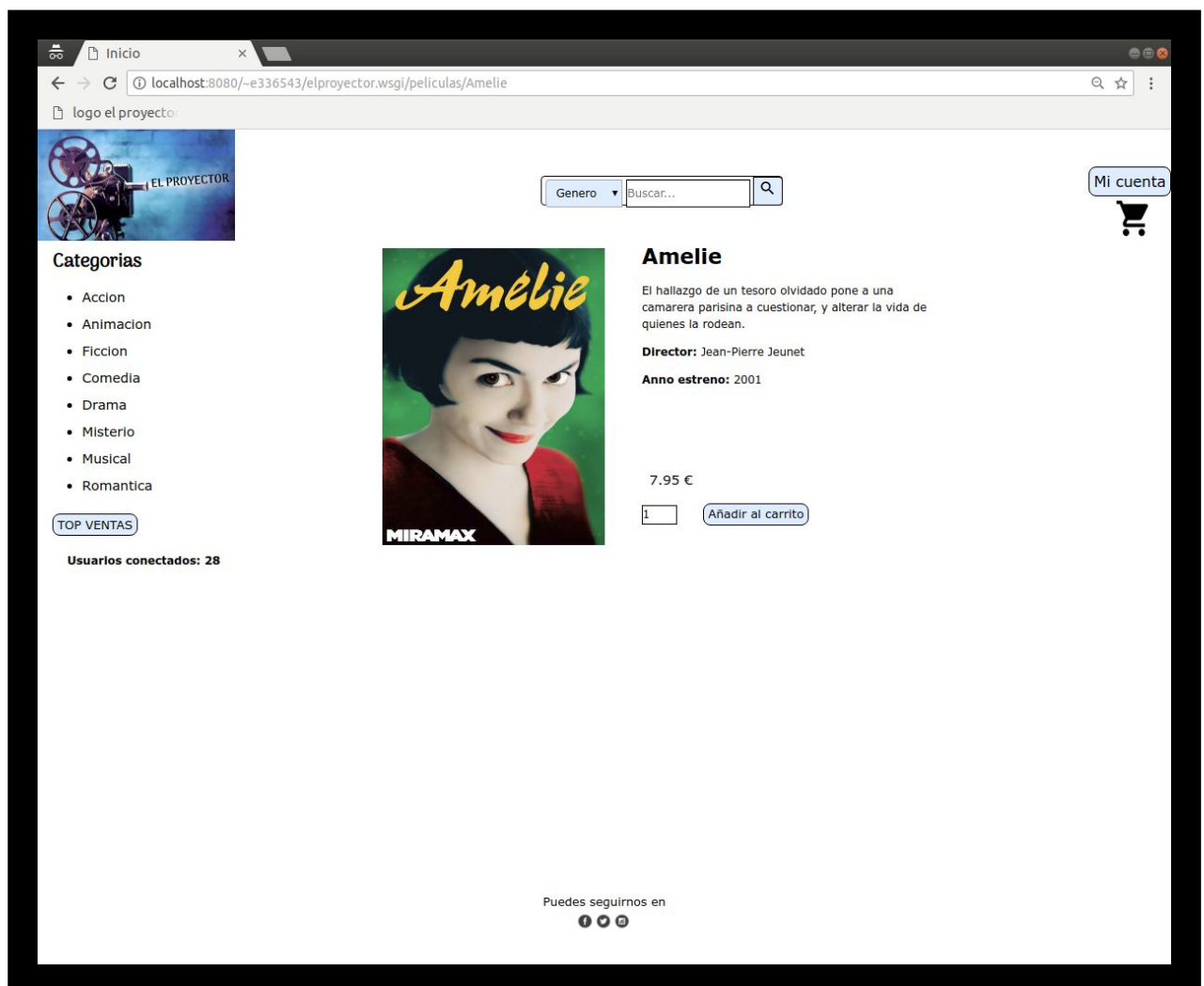
También se puede añadir al carrito las películas que se deseen adquirir gracias al uso de Ajax en javascript y Flask. Hemos considerado útil utilizar Ajax para no tener que cargar el contenido de la página ppal de nuevo, ya que es más eficiente de esta forma. Dicha información se guarda en una clave de la variable session de Flask session['carrito'], que es una lista de diccionarios con la información de la película y su número de unidades. También hacemos uso de una variable session['precio'], que va sumando el precio de cada película añadida teniendo en cuenta el número de

unidades. Gracias a session podemos mantener esta información se haya iniciado o no sesión por parte del usuario.

Desde la página principal también se puede acceder a la página de iniciar sesión o registrarse o a la página del historial de un usuario en el caso de que ya se haya logueado, utilizando Flask.

Por otra parte, tanto en esta página como en el resto, se puede ver en el menú los usuarios conectados. Esto se ha conseguido llamando en el body de la plantilla a una función de Javascript que utiliza Ajax+python para crear un número aleatorio y modificarlo cada 3s.

CONTENIDO PELÍCULA



url = <http://localhost:8080/~usuario/elproyector.wsgi/peliculas/Amelie>

html = pelicula.html

Se puede ver la información relativa a esta gracias al uso de Flask+catalogo.json. También se pueden añadir al carrito la cantidad de unidades que desees de dicha película, utilizando nuevamente Flask+Ajax, al igual que en la página ppal por eficiencia, y accediendo de la misma forma a la variable session de Flask.

También se puede acceder al resto de elementos comentados en la página principal propios del menú y la cabecera. (Funcionalidad de todas las páginas).

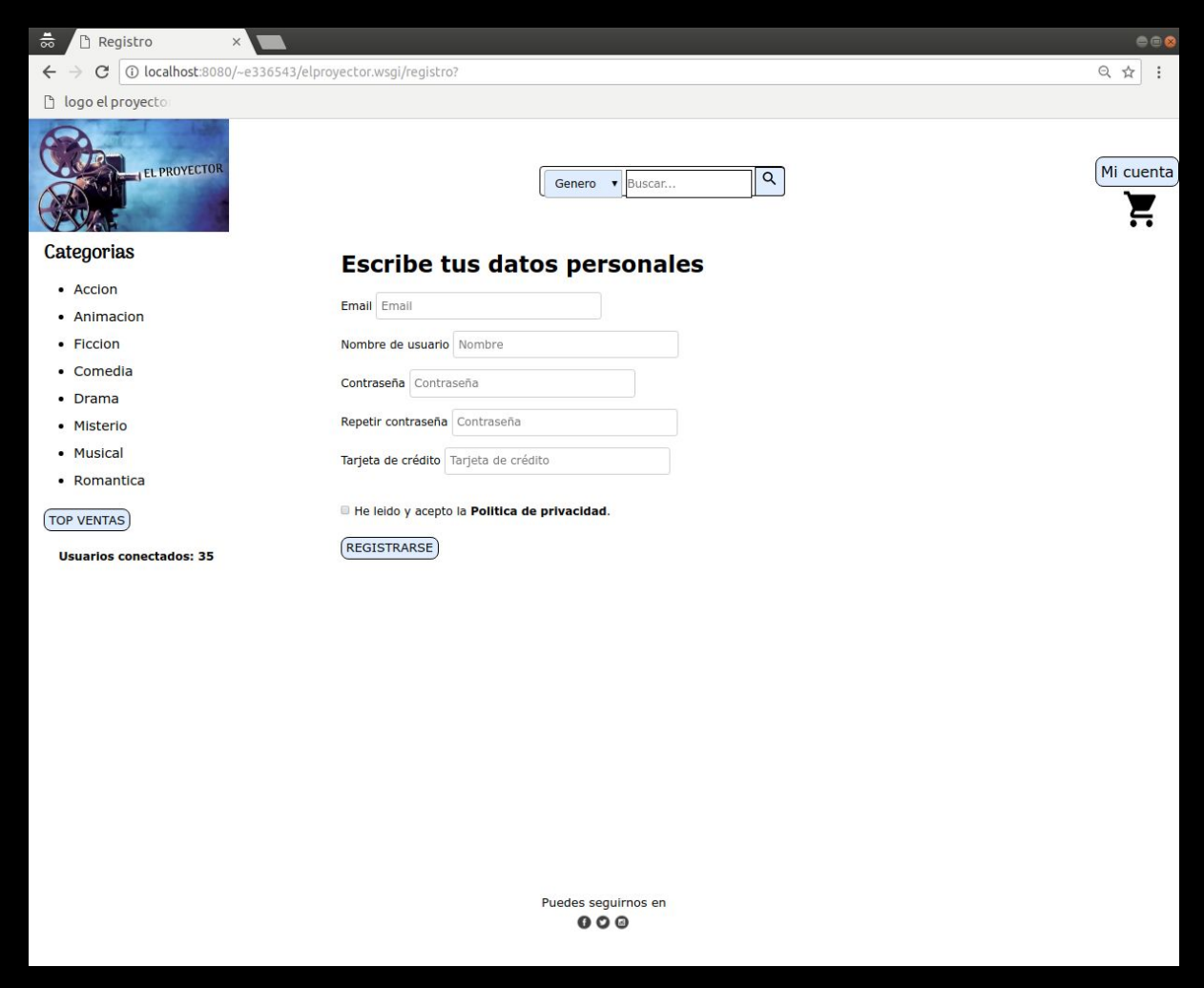
INICIO SESION

url = <http://localhost:8080/~usuario/elproyector.wsgi/micuenta?>

html = sesion.html

Pulsando en el botón de mi cuenta sin haber iniciado sesión, accedemos a esta pantalla donde se nos da la opción de iniciar sesión si ya estamos registrados o crear una nueva cuenta. En el caso de haber iniciado sesión previamente, el usuario aparecerá por defecto por el uso de las cookies. Se comprobará antes de iniciar sesión que el usuario. Antes de iniciar sesión, se comprobará mediante Javascript que el formato del usuario es correcto (email válido) y que no hay campos vacíos. Una vez se ha comprobado esto, se utilizará AJAX para comprobar que el usuario existe y que la contraseña introducida es correcta. Si estas condiciones no se satisfacen, aparecerá un aviso indicándolo. Al iniciar sesión guardamos el nombre de usuario en la variable session de flask para poder acceder a sus datos y su historial de manera global. Ese dato lo utilizaremos para fijar las cookies correspondientes. Si no se tiene una cuenta, pulsamos en 'Crear Cuenta' y se procede al registro.

REGISTRO DE USUARIO



The screenshot shows a web browser window with the address bar displaying `localhost:8080/~e336543/elproyector.wsgi/registro?`. The page title is "Registro". The header includes a logo for "EL PROYECTOR" (a film reel) and a search bar with a "Genero" dropdown and a "Buscar..." input field. A "Mi cuenta" button with a shopping cart icon is in the top right.

Categorías

- Accion
- Animacion
- Ficción
- Comedia
- Drama
- Misterio
- Musical
- Romantica

TOP VENTAS

Usuarios conectados: 35

Escribe tus datos personales

Email

Nombre de usuario

Contraseña

Repetir contraseña

Tarjeta de crédito

☐ He leído y acepto la **Política de privacidad**.

Puedes seguirnos en

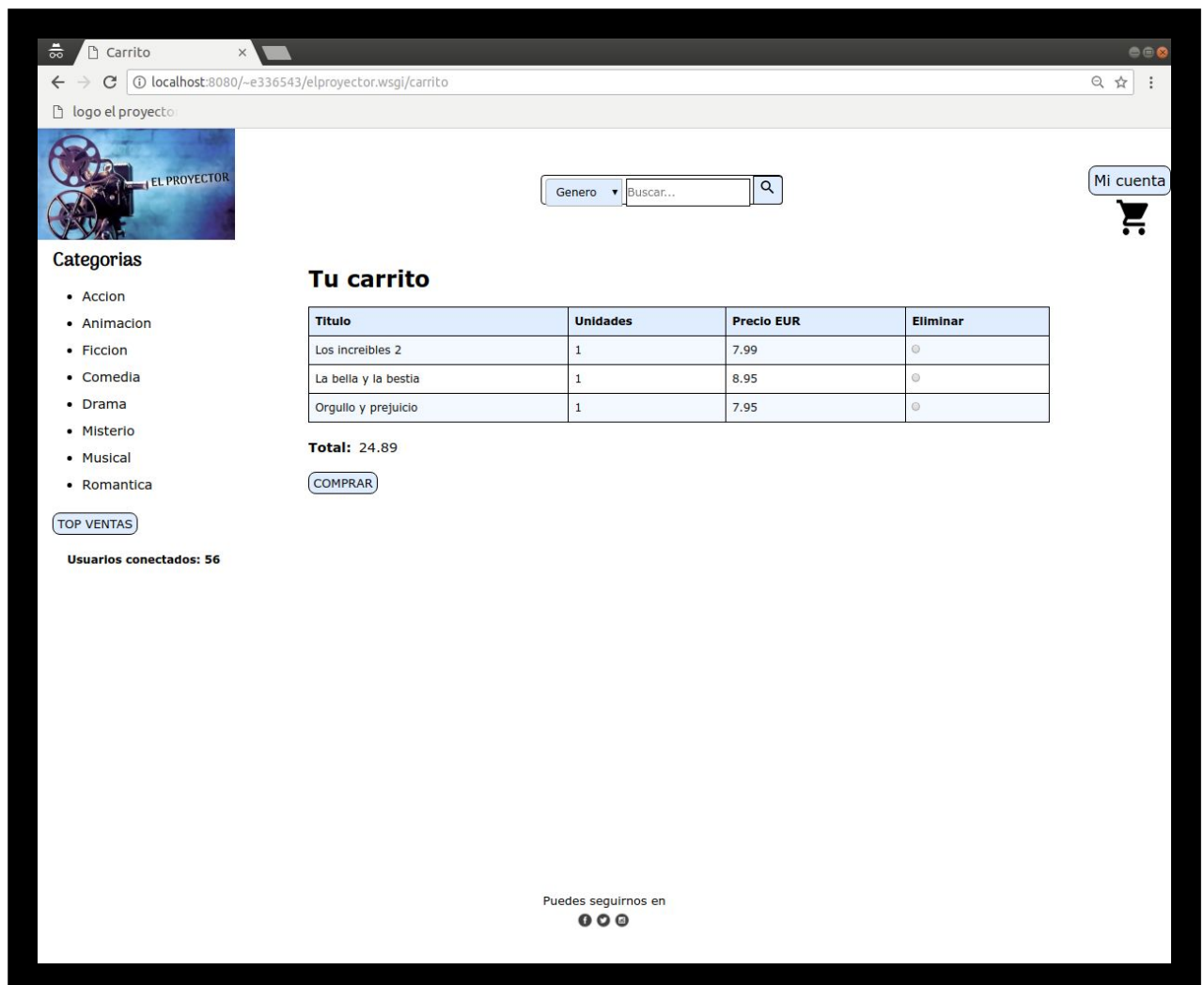
Facebook, Twitter, YouTube icons

url = <http://localhost:8080/~e336543/elproyector.wsgi/registro?>

html = registro.html

Si no se tiene una cuenta, se procede al registro. Accedemos al formulario de la pantalla donde se tienen que introducir los datos que se solicitan. Mediante un Javascript se controla el formato de los datos introducidos avisando de todos los posibles errores con alerts. También se ha añadido un fragmento de código que va clasificando la contraseña introducida en 3 tipos diferentes de seguridad en función de los caracteres que vamos introduciendo. Para implementar esta funcionalidad nos hemos basado en diversos ejemplos de internet. Será necesario aceptar la política de privacidad para poder proceder a registro. Si todo es correcto, se comprobará con AJAX que el correo está disponible. En caso de registrarse, se iniciará la sesión de flask con el nuevo usuario y se procederá a configurar las cookies.

CARRITO



url = <http://localhost:8080/~e336543/elproyector.wsgi/carrito>
html = carrito.html

Desde la página de carrito, se carga la información guardada en la variable `session['carrito']` y `session['precio']` ya comentadas gracias al uso de Flask. Se mostrará una tabla con las diferentes películas seleccionadas para comprar, las cantidades de cada una y el precio a pagar por cada película. También se podrá eliminar una película pulsando en el botón de la fila correspondiente. Al borrar, se elimina de `session['carrito']` la información relativa a esa película y se actualiza `session['precio']`. Mediante AJAX se actualiza esta información y desaparece la fila correspondiente. En el caso de que `session['precio']` se quede a 0 recargamos la página para que aparezca el mensaje de que el carrito está vacío. Al pulsar en comprar, se comprueba que hay una sesión iniciada y en caso contrario se procede a iniciar sesión o registrarse. El carrito se mantiene ya que estaba almacenado en la variable `session['carrito']`. Se comprobará también que el saldo del usuario de la sesión tiene saldo suficiente para efectuar la compra. Si todo es correcto, se guarda en el historial la compra realizada, se resta del saldo del usuario el dinero

correspondiente, se vacía la variable session['carrito'] y la variable session['precio'] se cambia a 0. ¹

HISTORIAL

Inicio x

localhost:8080/~e336543/elproyector.wsgi/micuenta?

logo el proyector

Genero Buscar...

Mi cuenta

Categorías

- Accion
- Animacion
- Ficcion
- Comedia
- Drama
- Misterio
- Musical
- Romantica

TOP VENTAS

Informacion Personal

Nombre: Prueba
Correo: prueba@gmail.com
Saldo: 42.05

Mi Historial

Id	Fecha de compra	Precio EUR
1	31/10/18	16.95

Titulo	Cantidad	Precio EUR
Intocable	1	8
El gran showman	1	8.95

CERRAR SESION

Puedes seguirnos en

1

url = <http://localhost:8080/~usuario/elproyector.wsgi/micuenta?>

html = micuenta.html

En esta página se muestra tanto la información personal del usuario obtenida del fichero datos.dat como el historial de un usuario logueado. Este historial es una tabla en la que se distinguen las compras por un id, una fecha de compra y un precio total. Todo ello se ha obtenido gracias al fichero historial.json de cada usuario. Este fichero consta de una lista de diccionarios caracterizados por un id, fecha, precio y pelis, que es una lista de pares peli y unidades, como se ha comentado con anterioridad. Para acceder a la información relativa a la compra se ha empleado JQuery, de forma que al seleccionar una determinada fila se muestra otra tabla con el resto de información gracias al uso de la función my_toggle que es una modificación de la función toggle.

Por último, si se quiere cerrar sesión se modifica la variable `session['logged_in']` a `False` y se redirige la página actual a la principal mediante Flask.

REFERENCIAS

Las fuentes de código que hayan servido de inspiración para la elaboración de la práctica.

Para la elaboración de la práctica, nos hemos ayudado de la documentación de Flask, JQuery, JavaScript y Ajax proporcionada por páginas como:

- <https://www.tutorialspoint.com/>
- <https://www.w3schools.com/>

Algunas funcionalidades específicas como la clasificación de la contraseña en función de la seguridad de la misma o expresiones regulares para ciertos campos de formularios, las hemos obtenido de ejemplos de <https://es.stackoverflow.com/>