

## Memoria Practica 2

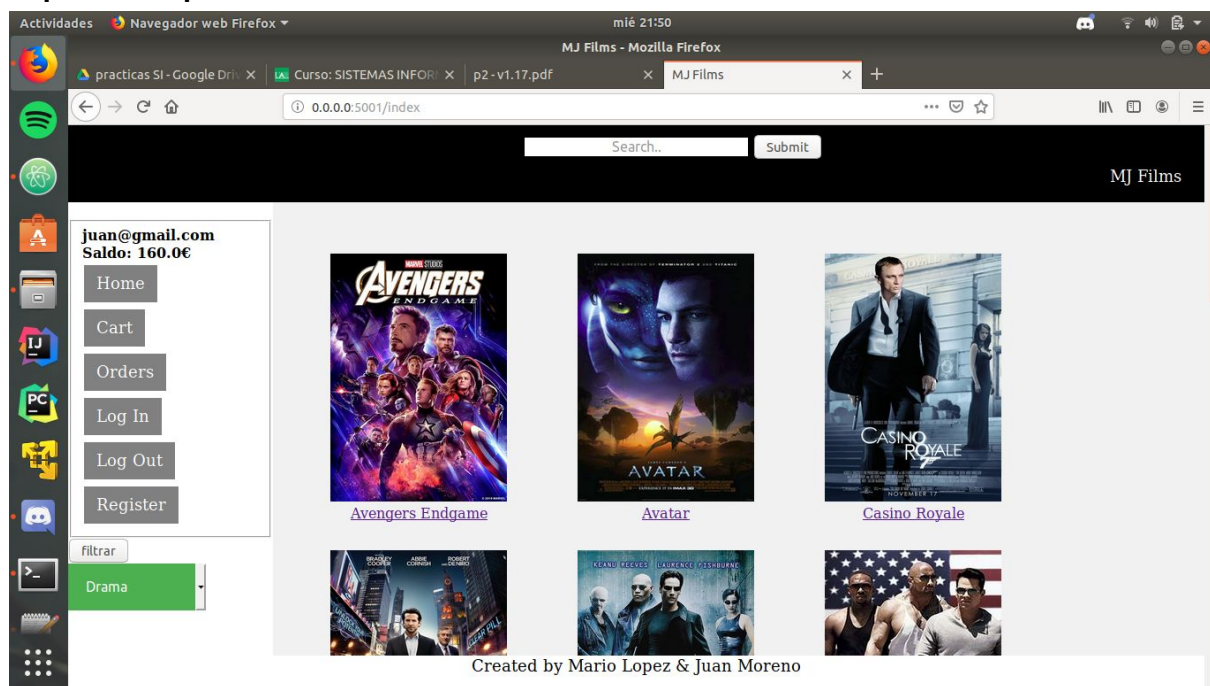
Descripción del funcionamiento asociado a cada template implementada:

- Index: usando un form para la búsqueda y otro para el boton de categorias, el routes.py recibe los datos solicitados y genera el html pertinente. Por otro lado con el uso de session se muestra el usuario conectado y su saldo en la pantalla principal. Al seleccionar una categoría y darle al botón filtrar, muestra las películas que pertenecen a dicha categoría. El motor de búsqueda funciona introduciendo el nombre exacto de la película.
- Register: el form envía los datos (verificados previamente que cumplen con la forma estándar) y se crea una carpeta con el email del usuario que contiene un datos.dat con el resto de sus datos. Para poder registrarse primero hay que hacer logout si es que se está logeado. Se asigna un saldo que se almacena en session y en el datos.dat. Si intentas registrar un email que ya registrado se redirige al login, mostrando mensaje de información usando un flag.
- Login: el form envía los datos a la función de routes, en caso de que ya haya un usuario logeado, se redirige al index. Se completa el login si se encuentra la carpeta correspondiente a tal usuario y coincide con su contraseña(encriptada en md5) del datos.dat.
- Detalles: en la función del fichero routes.py “detalles”, recibimos como parámetro el título de la película de la cual queremos mostrar los detalles. Este parámetro lo recibimos desde el archivo index.html. Leemos del fichero json los detalles que nosotros queremos mostrar de cada película, en nuestro caso descripción, título y precio.
- Cart: se muestran todas las peliculas que estan almacenadas en session['carrito'] (el diccionario que contiene para cada título de una película, su precio.). Esta página contiene un form para incrementar el saldo y otro para comprar las películas que hay en el carrito con el botón purchase, posteriormente se actualiza el saldo. También escribe en el history.json las películas compradas. Si se quiere incrementar el saldo, habrá que poner la tarjeta de crédito correspondiente con el usuario, sino redigirá al mismo carrito vaciando los datos de los campos.
- History: se accede desde el botón orders. La función del routes le pasa los datos del history.json. Hemos conseguido que en el historial.json de cada usuario se muestre la última compra. Si al realizar una compra, intentamos volver a hacer la misma acción, se actualizará el json con la última compra realizada.

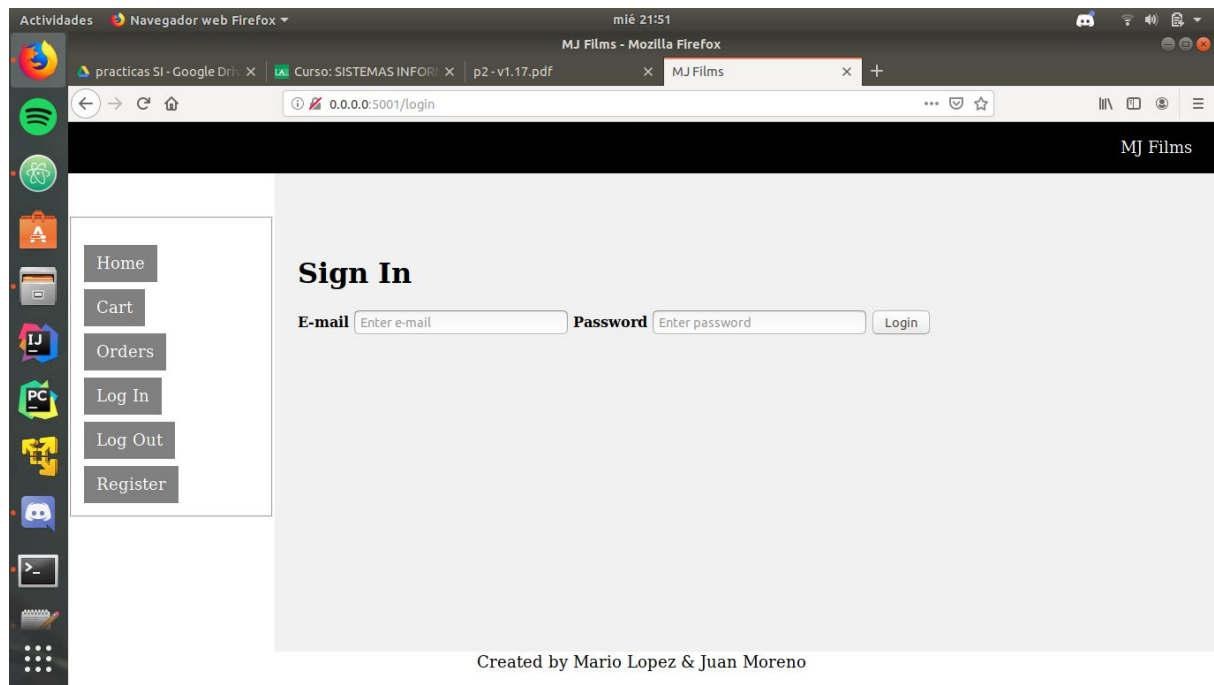
Función `elim(film)` del `routes.py`: hemos creado esta función la cual recorre las keys del diccionario `session['carrito']` y lo compara con la película introducida que se obtiene desde el `cart.html` usando `urlfor`. Hemos implementado una cookie la cual recuerda el correo de el último inicio de sesión. Librería nueva utilizada: `make_response`, la cual nos permite el uso de sus funciones para poder trabajar con la cookie.

Las fuentes de ayuda que principalmente hemos utilizado son: StackOverflow para resolver dudas generales y la bibliografía que se nos proporciona en el enunciado.

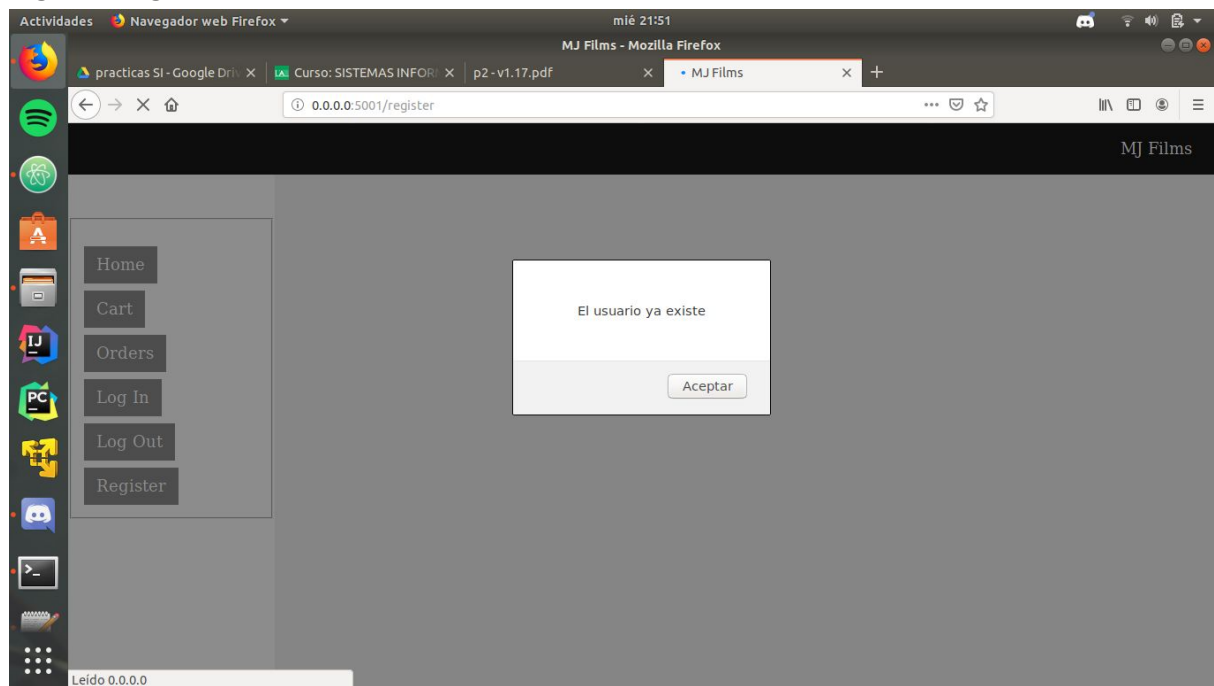
### Capturas de pantalla:



Home (index.html)



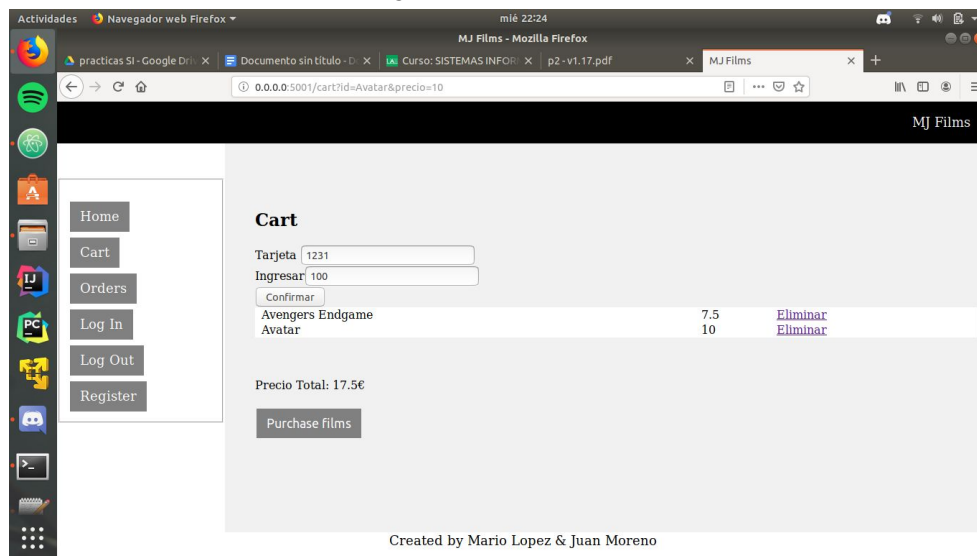
### Sign in (login.html)



### Register(register.html tras registrar un usuario)



## Historial de compras (history.html)



## Cart(cart.html)

Mario Lopez y Juan Moreno