



PRÁCTICA 2

Programación Web Dinámica

Asignatura: Sistemas Informáticos 1

Profesor: Antonio Garmendia

Realizado por: Vanessa Salazar

Nicolás Trejo

Curso: 2018 / 2019

Tabla de contenido

Ficheros que componen la entrega3

Instrucciones de uso4

 Preparación del entorno4

 Ejecución de la aplicación4

Funcionamiento de la app6

 Navegación – frontend.....6

 Lógica - backend.....9

Bibliografía12

Ficheros que componen la entrega

hello.wsgi

app

- `__init__.py`
- `__main__.py`
- `routes.py`
- `catalogue`
 - `catalogue.json`
- `static`
 - `estilo.css`
 - `función.js`
- `templates`
 - `base.html`
 - `carrito.html`
 - `categorías.html`
 - `datousuario.html`
 - `historial.html`
 - `index.html`
 - `login.html`
 - `pago`
 - `película.html`
 - `signin.html`
- `thesessions`
 - Incluirá las sesiones de la aplicación conforme el uso de la misma
- `usuarios`
 - Contendrá ficheros con el nombre `{{user}}`
 - `data.dat`
 - `historial.json`

Instrucciones de uso

Preparación del entorno

Después de haber instalado los paquetes necesarios para el desarrollo del proyecto tales como son: apache2 , python versión 2, flask, virtualenv, etc. Se requiere crear un entorno virtual de Python con Flask, el mismo que nos permitirá probar las páginas desarrolladas, esto se realizará mediante los mandatos descritos a continuación, “si1pyenv” será el nombre que daremos al entorno virtual a emplear.

```
$ virtualenv -p python2 si1pyenv
$ source si1pyenv/bin/activate
$ pip install Flask SQLAlchemy Flask-SQLAlchemy SQLAlchemy-Utils \ pycopg2
'itsdangerous
```

Una vez creado el entorno virtual se creará un directorio denominado “public_html” que deberá estar publicacio en la ruta `http://localhost/~usuario/`, ubicación donde se alojarán los ficheros del proyecto, para esto es necesario ejecutar los siguientes mandatos dentro de \$HOME (home/usuario) :

```
$ mkdir ~/public_html
$ sudo a2enmod userdir
$ sudo service apache2 restart
```

Después de creado el directorio, se deberá configurar el módulo “mod_wsgi”, cuya función es ser un nexo entre el servidor web(Apache) y la aplicación web desarrollada(python), que a través de un fichero de extensión .wsgi que se ejecutará al inicio y permitirá acceder a la aplicación. Esto se realiza con los siguientes pasos:

- Se ejecuta en el terminal

```
$ sudo a2enmod wsgi
```

- Posteriormente se abre y edita el archivo “userdir.conf”, ubicado en la ruta `/etc/apache2/mods-enabled/userdir.conf`, en donde se debe añadir ExecCGI a las opciones y wsgi-script .wsgi como handler.

```
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec ExecCGI
AddHandler wsgi-script .wsgi
```

Ejecución de la aplicación

El acceso a los ficheros de la aplicación puede realizarse de dos formas:

1. Accediendo a través del browser, mediante la ruta:

```
http://localhost/~usuario/hello.wsgi
```

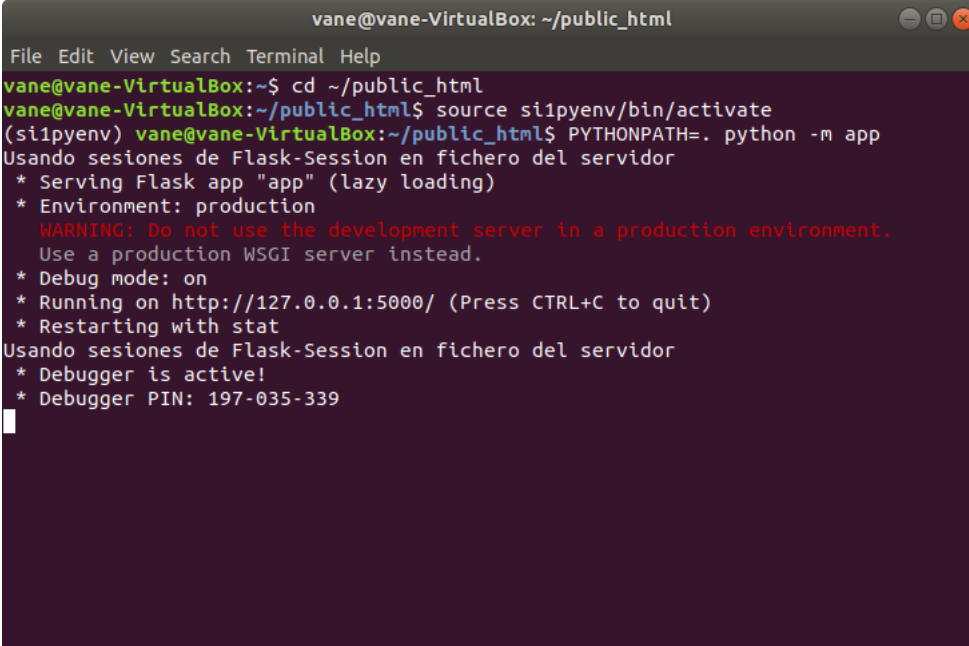
El archivo “hello.wsgi” activa el entorno virtual e importa los archivos que se encuentran en la carpeta app, mediante la sentencia:

```
from app import app as application
```

2. Empleando el terminal, activando el entorno virtual y ejecutando la aplicación mediante los mandatos descritos a continuación, los mismos que al ser ejecutados activan el modo

debug y permiten acceder a la aplicación desde el browser a través de la ruta `http://127.0.0.1:5000/`

```
cd~/public_html
source si1pyenv/bin/activate
PYTHONPATH=. python -m app
```

A screenshot of a terminal window titled "vane@vane-VirtualBox: ~/public_html". The terminal shows the following commands and output:

```
vane@vane-VirtualBox:~$ cd ~/public_html
vane@vane-VirtualBox:~/public_html$ source si1pyenv/bin/activate
(si1pyenv) vane@vane-VirtualBox:~/public_html$ PYTHONPATH=. python -m app
Usando sesiones de Flask-Session en fichero del servidor
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
Usando sesiones de Flask-Session en fichero del servidor
* Debugger is active!
* Debugger PIN: 197-035-339
```

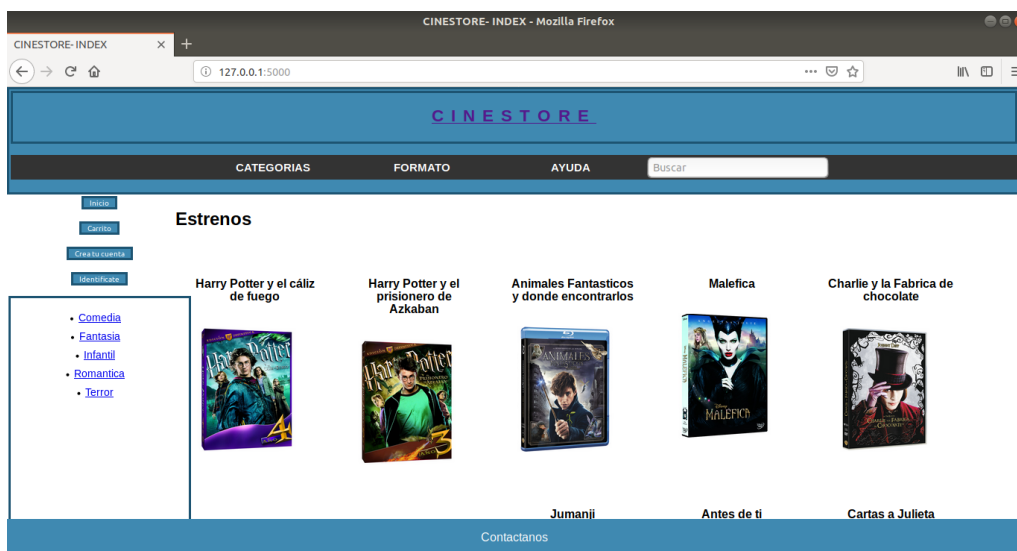
Funcionamiento de la app

Navegación – frontend

Al iniciar el debug, se muestra el template “index.html”, en el caso de esta aplicación , esta pagina se considera como pagina de inicio.

Index: muestra en su contenido el catálogo de películas del fichero .json, cada una con su respectivo poster y título.

Desde esta pagina es posible acceder a los templates de : detalle de películas, Sign in, Log in, Log out, Carrito e Historial



Detalle de Peliculas: Al dar click sobre cualquiera de las peliculas mostradas en el index, se accede al detalle de la misma , en donde se puede ver la portada, titulo, actores, sinopsis, formato y precio de la misma. Además desde esta página será posible añadir la pelicula al carrito de compras, para la posterior gestión de compra de ser deseado.



Sign in : Esta opción nos permite crear un nuevo usuario en la aplicación, el mismo que deberá ingresar un username, nombre, apellido, correo electrónico y número de tarjeta, la información ingresada deberá cumplir con ciertas restricciones para ser válida así como:

Username: no deberá tener espacios y deberá tener menos de 10 caracteres y a más de 5, además de verificará que no exista otro usuario igual, de ser así se mostrará un mensaje que lo indique.

Nombre: deberán ser solo letras y deberá tener más de 2 caracteres y menos 12

Apellido: deberán ser solo letras y deberá tener más de 2 caracteres y menos 12

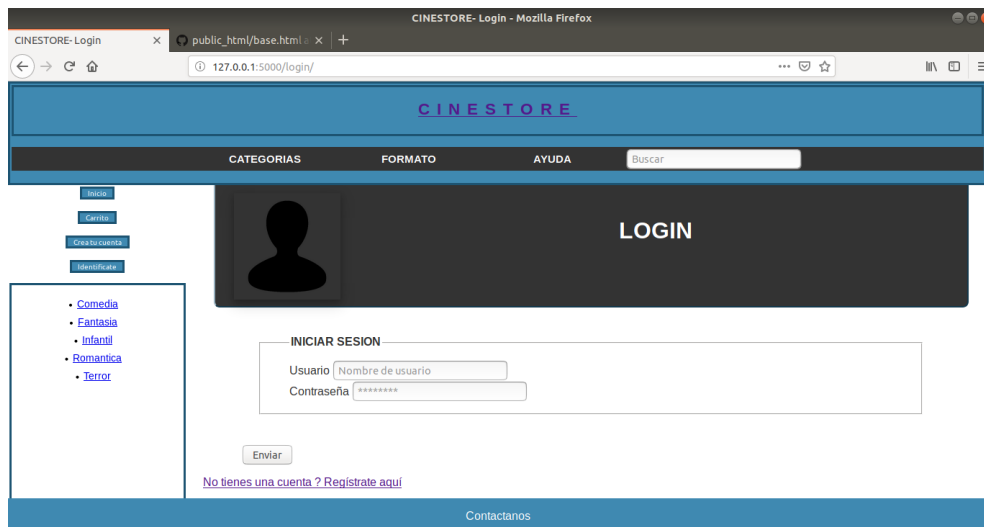
Contraseña: deberá tener 8 caracteres como mínimo, esta se comparará con la confirmación de contraseña para corroborar que ambas sean iguales.

Correo: deberá tener formato correcto xxxx@mail.com

Tarjeta: deberá tener 10 caracteres y deberán ser solo números.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/signin/'. The page title is 'CINESTORE- Registro usuario - Mozilla Firefox'. The main content area is titled 'CINESTORE' and features a navigation bar with 'CATEGORIAS', 'FORMATO', and 'AYUDA'. Below the navigation bar, there is a sidebar menu with links for 'Inicio', 'Cine', 'Cine en casa', 'Identificarse', and a list of genres: 'Comedia', 'Fantasia', 'Infantil', 'Romantica', and 'Terror'. The main form is titled 'PERSONA' and contains the following fields: 'Username' (filled with 'Pepe17'), 'Nombre' (filled with 'pepe'), 'Apellido' (filled with 'gonzalez'), 'Correo electrónico' (filled with 'pepe@mail.com'), 'Tarjeta' (filled with '1112223334'), 'Contraseña' (filled with '*****'), and 'Confirme Contraseña' (filled with '*****'). A file upload button is labeled 'Browse...' and 'No file selected.'. Below the form, there is a checkbox for 'Acepta las condiciones y términos de esta página' and an 'Enviar' button. The footer of the page displays 'Contactanos'.

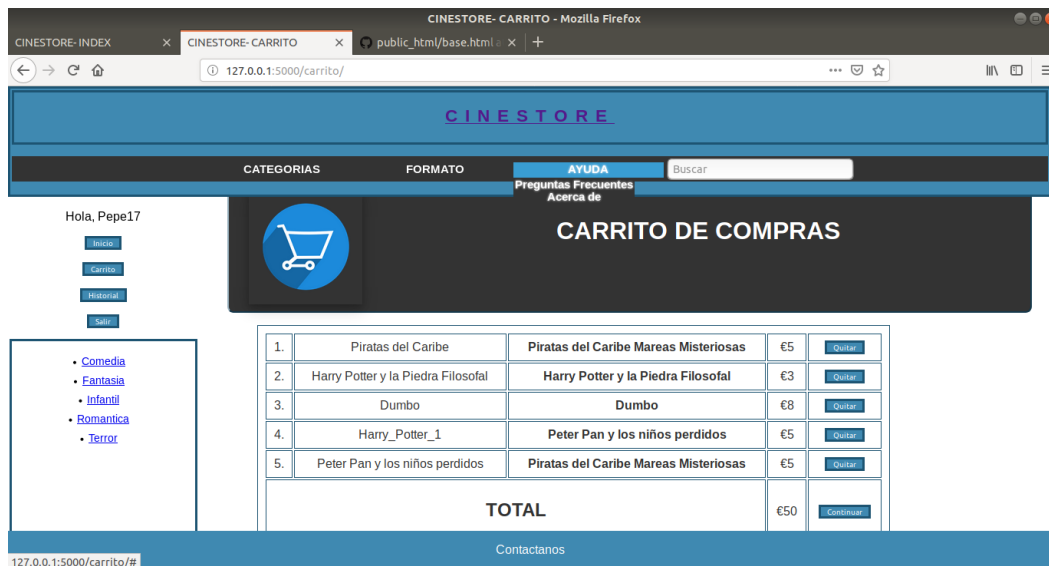
Log In: Una vez creado el usuario, este podrá acceder al sistema con el nombre de usuario y contraseña, se verificarán en los datos ingresados en el registro corroborando que la contraseña ingresada sea igual a la contraseña que fue almacenada por la aplicación.



Historial: La opción de historial aparecerá una vez que el usuario haya ingresado al sistema con su nombre de usuario y contraseña, en esta página podrá acceder a la información de las películas que ha comprado anteriormente, además contendrá una opción que permitirá visualizar los datos del usuario.

Datos de Usuario: Mediante esta página el usuario podrá visualizar los datos que ingreso al momento del registro además de poder conocer cual es el saldo restante que le queda para realizar compras.

Carrito: El carrito de compras permite almacenar temporalmente las películas que el usuario a añadido al mismo. Para gestionar la compra de los items que se encuentran en el carrito es necesario que el usuario ingrese al sistema, con un usuario y contraseña previamente creados.



Pago: Dentro de esta página el usuario podrá terminar el proceso de compra, sumando los valores de las películas que se encuentran en el carrito y restandolas del saldo que este posee. Se mostrará un mensaje de compra exitosa o de saldo insuficiente dependiendo de cual sea el caso.

Lógica - backend

Las paginas de la aplicación incluyen el template “base.html”, este es usado con el fin de hacer uniforme la presentacion del header, footer y menú lateral en todas las páginas de la aplicación. En este template se incluye las rutas de enlace de todos los estilos de css y funciones de js que maneja la aplicación.

Las categorias que se muestran en el menu lateral , se generan de forma automatica mediante una busqueda de los generos disponibles en el json de la aplicación y la publicación de los mismos, estos variaran en función del catalogo de JSON.



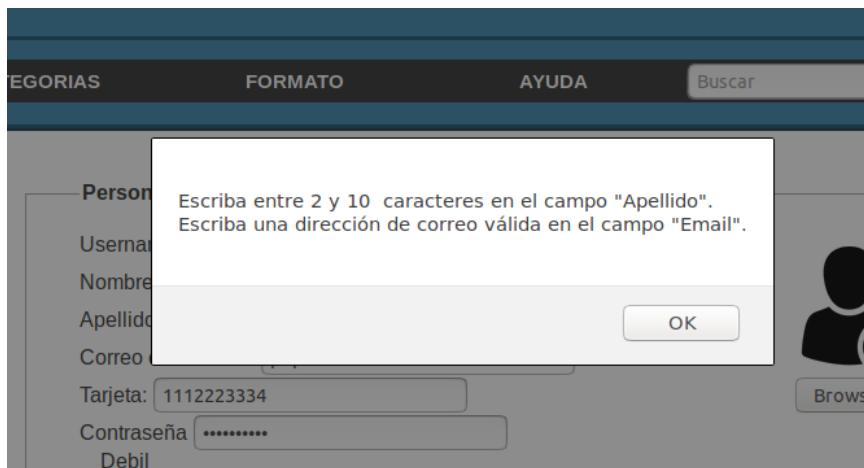
```
{
  "peliculas": [
    {
      "titulo": "Harry Potter y el cáliz de fuego ",
      "sigla": "harry4",
      "poster": "imagenes/harry4.jpg",
      "director": "Alfonso Cuarón",
      "precio": "14.99",
      "genero": "Fantasia",
      "anno": "2016",
      "formato": "DVD",
      "sinopsis": "La cuarta parte de la serie del niño mago comienza con la Copa Internacional de Quidditch. Cuenta también el inicio de la atracción por Cho Chang y otro año de magia, en el que una gran sorpresa obligará a Harry a enfrentarse a muchos desafíos terribles. También habrá un torneo de magia para tres escuelas, y el temido regreso de Aquel-que-no-debe-ser-nombrado. ",
      "actores": [
        "Daniel Radcliffe",
        "Emma Watson",
        "Rupert Grint"
      ]
    },
    {
      "titulo": "Harry Potter y el prisionero de Azkaban ",
      "sigla": "harry3",
      "poster": "imagenes/harry3.jpg",
      "director": "Alfonso Cuarón",
      "precio": "14.99",
      "genero": "Fantasia",
      "anno": "2016",
      "formato": "DVD",
      "sinopsis": "Cuando Harry Potter y sus amigos vuelven a Hogwarts para cursar su tercer año de estudios, se ven involucrados en un misterio: de la prisión para magos de Azkaban se ha fugado Sirius Black, un peligroso mago que fue cómplice de Lord Voldemort y que intentará vengarse de Harry Potter. El joven aprendiz de mago contribuyó en gran medida a la condena de Sirius, por lo que hay razones para tener por su vida.",
      "actores": [
        "Daniel Radcliffe",
        "Emma Watson",
        "Rupert Grint"
      ]
    }
  ]
}
```

Adicionalmente en el menu lateral del template base, se encuentran las opciones de acceso a varias paginas del sistema, las mismas que estan disponibles de acuerdo al estado de la session que maneje en ese momento la aplicación. Si “user” no se encuentra “in session”, estaran disponibles las opciones de carrito, Sig in y Log in, y cuando “user” se encuentre “in session”, se harán visibles las opciones de historial y Log in y desaparecerán las dos primeras, además se mostrará un mensaje de bienvenida con el nombre del usuario que se encuentre ese momento “in session”.

Index: El index esta programado para realizar una busqueda en el catalogo de JSON todas las peliculas disponibles y las muestra con su respectivo titulo y poster.

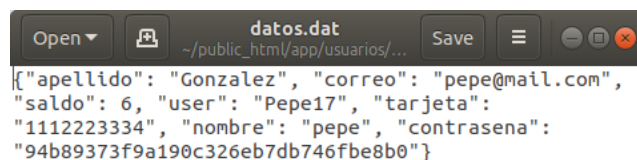
Detalle de peliculas: El detalle de peliculas a traves de una funcion identifica la peliual que ha seleccionado el usuario y muestra los datos almacenados en el catalogo de JSON correspondientes a esa pelicula, ademas a través del boton añadir al carrito podemos agregar esta pelicula a la lista temporal que el carrito de compras almacena.

Sign in : Se podrá crear un nuevo registro siempre y cuando no haya ninguna sesion iniciada, es necesario ingresar al formulario los datos solicitados, los mismos que serán validados por la funcion validar.js.



Cuando los datos cumplan con el formato que se solicita y al hacer submit , se comprobará a través de “signin_activate”, que el Nombre de usuario ingresado no exista, esto se realizará buscando en la carpeta “usuarios”, un fichero con el nombre ingresado, de existir una carpeta con ese nombre se muestra un mensaje indicando que el usuario ya existe, caso contrario se crea una carpeta con el nombre del usuario ingresado.

La contraseña se encriptara con md5, para esto es necesario incluir en routes el uso de la librería mediante: `from hashlib import md5`, y llamando a la función “createuserlogin” que además de realizar la encriptación de contraseña, almacenará los datos ingresados en un archivo “datos.dat” , añadiendo una categoría adicional llamada saldo, a la que se le asignara un valor random entre 0 y 100.



Cuando esto se realice, el usuario hace log in automático a la aplicación con el nombre de usuario y contraseña ingresados, iniciando a “user” is “in session”.

Log in: Dentro de esta página el usuario debe ingresar el nombre de usuario y contraseña que ha registrado en el sistema, a través de “login_activate”, se buscan dentro del archivo “datos.dat” la contraseña almacenada en el momento del registro y mediante la función “check_password” se comprueba si es la misma que ha ingresado el usuario, de no ser así muestra un mensaje de error, y si es la misma contraseña se inicia la session “user” , con todos los datos del usuario para su uso posterior.

Hola, Pepe17

Inicio

Carrito

Historial

Salir

Log out: Esta opción se encuentra disponible únicamente cuando existe una sesión iniciada, a través de la sentencia “session.clear()” elimina toda la información de la sesión del “user” actual , y permite que otro usuario pueda iniciar una nueva sesión

Historial: El historial permite leer dentro de la ruta: “usuarios/{{user}}/historial.json” la información sobre las películas que el cliente ha comprado anteriormente , así como permite escribir en este las nuevas películas que el usuario ha adquirido.

Datos de usuario: Con la sesión activa, la página datos de usuario muestra la información del usuario que fue ingresada en el momento del registro , extrayéndola de la ruta “usuarios/{{user}}/datos.dat”

Carrito: Al añadir una película al carrito, se almacena en una lista temporal dentro de la sesión actual, en donde se incluyen las películas seleccionadas por el usuario, se extrae del catálogo de JSON información sobre esta, como es el título de la película y el precio. Para poder continuar con la compra es necesario que el usuario haya iniciado sesión, solo de esta forma podrá ingresar a la opción de pago.

Pago: En la opción pago se realiza la suma de precios de las películas que se encuentran en el carrito , y este valor se resta de la variable saldo almacenada en el archivo “datos.dat”, si el valor de saldo es mayor al valor de la suma , se efectúa la compra y se eliminan los ítems del carrito, de no ser así se muestra un mensaje de fondos insuficientes.

Bibliografía

WEB:

Miguel Grinberg (Octubre 2018). *The Flask Mega-Tutorial Part I: Hello World!* - *miguelgrinberg.com*. Recuperado de <https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>

Tutorials Point (Octubre 2018). *Flask Tutorial*. Recuperado de <https://www.tutorialspoint.com/flask>

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*.

O'Reilly Media, Inc.

Ronacher, A. (2010, 2018). Flask (A Python Microframework). Retrieved October 31, 2018, from <http://flask.pocoo.org/>

Maia, I. (2015). *Building Web Applications with Flask*. Packt Publishing Ltd.