

Actividad 1 - Flask: Formulario de Películas

Desarrollo de Aplicación con Flask, SQLite y Bootstrap

Autor: Abel Jiménez

Curso: 2º DAW

Fecha: 17 de octubre de 2025

Introducción

El presente documento describe el desarrollo y funcionamiento de una aplicación web creada con Flask. Esta aplicación tiene como objetivo permitir la gestión de películas mediante un formulario que permite añadir títulos a una base de datos SQLite. El trabajo se ha llevado a cabo utilizando las tecnologías Flask, Jinja2 y Bootstrap, con un enfoque en la validación de datos, diseño responsivo y buenas prácticas de programación.

Desarrollo del Proyecto

La aplicación se estructura en una arquitectura sencilla pero eficaz. En primer lugar, se creó una plantilla HTML llamada 'add_movie.html', la cual hereda del archivo base 'base.html'. Gracias al uso de Bootstrap 5, el formulario presenta un diseño moderno, ordenado y adaptable a diferentes tamaños de pantalla. Dentro del formulario se incluyen campos para capturar los principales atributos de una película, como el título, director, año, género, calificación, descripción y la URL del póster. El campo del título se definió como obligatorio para asegurar la integridad de los datos.

Código de add_movie.html

```
.Práctica Flask > 1 - Práctica Flask > frontend > templates > add_movie.html
1  {% extends "base.html" %} 
2
3  {% block title %}Añadir película - CineFlask{% endblock %}
4
5  {% block content %} 
6      

7          

8              

### Añadir nueva película


9              
10                  

11                      Título
12                      
13


14                  

15                      Director
16                      
17


18                  

19                      

20                          

21                              Año
22                              
23


24                          

25                              Género
26                              
27


28                          

29                              Rating
30                              
31


32


33


34                  

35                      URL del póster
36                      
37


38                  

39                      Descripción
40                      
41


42                  

43                      Guardar
44                      Cancelar
45


46              
47  {% endblock %}


```

Resultado:

The screenshot shows a web application interface for adding a new movie. At the top, there is a navigation bar with links for 'Inicio', 'Películas', 'Añadir película', 'Sobre', and 'Contacto'. The main content area has a title 'Añadir nueva película'. Below the title are several input fields: 'Título *' (Title *), 'Director', 'Año' (Year), 'Género' (Genre), 'Rating', 'URL del póster' (Poster URL), and 'Descripción' (Description). There are two buttons at the bottom: a blue 'Guardar' (Save) button and a white 'Cancelar' (Cancel) button.

Para mejorar la navegación dentro del sitio, se incorporó un enlace en el menú principal que permite acceder al formulario de manera rápida y visual, complementado con un ícono representativo de Font Awesome. El código de Flask gestiona esta funcionalidad mediante una ruta definida en el archivo principal 'app.py'. Dicha ruta acepta los métodos GET y POST, de modo que al acceder mediante GET se muestra el formulario vacío, y al enviar el formulario con POST se procesan los datos y se almacenan en la base de datos.

Implementación Técnica

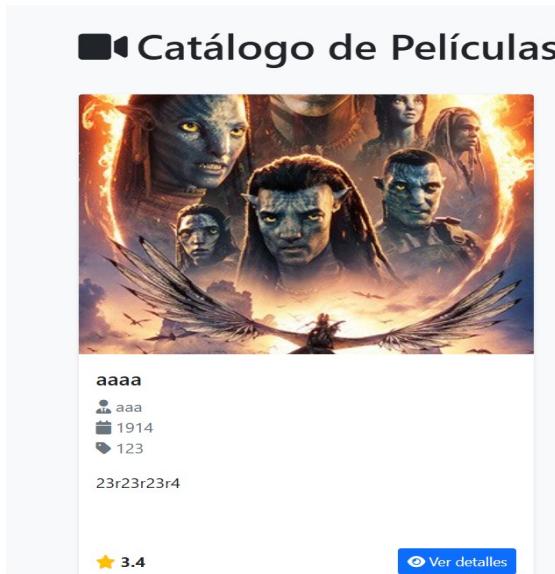
Durante el desarrollo se implementaron validaciones básicas, como la obligatoriedad del campo título, y la conversión segura de tipos numéricos en los campos año y calificación. Los datos son insertados en la base de datos SQLite utilizando sentencias parametrizadas, lo que garantiza la seguridad frente a inyecciones SQL. Además, se utilizaron mensajes 'flash' para informar al usuario sobre el resultado de sus acciones, ya sea la inserción exitosa de una película o la falta de datos requeridos.

La tabla principal, denominada 'movies', contiene los campos necesarios para almacenar toda la información relevante: identificador, título, director, año, género, calificación, descripción y URL del póster. De esta forma, el sistema permite mantener un registro organizado y persistente de las películas añadidas por el usuario.

```
def init_db():
    """Create the movies table if it doesn't exist."""
    conn = get_db_connection()
    with conn:
        conn.execute('''
            CREATE TABLE IF NOT EXISTS movies (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                title TEXT NOT NULL,
                director TEXT,
                year INTEGER,
                genre TEXT,
                rating REAL,
                description TEXT,
                poster_url TEXT
            )
        ''')
    conn.close()

init_db()
```

Resultado:



Uso y Verificación

Para utilizar la aplicación, basta con ejecutar el archivo 'app.py' desde el entorno local y acceder al enlace <http://localhost:5000>. Desde la barra de navegación se puede seleccionar la opción 'Añadir película', donde se despliega el formulario. Al completar los campos y enviar la información, la aplicación muestra un mensaje confirmando la inserción y redirige automáticamente a la lista de películas registradas. Los datos permanecen almacenados en la base de datos 'movies.db', pudiendo verificarse mediante herramientas como SQLite Browser.

Conclusión

El desarrollo de esta actividad ha permitido comprender de manera práctica el funcionamiento de Flask como framework web y su integración con plantillas Jinja2, Bootstrap para el diseño y SQLite para el almacenamiento de datos. El formulario de películas cumple con todos los requisitos solicitados, aplicando validaciones, una estructura limpia y un flujo de navegación intuitivo. Este proyecto demuestra la correcta aplicación de buenas prácticas en el desarrollo de aplicaciones web con Python.