

# **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.

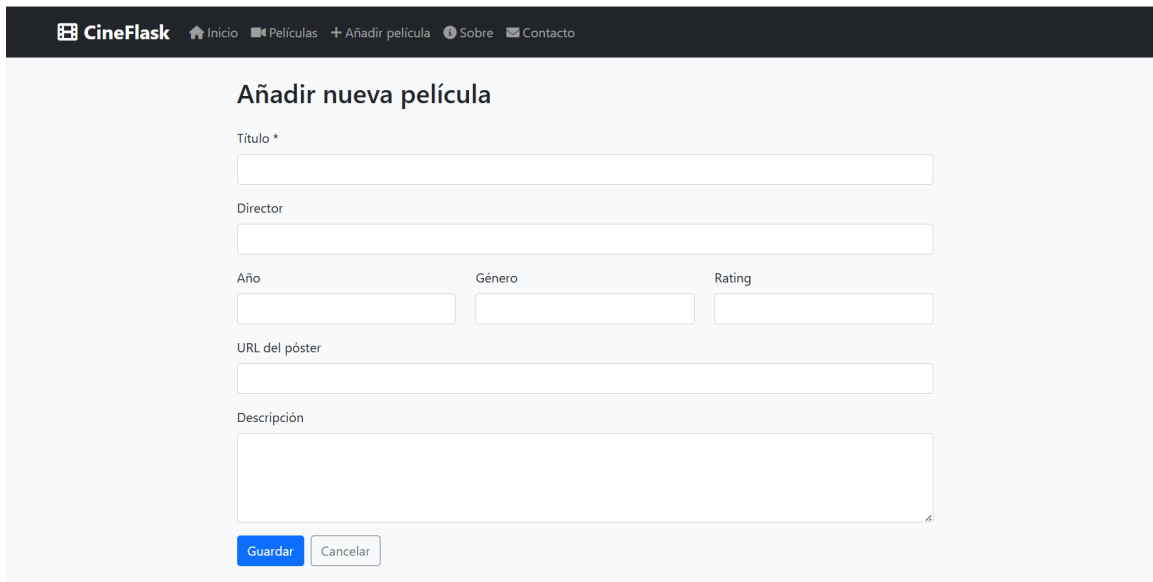
## Desarrollo del Proyecto

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 > templates > @ add_movie.html
1  {% extends "base.html" %}
2
3  {% block title %}Añadir película - {{ flask.request.args.get('movie') }}{% endblock %}
4
5  {% block content %}
6      <div class="row justify-content-center">
7          <div class="col-md-8">
8              <h1 class="mb-4">Añadir nueva película /</h1>
9              <form method="post" action="{ url_for('add_movie') }}">
10                  <div class="mb-3">
11                      <label for="title" class="form-label">Título *(</label>
12                      <input type="text" class="form-control" id="title" name="title" required>
13                  </div>
14                  <div class="mb-3">
15                      <label for="director" class="form-label">Director(</label>
16                      <input type="text" class="form-control" id="director" name="director">
17                  </div>
18                  <div class="row">
19                      <div class="col-md-4 mb-3">
20                          <label for="year" class="form-label">Año(</label>
21                          <input type="number" class="form-control" id="year" name="year" min="1888">
22                      </div>
23                      <div class="col-md-4 mb-3">
24                          <label for="genre" class="form-label">Género(</label>
25                          <input type="text" class="form-control" id="genre" name="genre">
26                      </div>
27                  </div>
28                  <div class="col-md-4 mb-3">
29                      <label for="rating" class="form-label">Rating(</label>
30                      <input type="number" step="0.1" class="form-control" id="rating" name="rating" min="0" max="10">
31                  </div>
32                  <div class="mb-3">
33                      <label for="poster_url" class="form-label">URL del póster(</label>
34                      <input type="url" class="form-control" id="poster_url" name="poster_url">
35                  </div>
36                  <div class="mb-3">
37                      <label for="description" class="form-label">Descripción(</label>
38                      <textarea class="form-control" id="description" name="description" rows="4"></textarea>
39                  </div>
40                  <div class="d-flex gap-2">
41                      <button type="submit" class="btn btn-primary">Guardar</button>
42                      <button type="button" class="btn btn-outline-secondary">Cancelar</button>
43                  </div>
44              </form>
45          </div>
46      </div>
47  {% endblock %}
```

Resultado:



The screenshot shows a web application interface for adding a new movie. At the top, there is a dark navigation bar with the 'CineFlask' logo and several menu items: 'Inicio', 'Películas', '+ Añadir película', 'Sobre', and 'Contacto'. The main content area is titled 'Añadir nueva película'. It contains a form with the following fields: 'Titulo \*' (required), 'Director', 'Año', 'Género', 'Rating', 'URL del póster', and 'Descripción'. The 'Año', 'Género', and 'Rating' fields are grouped together. At the bottom of the form, there are two buttons: 'Guardar' (blue) and 'Cancelar' (white).

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 icono 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

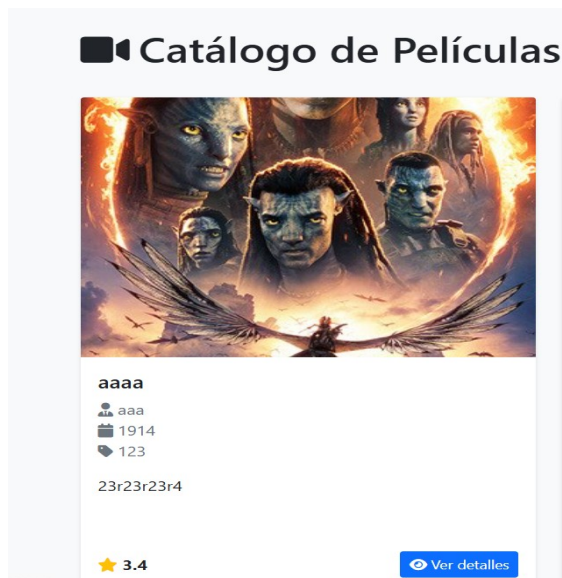
Durante el desarrollo se implementaron validaciones básicas, como la obligatoriedad del campo título, y la conversión segura de tipos numéricos en los campos año y calificación. Los datos son insertados en la base de datos SQLite utilizando sentencias parametrizadas, lo que garantiza la seguridad frente 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.