Lidia Artero Fernández – 2º DAW

Actividad 1: Flask

(Añadir formulario para películas)

Módulo: Frameworks aplicados a la web
Fecha: 17/10/2025

Índice

1. Introducción

En esta actividad se ha implementado un sistema para añadir películas a través de un formulario web. La funcionalidad incluye:

- Creación de formulario HTML con Jinja2
- Integración en la navegación del sitio
- Validación de datos del formulario
- Procesamiento de datos con método POST

2. Paso 1: Creación del Formulario HTML (add_movie.html)

2.1. Estructura del Template

El formulario se creó extendiendo base.html para mantener la consistencia del diseño:

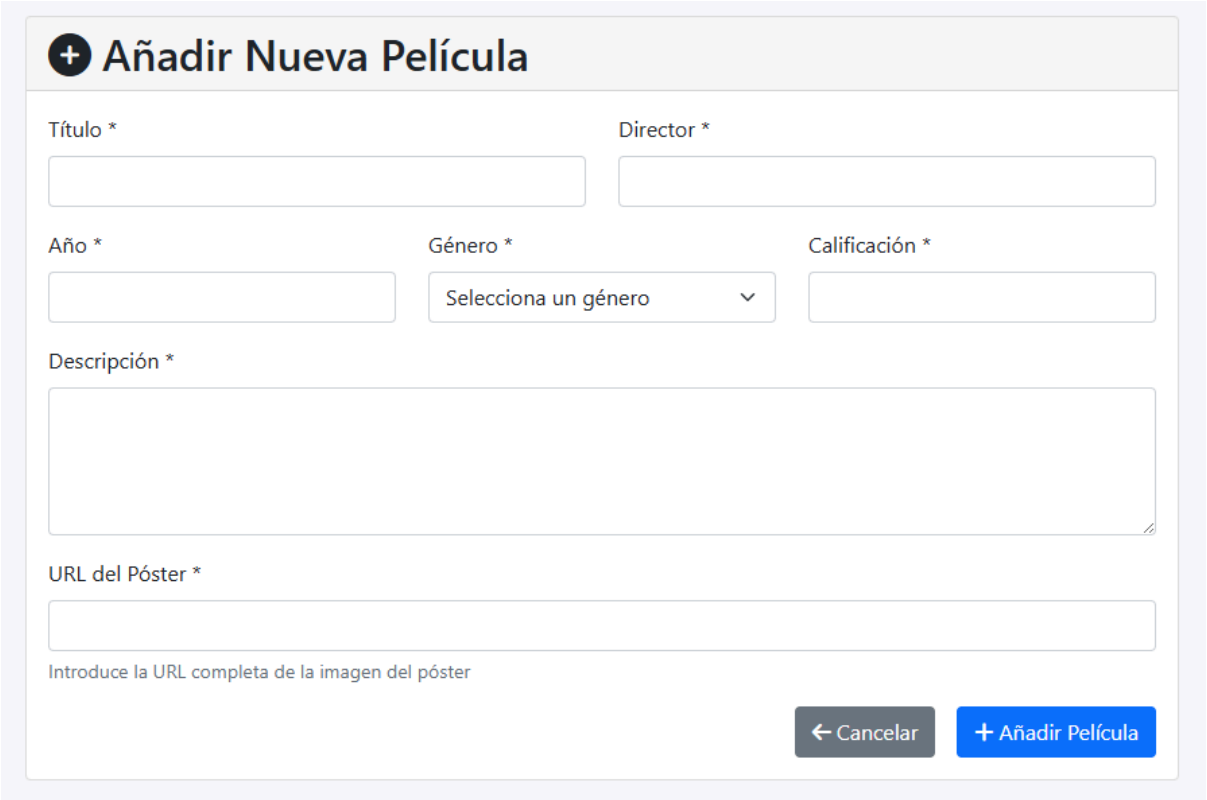
```
{% extends "base.html" %}

{% block title %}Añadir Película - CineFlask{% endblock %}

{% block content %}
<!-- Contenido del formulario -->
{% endblock %}
```

2.2. Campos del Formulario

El formulario incluye los siguientes campos para capturar toda la información de una película:

- **Campos implementados:**
  - **Título** (text): Nombre de la película
  - **Director** (text): Nombre del director
  - **Año** (number): Año de estreno (rango: 1900-2030)
  - **Género** (select): Lista desplegable con géneros predefinidos
  - **Calificación** (number): Puntuación de 0 a 10 (con decimales)
  - **Descripción** (textarea): Sinopsis de la película
  - **URL del Póster** (url): Enlace a la imagen del póster
- **Atributos importantes:**
  - method="POST": Envía los datos de forma segura
  - action="{{ url_for('add_movie') }}": Define la ruta de destino

A continuación se muestra como quedaría el resultado del formulario antes de introducir los datos de película:



2.3. Géneros Disponibles en el Selector

Se incluyeron los siguientes géneros cinematográficos:

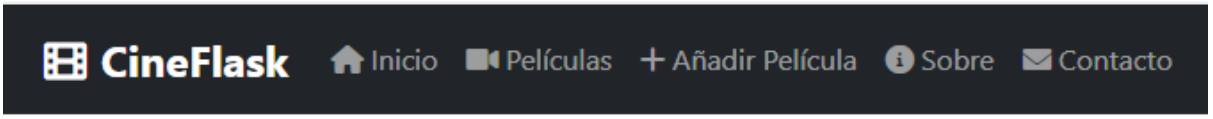
- Drama
- Crimen
- Acción
- Comedia
- Thriller
- Terror
- Ciencia Ficción
- Romance
- Aventura
- Fantasía

3. Paso 2: Añadir Enlace en la Navegación

3.1. Modificación del Navbar en base.html

Se añadió un nuevo elemento en el menú de navegación para acceder al formulario:

```
<li class="nav-item">
    <a class="nav-link" href="{{ url_for('add_movie') }}">
        <i class="fas fa-plus me-1"></i>Añadir Película
    </a>
</li>
```



4. Paso 3: Crear la Ruta en app.py

4.1. Definición de la Ruta

Se implementó la ruta /add-movie que acepta tanto GET como POST:

```
@app.route('/add-movie', methods=['GET', 'POST'])
def add_movie():
    # Lógica del formulario
```

5. Paso 4: Lógica para Procesar el Formulario

5.1. Comprobar el Método de la Petición

```
if request.method == 'POST':
    # Procesar datos del formulario
else:
    # Mostrar formulario (GET)
    return render_template('add_movie.html')
```

5.2. Recoger Datos del Formulario

Los datos se obtienen usando request.form.get():

```
title = request.form.get('title', '').strip()
director = request.form.get('director', '').strip()
year = request.form.get('year', '').strip()
genre = request.form.get('genre', '').strip()
rating = request.form.get('rating', '').strip()
description = request.form.get('description', '').strip()
poster_url = request.form.get('poster_url', '').strip()
```

5.3. Validación de Campos Requeridos

Primera validación: Verificar que todos los campos estén presentes:

```
if not all([title, director, year, genre, rating, description,
poster_url]):
    # Mostrar error o redirigir
    return redirect(url_for('add_movie'))
```

5.4. Validación de Tipos de Datos
Convertir strings a tipos numéricos apropiados:

```
try:
    year = int(year)
    rating = float(rating)
except ValueError:
    # Manejar error de conversión
    return redirect(url_for('add_movie'))
```

5.5. Validación de Rangos
Verificar que los valores estén dentro de rangos lógicos:

```
if year < 1900 or year > 2030:
    # Año fuera de rango
    return redirect(url_for('add_movie'))

if rating < 0 or rating > 10:
    # Calificación fuera de rango
    return redirect(url_for('add_movie'))
```

6. Paso 5: Inserción en la Base de Datos SQLite

6.1. Función para Añadir Películas
Se creó una función para insertar datos:

```
def add_movie_to_db(title, director, year, genre, rating,
description, poster_url):
    with get_db_connection() as conn:
        conn.execute('''
            INSERT INTO movies (title, director, year, genre,
rating, description, poster_url)
            VALUES (?, ?, ?, ?, ?, ?, ?)
        ''', (title, director, year, genre, rating, description,
poster_url))
        conn.commit()
```

6.2. Llamada a la Función
Una vez validados todos los datos:

```
add_movie_to_db(title, director, year, genre, rating, description,
poster_url)

return redirect(url_for('movies_list'))
```