

Desarrollo de una API

Vamos a crear una API robusta utilizando el lenguaje de programación Python y el framework Flask.

El proyecto se divide en cuatro partes, comenzando con la configuración del entorno mediante la creación de un entorno virtual con virtualenv.

A continuación, construimos la API principal, destinada a administrar una base de datos de películas que

incluirá un campo adicional para el género de cada obra cinematográfica.

La segunda parte del proyecto implica la implementación de la funcionalidad para consultar las películas

por género, aprovechando las capacidades flexibles de Flask.

La tercera fase presenta una integración interesante al conectar nuestra API cinematográfica con una

API externa de feriados, utilizando la información sobre las próximas fechas festivas para recomendar

películas que se ajusten al género solicitado para ese día en particular.

Finalmente, la actividad concluye con la evaluación de la API, donde se considerarán aspectos cruciales

como la respuesta de la API, códigos de estado HTTP, escalabilidad y seguridad. Se espera que se realicen pruebas exhaustivas, asegurándose de que la API funcione eficientemente en

diversas situaciones.

Parte 1: Configuración del entorno e instalación de librerías

Crear un entorno virtual de Python utilizando venv.

```
python3 -m venv .venv #
```

En lugar de .venv puede ser cualquier nombre, pero es un estándar

Activa el entorno virtual

```
source .venv/bin/activate
```

cuando lo quieras dejar de usar solo escribe

```
deactivate
```

con el venv activo vamos a instalar las librerías necesarias.

```
pip install -r requirements.txt
```

Parte 2: Creación de una API con Flask

1. Complete las funcionalidades de la api que se les entrega en main.py
 - a. Lógica para buscar la película por su ID y devolver sus detalles
 - b. Lógica para buscar la película por su ID y actualizar sus detalles
 - c. Lógica para buscar la película por su ID y eliminarla
2. Implementa la funcionalidad para devolver el listado de películas de un género específico.
3. Implementa la funcionalidad de búsqueda de películas, devolviendo la lista de películas que
tengan determinado string en el título.
4. Implementa la funcionalidad de sugerir una película aleatoria
5. Implementa la funcionalidad de sugerir una película aleatoria según género

Recomendaciones, para probar el funcionamiento de la api pueden utilizar curl o postman.

- [Cómo probar apis con curl](#)
- [Cómo probar apis con postman](#)

Parte 3: Consumo de una API externa

Integra tu API con una API externa de feriados. Puedes utilizar la siguiente API: [API de Feriados](#).

Se les entrega una aplicación de Python que utiliza la API de nolaborables.com.ar para obtener información sobre los feriados en Argentina. La aplicación busca y muestra el próximo feriado disponible.

- a) Modifica el código para que agregar la opción de buscar feriados por tipo:
inamovible | trasladable | nolaborable | puente
- b) En la api de películas Utiliza la API de feriados para agregar la siguiente funcionalidad:
Obtener la próxima fecha de feriado y recomendar una película que se ajuste al género solicitado para ese día.
El comportamiento debería ser el siguiente:
Pregunta en lenguaje humano “Sugeríme una película de DRAMA para ver el próximo feriado”
Respuesta en lenguaje humano: “El próximo feriado es el xxxx con motivo yyyy , Te sugiero ver la película de DRAMA <título de película>
Obviamente no tienen que entregar este texto, el ejemplo es solo de comportamiento, deberán hacer las consultas usando las url de la api y las respuestas en formato json con los campos correspondientes.

Parte 4: Evaluación de la API

1. Usar el archivo test.py o test_pytest.py
2. Agregar tests para las funcionalidades nuevas
3. Replicar los test en postman, sugerencia, para testear en local usen la versión desktop