Laboratorio 1: Desarrollo de una API

Cátedra de Redes y Sistemas Distribuidos

Objetivos

- Utilizar y diseñar Apis, que corresponden a la programación en la capa de aplicación de redes según el modelo OSI de redes
- Incorporar fluidez y buenos hábitos en la programación con python.

Api para gestión de información

En esta actividad, 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 los participantes realicen pruebas exhaustivas, asegurándose de que la API funcione eficientemente en diversas situaciones. La entrega del proyecto incluirá un informe detallado documentando el proceso de creación, decisiones de diseño, tecnologías utilizadas, resultados de la evaluación y posibles mejoras o expansiones para la API.

Parte 1: Configuración del entorno e instalación de librerías

1. Crea 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. implemente la funcionalidad de búsqueda de películas, devolviendo la lista de películas que tengan determinado string en el título.
4. Implemente la funcionalidad de sugerir una película aleatoria
5. Implemente 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 , en este link pueden encontrar toda la documentación de la api.

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 “Sugerime 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 pelicula de DRAMA <titulo de pelicula>

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

Entrega

Junto con el código deberá entregar una presentación (tipo powerpoint) y un video de 10 minutos les damos una estructura de base como idea, pero pueden modificarla/ ampliarla.

1. **Introducción al proyecto**:
    a. Presenta brevemente el contexto del proyecto y sus objetivos.
    b. Explica la importancia de desarrollar una API robusta y bien diseñada.
2. **Configuración del entorno e instalación de librerías**:
    a. Explica por qué es importante utilizar un entorno virtual de programación.
    b. Muestra cómo configurar un entorno virtual de Python usando venv.
    c. Explica cómo activar y desactivar el entorno virtual.
    d. Guía sobre la instalación de las librerías necesarias utilizando pip.
3. **Creación de una API con Flask**:
    a. Describe las funcionalidades básicas de la API a desarrollar.
    b. Muestra cómo completaron las funcionalidades proporcionadas en el archivo main.py.
    c. Explica cada una de las partes de la API y su propósito.
4. **Pruebas de la API**:
    a. Presenta las recomendaciones para probar la API utilizando curl o Postman.
    b. Explica cómo realizar pruebas con curl.
    c. Muestra cómo realizar pruebas con Postman y replica los mismos tests que se realizan con curl.
5. **Consumo de una API externa**:
    a. Introduce la integración con la API externa de feriados.
    b. Explica cómo modificar el código existente para agregar funcionalidades relacionadas con los feriados.
    c. Demuestra cómo utilizar la API de feriados para obtener información y recomendar películas.
6. **Evaluación de la API**:
    a. Explica la importancia de la evaluación de la API.
    b. Presenta el archivo test.py o test_pytest.py y muestra cómo agregar tests para las nuevas funcionalidades.
    c. Reproduce los mismos tests en Postman para verificar la funcionalidad desde el punto de vista del cliente.
7. **Conclusiones y próximos pasos**:
    a. Resalta los resultados de la evaluación de la API.
    b. Propone posibles mejoras o ampliaciones para la API.
    c. Concluye el video resumiendo los principales puntos y agradeciendo por la atención.

¡Buena suerte con el laboratorio!