Universidad Tecnológica de Xicotepec de Juárez

**Ingeniería en Desarrollo y Gestión de Software**

**SWAGGER EN PYTHON**

**Jose Luis Galindo Picazo**

**Docente: MTI. Marco Antonio Ramírez**

**9° “A”**

Swagger

Swagger es una especificación abierta para definir las API REST.

Un documento Swagger es el equivalente de la API REST de un documento WSDL para un servicio web basado en SOAP.

El documento Swagger especifica la lista de recursos que están disponibles en la API REST y las operaciones que se pueden llamar en esos recursos. El documento Swagger también especifica la lista de parámetros de una operación, incluido el nombre y el tipo de los parámetros, si los parámetros son necesarios o opcionales, e información sobre los valores aceptables para dichos parámetros. Además, el documento Swagger puede incluir el esquema JSON que describe la estructura del cuerpo de solicitud que se envía a una operación en una API REST, y el esquema JSON describe la estructura de los cuerpos de respuesta que se devuelven de una operación.

Los documentos Swagger deben estar en formato JSON con la extensión de archivo .json o en formato YAML con la extensión de archivo .yaml o .yml.

Hay una gama de herramientas de terceros disponibles para que pueda utilizar con documentos Swagger:

**Swagger Editor**

Le ayuda a crear un documento Swagger desde un navegador web proporcionando una vista de lado a lado del documento Swagger y las definiciones de API REST resultantes.

**IU de Swagger**

Permite visualizar y probar una API REST definida con Swagger desde cualquier navegador web. Las funciones de pruebas incorporadas le permiten especificar las entradas de una operación definida en esa API REST, llamar a dicha operación desde el navegador web e inspeccionar los resultados de llamar a dicha operación.

**Swagger Codegen**

Genera un SDK (Software Development Kit) en varios idiomas, incluidos Java™, Objective-C, PHP y Python, desde un documento Swagger para una API REST. El SDK resultante se puede utilizar para incorporar llamadas a las operaciones en dicha API REST en un programa de software escrito en uno de los idiomas soportados, sin necesidad de manejar el transporte HTTP subyacente.

**¿Cómo Empezar con FastAPI?**

Asegúrate de que tienes un editor de texto/IDE de programador, como [Visual Studio Code](https://code.visualstudio.com/download). Otras opciones son [Sublime Text](https://kinsta.com/es/blog/como-usar-sublime-text/) y [Espresso](https://www.espressoapp.com/" \t "_blank).

Es una práctica habitual tener tus aplicaciones de Python y sus instancias funcionando en entornos virtuales. Los entornos virtuales permiten que se ejecuten simultáneamente diferentes conjuntos de paquetes y configuraciones, y evitan conflictos debidos a versiones de paquetes incompatibles.

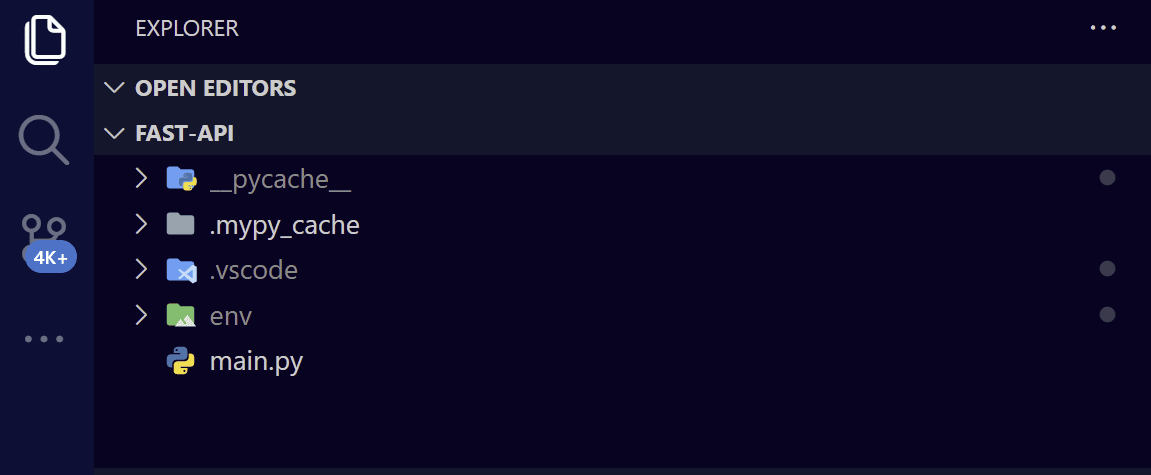
Instala FastAPI:

$ pip3 install fastapi

FastAPI es un marco para construir APIs, pero para probar tus APIs necesitarás un servidor web local. [Uvicorn](https://www.uvicorn.org/" \t "_blank) es un servidor web de interfaz de puerta de enlace de servidor asíncrono (ASGI) para Python, muy rápido, que es ideal para el desarrollo. Para instalar Uvicorn, ejecuta este comando:

$ pip3 install "uvicorn[standard]"

Tras la instalación, crea un archivo llamado main.py en el directorio de trabajo de tu proyecto. Este archivo será el punto de entrada de tu aplicación.



**Documentación Interactiva de la API de FastAPI**

FastAPI utiliza [Swagger UI](https://github.com/swagger-api/swagger-ui) para proporcionar documentación interactiva automática de la API. Para acceder a ella, navega a http://localhost:8000/docs y verás una pantalla con todos tus endpoints, métodos y esquemas.

