

Trabajo Práctico: Introducción a APIs, JSON y Postman:

API REST (Interfaz de Programación de Aplicaciones Representacional): Un API REST es un conjunto de reglas y convenciones que permiten a las aplicaciones comunicarse entre sí a través de la web. Proporciona una forma estandarizada de solicitar y enviar datos a través de HTTP. Los recursos en una API REST se representan como URLs, y las operaciones se realizan utilizando los métodos HTTP, como GET, POST, PUT y DELETE.

JSON (JavaScript Object Notation): JSON es un formato de intercambio de datos ligero y legible por humanos. Se utiliza para estructurar datos en un formato fácil de entender para las aplicaciones. Los datos se representan como pares clave-valor y se utilizan comúnmente en las respuestas de las APIs.

Postman: Postman es una herramienta que facilita el desarrollo y la prueba de APIs. Permite enviar solicitudes HTTP a una API y visualizar las respuestas de manera eficiente. También es útil para automatizar pruebas y colaborar en equipos de desarrollo.

Framework Flask: Flask es un framework web de Python que facilita la creación de aplicaciones web y APIs de manera sencilla y rápida.

Tarea - Creación de una API con Flask y MySQL:

Paso 1: Configuración de la Base de Datos:

Comenzaremos configurando una base de datos MariaDB. Puedes utilizar cualquier base de datos que prefieras. Crea una base de datos y una tabla sencilla para este proyecto.

Paso 2: Creación de la API con Flask:

Usaremos Flask para crear nuestra API. Aquí están los puntos clave:

Listar Todos los Elementos: Crea una ruta que responda a solicitudes GET y devuelva una lista de todos los elementos en la base de datos.

Listar por ID: Crea una ruta que permita buscar elementos por su ID.

Crear un Nuevo Elemento: Crea una ruta que permita agregar un nuevo elemento a la base de datos. Utiliza solicitudes POST.

Modificar un elemento existente: Crea una ruta que permita actualizar elementos existentes en la base de datos. Utiliza solicitudes PUT.

Eliminar un elemento: Crea una ruta que permita eliminar elementos de la base de datos. Utiliza solicitudes DELETE.

Paso 3: Prueba con Postman:

Después de crear cada ruta de la API, utiliza Postman para realizar pruebas y asegurarte de que la API funcione según lo esperado. Puedes enviar solicitudes GET, POST, PUT y DELETE para probar cada uno de los puntos anteriores.

Paso 4: Repositorio en GitHub:

Sube tu proyecto a un repositorio en GitHub para que otros puedan acceder a él.

Este proyecto te permitirá familiarizarte con la creación de APIs RESTful utilizando Flask y la gestión de una base de datos MySQL. Además, te ayudará a comprender la importancia de probar y documentar tu trabajo.