

Seminario de Lenguajes

Opción: PHP, React y API Rest

2023

API REST / Slim

Slim es un micro framework de PHP que se utiliza para crear aplicaciones web y APIs REST.

Las API REST siguen el modelo CRUD, que significa Create (Crear), Read (Leer), Update (Actualizar) y Delete (Eliminar)

Los recursos de una API REST se identifican mediante endpoints o URI (Uniform Resource Identifier). Cada recurso tiene una dirección única a la que se puede acceder para realizar operaciones sobre él.

Generalmente, se utiliza JSON (JavaScript Object Notation) como formato para el intercambio de datos entre el cliente y el servidor.

SEGUNDA ENTREGA

En esta segunda entrega, continuaremos con la entrega anterior, pero centrándonos solo en la implementación realizada del lado del servidor. Para esto deberán tener creado un proyecto utilizando el framework Slim (Ver archivo "Instalación de Framework Slim")

- 1) Crear una clase llamada DB que contenga un constructor que realice la conexión a la base de datos utilizando PDO
Crear la siguiente estructura de carpetas.
src/Models/Db.php
- 2) Diseñar una API Rest que cumpla con los siguientes ítems:

IMPORTANTE: en todos los casos deberán devolver el estado de la petición realizada

- 200 OK: Indica que la solicitud se procesó correctamente y se devolvió una respuesta exitosa.
- 400 Bad Request: Se utiliza cuando la solicitud enviada por el cliente es incorrecta o no se puede procesar debido a parámetros faltantes, valores inválidos o problemas de formato.
- 404 Not Found: Indica que el recurso solicitado no se pudo encontrar en el servidor.
- 409 Conflict: Indica que el registro no pudo ser eliminado.

En los casos en que se produce un error, además, deberán devolver un mensaje indicando por qué no se pudo realizar la acción

- a) Crear un nuevo ítem: implementar un endpoint para crear un nuevo ítem en la tabla de items_menu. El endpoint debe permitir enviar el nombre, el precio, el tipo (comida o bebida) y la foto.
- b) Actualizar información de un ítem: implementar un endpoint para actualizar la información de un ítem existente en la tabla items_menu. El endpoint debe permitir enviar el id y los campos que se quieran actualizar.
- c) Eliminar un ítem: el endpoint debe permitir enviar el id del ítem y eliminarlo de la tabla solo si no existen pedidos para ese ítem.
- d) Obtener los ítems ordenados por precio: implemente un endpoint para obtener los ítems de la tabla, permitiendo pasar los filtros por tipo (comida/bebida) y por nombre de producto (parcial o totalmente, por ejemplo, "gu" encontraría agua y hamburguesa) así como también si el orden será ascendente o descendente.
Si no paso los filtros, todos los registros serán devueltos. Si no paso el orden, por defecto será ascendente.
- e) Obtener todos los pedidos del más nuevo al más viejo
- f) Crear un nuevo pedido: implementar un endpoint para crear un nuevo pedido en la tabla de pedidos. El endpoint debe permitir enviar el número de mesa, el id del ítem del menú y un comentario (opcionalmente).
- g) Obtener un pedido: implementar un endpoint que permita obtener un pedido a partir de su id. Este debe mostrar todos los datos del pedido, incluyendo el nombre del ítem del menú, precio, tipo y foto
- h) Eliminar un pedido: el endpoint debe borrar un pedido a partir de su id.

Metodología para la entrega en Ideas

- Hacer un archivo zip o rar con todos los archivos del proyecto
- Subir a la sección del grupo asignada
- **Fecha de entrega 9/11**