

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.

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 género: implementar un endpoint para crear un nuevo género en la tabla de géneros. El endpoint debe permitir enviar el nombre.
- b) Actualizar información de un género: implementar un endpoint para actualizar la información de un género existente en la tabla de géneros. El endpoint debe permitir enviar el id y los campos que se quieran actualizar
- c) Eliminar un género: el endpoint debe permitir enviar el id del género y eliminarlo de la tabla.
- d) Obtener todos los géneros: implemente un endpoint para obtener todos los géneros de la tabla.
- e) Crear una nueva plataforma: implementar un endpoint para crear una nueva plataforma en la tabla de plataformas. El endpoint debe permitir enviar el nombre.
- f) Actualizar información de una plataforma: implementar un endpoint para actualizar la información de una plataforma existente en la tabla de plataformas. El endpoint debe permitir enviar el id y los campos que se quieran actualizar
- g) Eliminar una plataforma: el endpoint debe permitir enviar el id de la plataforma y eliminarla de la tabla.
- h) Obtener todos los géneros: implemente un endpoint para obtener todos los géneros de la tabla.
- i) Crear un nuevo juego: implementar un endpoint para crear un nuevo juego en la tabla de juegos. El endpoint debe permitir enviar el nombre, imagen, descripción, plataforma, URL y género.
- j) Actualizar información de un juego: implementar un endpoint para actualizar la información de un juego existente en la tabla de juegos. El endpoint debe permitir enviar el id y los campos que se quieran actualizar
- k) Eliminar un juego: el endpoint debe permitir enviar el id del juego y eliminarlo de la tabla.
- l) Obtener todos los juegos: implemente un endpoint para obtener todos los juegos de la tabla.
- m) Buscar juegos: implementar un endpoint que permita buscar juegos por nombre, plataforma y género. El endpoint deberá aceptar un nombre, un id de género, un id de plataforma y un orden por nombre (ASC o DESC)

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: 1/6**