

Seminario de Lenguajes

Opción: PHP, React y API Rest

2024

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), Retrieve (Recuperar), 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.

ACLARACIONES

Se deben tener las siguientes consideraciones:

- Todos los endpoints deberán tener los métodos y nombres que se detallan en cada ítem.
- Las eliminaciones sólo se podrán ejecutar correctamente en los casos en que el dato no esté siendo utilizado en otra tabla, caso contrario se debe informar el error.
- Después del **AppFactory create** de slim agregar **\$app->addBodyParsingMiddleware();** para facilitar la extracción de datos en formato json del body de cada service con **\$data = \$request->getParsedBody();**

IMPORTANTE

En todos los casos, los endpoints 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.
- **401 Unauthorized:** indica que un usuario no puede realizar la acción en cuestión porque no está autorizado.
- **404 Not Found:** Indica que el recurso solicitado no se pudo encontrar en el servidor. No existe el endpoint o el recurso (por ejemplo, se buscó un juego con un id que no existe).
- **409 Conflict:** Indica que el registro no pudo ser eliminado.

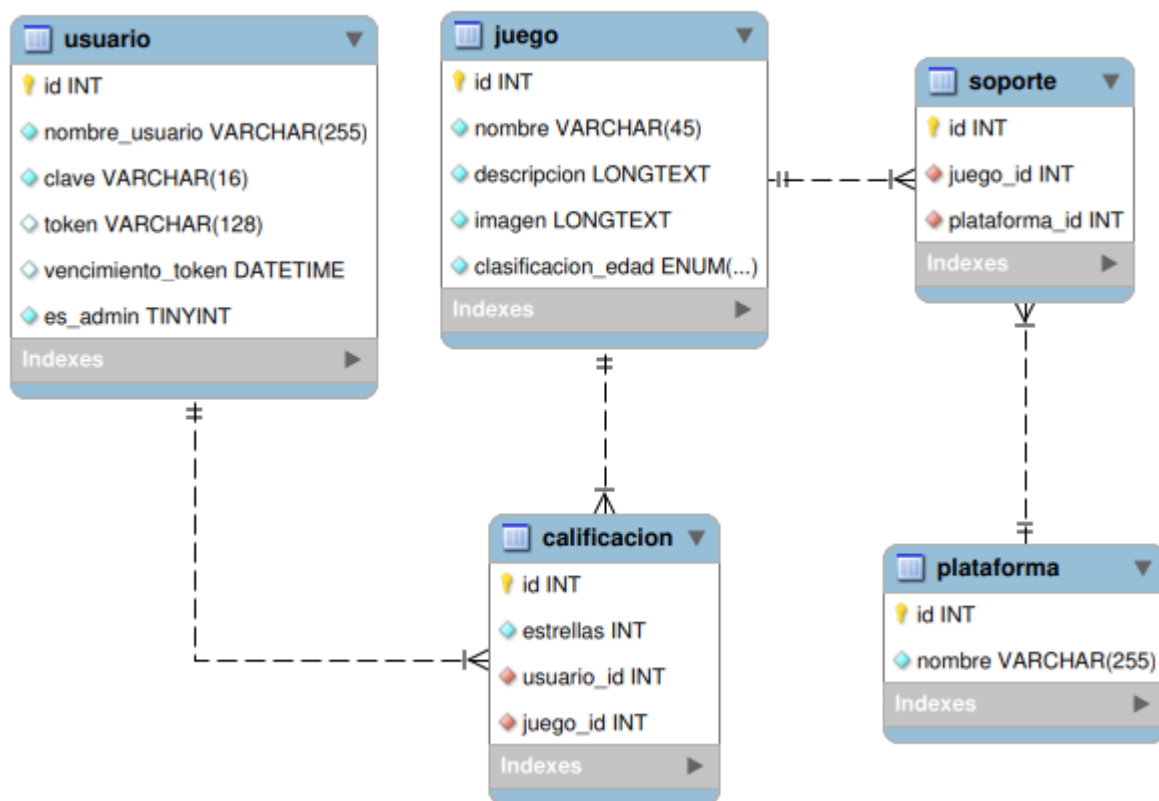
En los casos en que se produce un error, además, deberán devolver un mensaje indicando por qué no se puede realizar la acción. Es necesario que el código de respuesta se ponga en el servicio para que dicho código sea enviado en un *header http* sino no será interpretado correctamente cuando se construya el front-end.

Por ejemplo:

- No se puede eliminar un soporte porque hay juegos usándolo.

PROYECTO

Se desea desarrollar una aplicación web que permita puntuar videojuegos. En dicha aplicación un usuario logueado podrá puntuar un videojuego en una escala de 1 a 5 estrellas según su gusto. El sistema mostrará un listado (paginado de a 5 juegos por vez) donde se podrá ver la calificación promedio de un juego. El listado de juegos podrá filtrarse por texto sobre el nombre del juego, por ejemplo, si el usuario busca “lo” podrían aparecer en el listado *Valorant* y *King of Glory*. Además, también se pueden filtrar por la plataforma en la que se puede jugar (PS, Xbox, PC, Android u otro) y por la clasificación por edad: si se elige ATP aparecen solo esos pero si pone +13 aparecen los +13 y los ATP y si pone +18 solo aparecen todos.



Además los usuarios podrán realizar el registro en la aplicación, completando su nombre de usuario y clave. Por otro lado, un usuario administrador de la plataforma podrá crear, modificar y borrar juegos.

En el registro de un usuario se deberá validar que:

- el nombre de usuario tenga por lo menos 6 caracteres y no más de 20. Que sean únicamente alfanuméricos. Además, se debe verificar que ese nombre de usuario no esté en uso.

- la clave tenga por lo menos 8 caracteres y que tenga mayúsculas, minúsculas, números y caracteres especiales.

En el alta de una calificación se debe verificar que sea un número entero entre 1 y 5.

En el alta y modificación de un juego se debe verificar que el nombre del juego no tenga más de 45 caracteres y que la clasificación por edad sea una de las tres válidas ('ATP', '+13', '+18'). Además, se debe convertir la imagen a base 64 para almacenarla en la base de datos.

BACKEND

Para la aplicación, es necesario desarrollar los siguientes endpoints:

Login

- **POST /login:** Verificar credenciales y retornar token.
 - Cuerpo de la solicitud: **JSON** con los campos **nombre_usuario** y **clave**.
 - **Respuesta:**
 - **Éxito:** Código 200 OK, JSON con un token de acceso (**token**). Se genera un nuevo **token**, se guarda en la base de datos y se pone **vencimiento_token** a 1 hora en el futuro.
 - **Fallo:** Código de estado 401 Unauthorized, con un mensaje de error.

Luego, cuando se requiera realizar una acción autorizada (como por ejemplo, puntuar un juego) se deberá invocar al endpoint con el token recibido.

- **POST /register:** Agrega un nuevo usuario con su nombre de usuario y clave.

Usuarios

- **POST /usuario:** Crear un nuevo usuario.
- **PUT /usuario/{id}:** Editar un usuario existente.
- **DELETE /usuario/{id}:** Eliminar un usuario.
- **GET /usuario/{id}:** Obtener información de un usuario específico.
 - En los últimos 3 casos (donde se recibe el id) se debe validar que el usuario se haya logueado.

Juegos

- **GET**
/juegos?pagina={pagina}&clasificacion={clasificacion}&texto={texto}&plataforma={plataforma} Listar los juegos de la página según los parámetros de búsqueda incluyendo la puntuación promedio del juego.
- **GET /juegos/{id}**: Obtener información de un juego específico y su listado de calificaciones.
- **POST /juego**: da de alta un nuevo juego. Solo lo puede hacer un usuario logueado y que sea administrador.
- **PUT /juego/{id}**: actualiza los datos de un juego existente. Solo lo puede hacer un usuario logueado y que sea administrador.
- **DELETE /juego/{id}**: borra el juego siempre y cuando no tenga calificaciones. Solo lo puede hacer un usuario logueado y que sea administrador.

Calificaciones

- **POST /calificacion**: Crear una nueva calificación. Solo lo puede hacer un usuario logueado.
- **PUT /calificacion/{id}**: Editar una calificación existente. Solo lo puede hacer un usuario logueado.
 - Éxito: Código 200 OK, se actualiza el valor de puntuación del juego solo si el token coincide con el almacenado para ese usuario y el token no está vencido.
 - Fallo: Código de estado 401 Unauthorized, con un mensaje de error.
- **DELETE /calificacion/{id}**: Eliminar una calificación. Solo lo puede hacer un usuario logueado.

METODOLOGÍA PARA LA ENTREGA EN IDEAS

- Hacer un archivo zip o rar con todos los archivos del proyecto *sin incluir la carpeta **vendor***. Si se utilizan librerías externas agregar un README indicando para qué las utilizaron.
- Subir a la sección del grupo asignada (En el repositorio general **NO** van)
- **Fecha de entrega: 29/9**