**Introducción:**

La API Gestor Mapas es una herramienta que permite gestionar mapas de videojuegos mediante un servidor web utilizando el framework Flask de Python. Permite obtener información de mapas almacenados en una base de datos MySQL, así como añadir nuevos mapas, modificar la información de los mapas existentes y eliminar mapas de la base de datos.

**Endpoints:**

La API Gestor Mapas ofrece los siguientes endpoints:

**Obtener todos los mapas:**

Para obtener todos los mapas almacenados en la base de datos, envíe una solicitud GET al endpoint /mapas/. La respuesta contendrá una lista de objetos JSON que representan cada uno de los mapas almacenados en la base de datos.

Texto

Descripción generada automáticamente

**Obtener un mapa específico:**

Para obtener información de un mapa específico, envíe una solicitud GET al endpoint /mapas/<id>, donde <id> es el identificador único del mapa que desea obtener.

Texto

Descripción generada automáticamente

**Añadir un nuevo mapa:**

Para añadir un nuevo mapa, envíe una solicitud POST al endpoint /insertarMapa/ con un objeto JSON que contenga los siguientes campos:

**id**: el identificador único del mapa que desea añadir.

**nombre**: el nombre del mapa que desea añadir.

**descripcion**: una descripción del mapa que desea añadir.

**imagen**: la ruta de la imagen del mapa que desea añadir.

Captura de pantalla de computadora

Descripción generada automáticamente

**Modificar un mapa existente:**

Para modificar la información de un mapa existente, envíe una solicitud PUT al endpoint /modificarMapa/ con un objeto JSON que contenga los siguientes campos:

**id**: el identificador único del mapa que desea modificar.

**nombre**: el nuevo nombre que desea asignar al mapa.

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

Texto

Descripción generada automáticamente**Eliminar un mapa existente:**

Para eliminar un mapa existente, envíe una solicitud DELETE al endpoint /eliminarMapa/ con un objeto JSON que contenga el siguiente campo:

**id**: el identificador único del mapa

que desea eliminar.

**Parámetros:**

Los siguientes parámetros se utilizan en los endpoints de la API:

**id**: identificador único del mapa (número entero de hasta 15 dígitos)

**nombre**: nombre del mapa (cadena de caracteres con un máximo de 20 caracteres)

**descripcion**: descripción del mapa (cadena de caracteres con un máximo de 650

caracteres).

**imagen**: ubicación de la imagen del mapa (cadena de caracteres con un máximo de 80

caracteres)

**Respuestas:**

La API de Mapas de Call of Duty devuelve respuestas en formato JSON con los siguientes campos:

**mensaje**: mensaje que indica si la operación se ha realizado con éxito o si se ha

producido un error.

**datos**: información sobre los mapas solicitados.

**Ejemplos de Uso:**

Texto

Descripción generada automáticamente**Texto

Descripción generada automáticamenteEjemplo 1:** Recuperar todos los mapas. **Ejemplo 2**: Recuperar un mapa específico por su ID

**Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteEjemplo 3:** Insertar un nuevo mapa. **Ejemplo 4:** Modificar un mapa existente

**Texto

Descripción generada automáticamente Ejemplo 5:** Eliminar un mapa existente.

**Creación de la base de datos COD:**

Para crear la base de datos utilizada en la API "Gestor Mapas", se siguen los siguientes pasos:

1. Abrir una herramienta de gestión de bases de datos MySQL, como XAMPP o MySQL Workbench.
2. Crear una nueva base de datos con el nombre "cod". Esto se puede hacer ejecutando el siguiente comando SQL: CREATE DATABASE cod;
3. Seleccionar la base de datos "cod" para poder crear la tabla "mapas". Esto se puede hacer ejecutando el siguiente comando SQL: USE cod;
4. Crear la tabla "mapas" con los campos "ID", "NOMBRE", "DESCRIPCIÓN" e "IMAGEN". Esto se puede hacer ejecutando el siguiente comando SQL:

Texto

Descripción generada automáticamente

**Referencias:**

**Flask**: <https://flask.palletsprojects.com/en/2.1.x/>

**Flask-CORS:** <https://flask-cors.readthedocs.io/en/latest/>

**MySQL Connector/Python:** <https://dev.mysql.com/doc/connector-python/en/>

**Consideraciones importantes:**

Al utilizar la API, es importante tener en cuenta los siguientes aspectos:

Seguridad: La API no incluye autenticación y autorización, lo que significa que cualquier persona que tenga acceso a la URL de la API puede interactuar con ella. Es importante agregar autenticación y autorización antes de poner la API en producción para proteger los datos almacenados en la base de datos.

Escalabilidad: La API actual no incluye paginación, lo que significa que, si hay muchos datos almacenados en la base de datos, la API puede devolver un gran conjunto de resultados, lo que puede afectar el rendimiento. Es importante implementar la paginación para devolver grandes conjuntos de resultados de manera eficiente.

**CONCLUSIONES:**

La API "Gestor Mapas" es una aplicación útil para la gestión de mapas de una web sobre COD zombies que permite a los usuarios acceder y modificar la información de los mapas almacenados en la base de datos. La API utiliza Flask para crear los endpoints y la biblioteca MySQL para conectarse a la base de datos MySQL.

Es cierto que falta seguridad para que los usuarios solo puedan verla y que los apartados de modificar, insertar y eliminar estén solo abiertas a los administradores de la web y hacer un front(El cual ya solo falta darle los últimos retoques a los html).