

ÍNDICE

ÍNDICE	1
DOCUMENTOS	2
ESTRUCTURA DEL CÓDIGO	3
Carpeta “datasources”	3
Carpeta “models”	3
Carpeta “repositories”	3
Carpeta “controllers”	3
Carpeta “services”	3
Levantar API	4
En local:	4
En el servidor:	4

DOCUMENTOS

En la carpeta “**docs**” podemos encontrar el sql empleado para crear la base de datos, el pdf con la idea con la que se basó para crear la api, e imágenes del esquema relacional de la base de datos.

En el documento, encontraréis la idea original, pero para el curso 2023/2024 se decidió eliminar la funcionalidad de los objetos y hacer que los ataques solo puedan tener una sola posición (no un array).

Tanto para el sql como para el esquema relacional, encontraréis la versión normal, y la versión 2. Siendo la primera con los objetos. Pero, el fichero “**SQL_server.sql**” es el más actualizado y el usado por el servidor.

ESTRUCTURA DEL CÓDIGO

Dentro de la carpeta “src” hay todo lo importante para la API. Cualquier duda, consultar la documentación de Loopback4

Carpeta “datasources”

En esta carpeta se encuentran los ficheros para la conexión con la base de datos.
En el fichero “mysql.datasource.ts” se establece la base de datos, la contraseña...

Carpeta “models”

Aquí se declaran las entidades de la base de datos y sus atributos. Cada fichero es una entidad.

Finalmente, se puede añadir las relaciones que tienen en el mismo fichero, por ejemplo:

```
@hasMany(() => Players_Games, {keyTo: 'game_ID', name:
"players_games"})
players_games?: Players_Games[];
```

Carpeta “repositories”

Los repositorios sirven de enlace con la base de datos y los models. Con ellos serán los que hagamos las solicitudes de la información de la base de datos en los controladores.

Carpeta “controllers”

Los controllers definen los endpoints de la api. En el constructor, se añaden los repositorios necesarios y los servicios que se deseen.

Para cada endpoint, se hace un método con este formato:

```
@post('path_endpoint', {
  /*información que aparecerá en el explorer/
})async nombreMetodo(/*parámetros para el usuario, ej:@param.path*/) {
  /*comprobaciones, peticiones a la base, establecer la respuesta*/
}
```

Carpeta “services”

Los ficheros creados se han usado como clases globales para usarlos en varios controladores como constantes. De esta forma evitamos la repetición de código y que esté todo más organizado.

Levantar API

En local:

En el terminal, en la carpeta del mismo proyecto, hacer “**npm start**”.

En el servidor:

Subir el proyecto sin la carpeta node_modules al servidor (Se puede usar filezilla o parecidos).

Una vez subido, recordar hacer “**npm install**”.

Para mantener la API encendida aunque se salga de la conexión de usuario con ssh en el servidor, usar la comanda “**npm start > log.txt 2>&1 &**”

En log.txt se escribirá los mensajes que la API muestra (los errores...)

Se creará un número de proceso (PID). Será conveniente guardarlo para poder guardar obtener información del proceso y para poderlo terminar con “**kill -9 PID**”.