

## SPRINGBOOT BACKEND

El proyecto para funcionar solo necesita compilarse ejecutándolo desde VScode o un IDE que tenga todo lo necesario para compilar java, ya tiene la configuración para la base de datos en Clever Cloud.

Para comprobar Endpoints se debe importar la colección de postman ubicada en la Ruta: /Docs/PostmanCollection del repositorio.

En caso de error al compilar verificar conexión a internet ya sea para las dependencias que se necesitan o falta de conexión a la base de datos.

## REACT FRONTEND

Usando la terminal accedemos a la ruta de la carpeta raíz e instalamos los paquetes y dependencias necesarias especificadas en el "package.json" usando "--npm i".

En caso de que algunas dependencias no sean reconocidas por el comando, debemos ejecutar individualmente los siguientes comandos:

```
--npm install react-router-dom.  
--npm install sweetalert2.
```

Para ejecutar finalmente debemos usar el comando `--npm run dev`.

## NESTJS MICROSERVICIO AUTENTIFICACIÓN

Para ejecutar el proyecto localmente, configura las variables de entorno necesarias para la base de datos y JWT. Instala las dependencias necesarias en la carpeta del microservicio: (npm install).

Copia el archivo .env.example a un nuevo archivo .env y configura las variables de entorno necesarias:  
cp .env.example .env

```
PORT=3000
DB_NAME=tu_nombre_de_base_de_datos
DB_USER=tu_usuario_de_base_de_datos
DB_PASS=tu_contraseña_de_base_de_datos
DB_HOST=tu_host_de_base_de_datos
DB_PORT=tu_puerto_de_base_de_datos
JWT_DURATION=3h
JWT_SECRET = contraseña ultra secreta, contraseña ultra secreta, contraseña ultra secreta
```

```
USER_ROLE = user
ADMIN_ROLE = admin
```

Estos pasos garantizan que tengas toda la configuración necesaria para ejecutar el proyecto localmente, ajustando las variables de entorno según las necesidades de tu entorno de desarrollo. -Levantar contenedor de base de datos.

En la raíz del proyecto hay un archivo docker-compose.yml, este contiene el manifiesto necesario para levantar la base de datos con el siguiente comando (docker-compose up).

### **-Conexión a la base de datos**

Puedes usar cualquier cliente de base de datos como dbeaver, usando las variables de entorno definidas en el archivo env o en el docker-compose.yml.

### **-Carga de información inicial**

Para que el proceso de register pueda ser exitoso se requiere ejecutar las siguientes queries en la base de datos para tener los roles creados:

```
INSERT into rol (id,name)values (1,'user');  
INSERT into rol (id,name)values (2,'admin');
```

### **-Levantar API**

Para levantar la api solo debes ejecutar el siguiente comando (npm run start:dev).