

Trabajo Práctico N°1 - Parte Práctica

Descripción del proyecto:

El proyecto consiste en una API desarrollada con NodeJS para la autenticación de usuarios utilizando Mysql para el almacenamiento y manipulación de datos de usuarios en una base.

La API cuenta con las librerías Sequelize y Mysql2 para la conexión con la base de datos. En lo que respecta a seguridad, las credenciales de acceso y los datos del servidor están protegidos con variables de entorno, a las que se accede con Dotenv, las claves de usuario están protegidas con bcrypt, que las encripta antes de cargarlas en la base de datos y luego las compara con las claves ingresadas cuando los usuarios desean loguearse. Las rutas solamente pueden ser accedidas por usuarios que se hayan logueado previamente, con excepción de la ruta de registro /register y la de logueo /login que deben ser accesibles para que los clientes puedan obtener un token y así acceder a los datos de la base.

Para el proyecto, se utilizará docker para crear un container que contenga la base de datos a la que ingresa la API. Para esta tarea, se utiliza un archivo docker-compose que inicia el servicio y carga la configuración del servidor:

```
## Version de sintaxis de compose
version: '3.1'

services:
  ## Contenedores a iniciar
  db:
    ## Imagen que utilizará
    image: mariadb
    ## Variables de entorno
    environment:
      ## Clave de usuario ROOT
      MARIADB_ROOT_PASSWORD: 'mysqlAPI'
      ## Base de datos a crear
      MARIADB_DATABASE: 'apiLogin'
      ## Añade un usuario adicional
      MARIADB_USER: 'apiAdmin'
      ## Contraseña del usuario anterior
      MARIADB_PASSWORD: 'mysqlAPI'
    ## Volúmenes a utilizar
    volumes:
      ## Almacén de datos de mariadb
      - mariadb_data:/var/lib/mysql
      ## Directorio para levantar la base de datos al inicio del container
      - ./db_init:/docker-entrypoint-initdb.d
    ## Puerto a utilizar
    ports:
      ## Se utiliza el puerto 27030 del equipo local y el 3306 del server mariadb
      - "27030:3306"

## Declaración de volúmenes
volumes:
  ## No se especifica otro directorio para este volumen
  mariadb_data:
```

Como se puede apreciar en la imagen, el servidor utiliza el puerto 27030. El otro puerto reservado es el 3579, el cual es utilizado por la API:

```
// Configuración de los endpoints
app.use('/user', UserRoutes);
app.use('/login', AuthRoute);
app.use('/register', registerRoute);

// Define el puerto de la api
const PORT = 3579
app.listen(PORT, () => {
  console.log(`Servidor corriendo en puerto ${PORT}`)
})
```

Endpoints:

Para interactuar con la api, se utilizan los siguientes endpoints:

- **Registrarse: (POST)**
 URL: http://localhost:3579/register
 Body:


```
{
      "username": "",
      "password": "",
      "email": "",
      "rol": "" (opcional)
    }
```
- **Iniciar sesión: (POST)**
 URL: http://localhost:3579/login/
 Body:


```
{
      "username": "",
      "password": "",
    }
```
- **Ver lista de usuarios: (GET)**
 URL: http://localhost:3579/users
- **Buscar usuario por nombre: (GET)**
 URL: http://localhost:3579/users/{username}
- **Borrar usuario: (DELETE)**
 URL: http://localhost:3579/users/{username}
- **Actualizar usuario: (PUT)**
 URL: http://localhost:3579/users/{username}
 Body: (Solamente colocar los datos que se quiere actualizar)


```
{
      "username": "",
      "password": "",
    }
```

```
    "email": "",  
    "rol": ""  
}
```