Sports Unit API

La siguiente API tiene la configuración y consultas para el modulo de la unidad deportiva.

Autores

- Jesus Manuel Leiva Bermudez - 20191020132
- Sergio David Paez Suarez - 20191020167

Tegnologías utilizadas

- La API se encuentra hecha en `Java` con el framework de `SpringBoot`

Dependencias

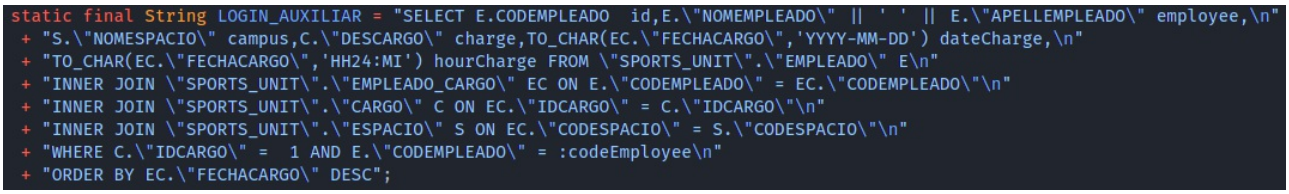
jdbc

Será nuestro intermedio entre las consultas e inserciones que deseamos realizar y el driver para conectar con la base de datos.

```java
@Repository
public class LoginRepository {

    @Autowired
    private JdbcTemplate jdbcTemplate;

    public List<LoginAuxiliarResponseDTO> loginAuxiliar(String codeEmployee) throws Exception {
        return this.jdbcTemplate.query(DatabaseOperations.LOGIN_AUXILIAR,
                BeanPropertyRowMapper.newInstance(mappedClass: LoginAuxiliarResponseDTO.class), codeEmployee);
    }
```



```java
static final String LOGIN_AUXILIAR = "SELECT E.CODEMPLEADO  id,E.\"NOMEMPLEADO\" || ' ' || E.\"APELLEMPLEADO\" employee,\n"
 + "S.\"NOMESPACIO\" campus,C.\"DESCARGO\" charge,TO_CHAR(EC.\"FECHACARGO\",'YYYY-MM-DD') dateCharge,\n"
 + "TO_CHAR(EC.\"FECHACARGO\",'HH24:MI') hourCharge FROM \"SPORTS_UNIT\".\"EMPLEADO\" E\n"
 + "INNER JOIN \"SPORTS_UNIT\".\"EMPLEADO_CARGO\" EC ON E.\"CODEMPLEADO\" = EC.\"CODEMPLEADO\"\n"
 + "INNER JOIN \"SPORTS_UNIT\".\"CARGO\" C ON EC.\"IDCARGO\" = C.\"IDCARGO\"\n"
 + "INNER JOIN \"SPORTS_UNIT\".\"ESPACIO\" S ON EC.\"CODESPACIO\" = S.\"CODESPACIO\"\n"
 + "WHERE C.\"IDCARGO\" =  1 AND E.\"CODEMPLEADO\" = :codeEmployee\n"
 + "ORDER BY EC.\"FECHACARGO\" DESC";
```

Configuración Base de Datos de forma manual

Para poder configurar la base de datos tiene que dirigirse a la carpeta `Oracle` que se encuentra entregada en este proyecto, alli se encuentran dos archivos

> El primero es el archivo `creation.sql` el cual contiene la definición de tablas y constraints para todas las tablas del modelo.

> El segundo es el archivo `inserts.sql` el cual es el archivo referente al segundo punto del modulo, el cual es un archivo que contiene todos los inserts iniciales para el funcionamiento de la aplicación.

deben ejecutarse estos archivos en el gestor de base de datos para poder usar la aplicación

Configuración mediante docker

Dentro de la carpeta `Oracle`, se encuentra un archivo denominado `docker-stack.yml`, contiene la definición del servicio a desplegar para configurar auto
máticamente el gestor de bases de datos y la estructura de las tablas, junto con algunos inserts por defecto.

Para desplegar el servicio de docker podemos usar dos comandos:

- Como servicio

```
docker stack deploy -c docker-stack.yml oracle
```

- Como un contenedor

```
docker-compose up -d
```

Configuración del proyecto

Para poder utilizar el proyecto es necesario configurar ciertas variables de entorno que este utiliza para asegurar el correcto funcionamiento y uso del modelo planteado para la base de datos.

```
spring.datasource.driver-class-name=oracle.jdbc.OracleDriver

spring.datasource.url=jdbc:oracle:thin:@localhost:1521/XE

spring.datasource.username=system

spring.datasource.password=m5S5AUC4ScE7

server.port=5000

server.servlet.context-path=/api/ud/v1
```

- El primer valor, el cual es `spring.datasource.driver-class-name` se refiere a el driver de configuración para que la API utilice `oracle`, como tal este valor se mantiene igual.
- La segunda propiedad (`spring.datasource.url`) se refiere a la url que se usa para que jdbc pueda conectarse a la base de datos.
- La tercera propiedad (`spring.datasource.username`) se refiere al usuario configurado para su base de datos
- La cuarta propiedad (`spring.datasource.password`) se utiliza para definir la contraseña del usuario a utilizar para conectar a la base de datos.

- La quinta propiedad (`server.port`) define el puerto por el cual seran expuestos los servicios de la api.
- La sexta propiedad (`server.servlet.context-path`) nos permite agregar un endpoint de contexto para los servicios de determinada versión, para nuestro caso utilizaremos por defecto /api/ud/v1.

Ejecución

El comando que se utiliza para iniciar el api en spring tanto para windows, como para linux será (Asegurese de tener instalado maven):

```
mvn spring-boot:run
```