

Parcial II

Daniel Felipe Caicedo Galvis

Universidad de Cundinamarca

Facultad de Ingeniería

William Alexander Matallana Porras

Bogotá D.C., Colombia

18 de febrero de 2025

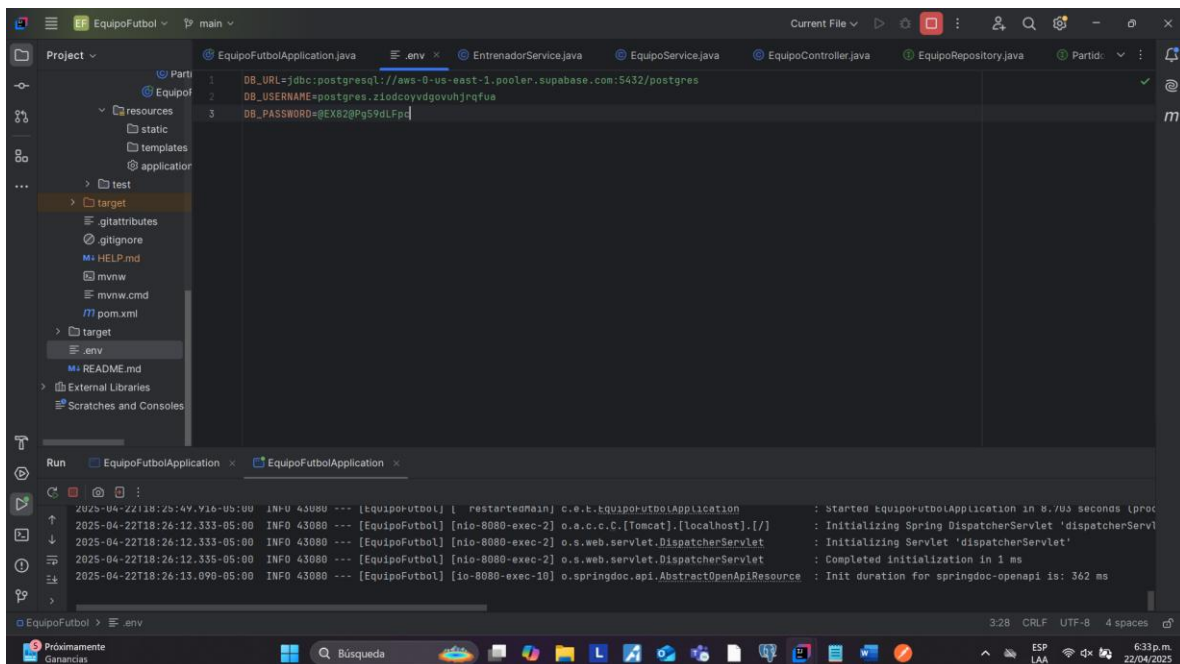
USO DE IA:

Porcentaje de uso: 20%

Se usó como consulta a las preguntas planteadas en el ejercicio y dio la estructura de la base de datos que se usó.

DESARROLLO:

1. Conexión con Supabase



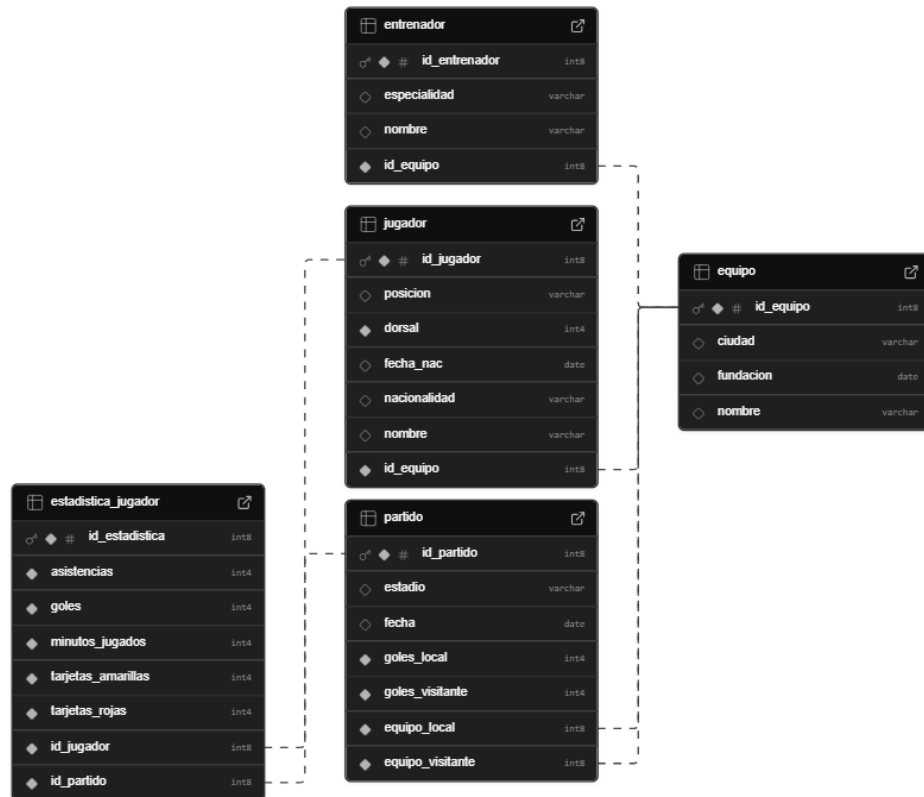
The screenshot shows an IDE with a project named 'EquipoFutbol'. The file explorer on the left shows a directory structure with 'resources' containing 'static', 'templates', and 'application'. The 'application' directory contains 'test', 'target', and '.env'. The 'target' directory contains 'gitattributes', 'gitignore', 'HELP.md', 'mvnw', 'mvnw.cmd', 'pom.xml', and 'target'. The '.env' file is open in the editor, showing the following configuration:

```
1 DB_URL=jdbc:postgresql://aws-0-us-east-1.pooler.supabase.com:5432/postgres
2 DB_USERNAME=postgres.ziodcoyvdgovuhjrqfua
3 DB_PASSWORD=qEX82Pg59dLfp
```

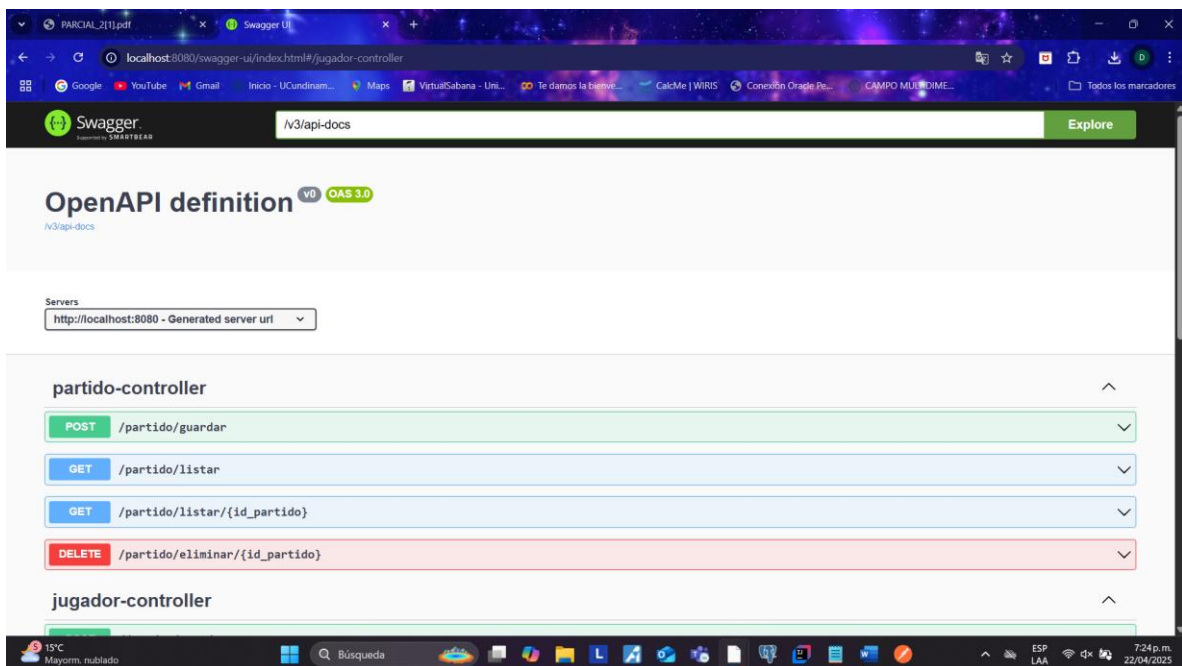
The Run console at the bottom shows the output of the application, indicating that it started successfully and initialized the Spring DispatcherServlet and Servlet 'dispatcherServlet'.

```
2025-04-22T18:26:12.333-05:00 INFO 43080 --- [EquipoFutbol] [nio-8080-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Started EquipoFutbolApplication in 0.703 seconds (process ID 43080)
2025-04-22T18:26:12.333-05:00 INFO 43080 --- [EquipoFutbol] [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-04-22T18:26:12.333-05:00 INFO 43080 --- [EquipoFutbol] [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-04-22T18:26:13.090-05:00 INFO 43080 --- [EquipoFutbol] [io-8080-exec-10] o.springdoc.api.AbstractOpenApiResource : Completed initialization in 1 ms
2025-04-22T18:26:13.090-05:00 INFO 43080 --- [EquipoFutbol] [io-8080-exec-10] o.springdoc.api.AbstractOpenApiResource : Init duration for springdoc-openapi is: 362 ms
```

2. Modelo de Datos



3. Documentar API con Swagger



PARCIAL_2[1].pdf x Swagger UI x +

localhost:8080/swagger-ui/index.html#/estadistica-jugador-controller

DELETE /partido/eliminar/{id_partido}

jugador-controller

- POST /jugador/guardar
- GET /jugador/listar
- GET /jugador/listar/{id_jugador}
- DELETE /jugador/eliminar/{id_jugador}

estadistica-jugador-controller

- POST /estadistica/guardar
- GET /estadistica/listar
- GET /estadistica/listar/{id_estadistica}
- DELETE /estadistica/eliminar/{id_estadistica}

equipo-controller

entrenador-controller

15°C Mayorm, nublado

15°C Mayorm, nublado

PARCIAL_2[1].pdf x Swagger UI x +

localhost:8080/swagger-ui/index.html#/entrenador-controller

DELETE /estadistica/eliminar/{id_estadistica}

equipo-controller

- POST /equipo/guardar
- GET /equipo/listar
- GET /equipo/listar/{id_equipo}
- DELETE /equipo/eliminar/{id_equipo}

entrenador-controller

- POST /entrenador/guardar
- GET /entrenador/listar
- GET /entrenador/listar/{id_entrenador}
- DELETE /entrenador/eliminar/{id_entrenador}

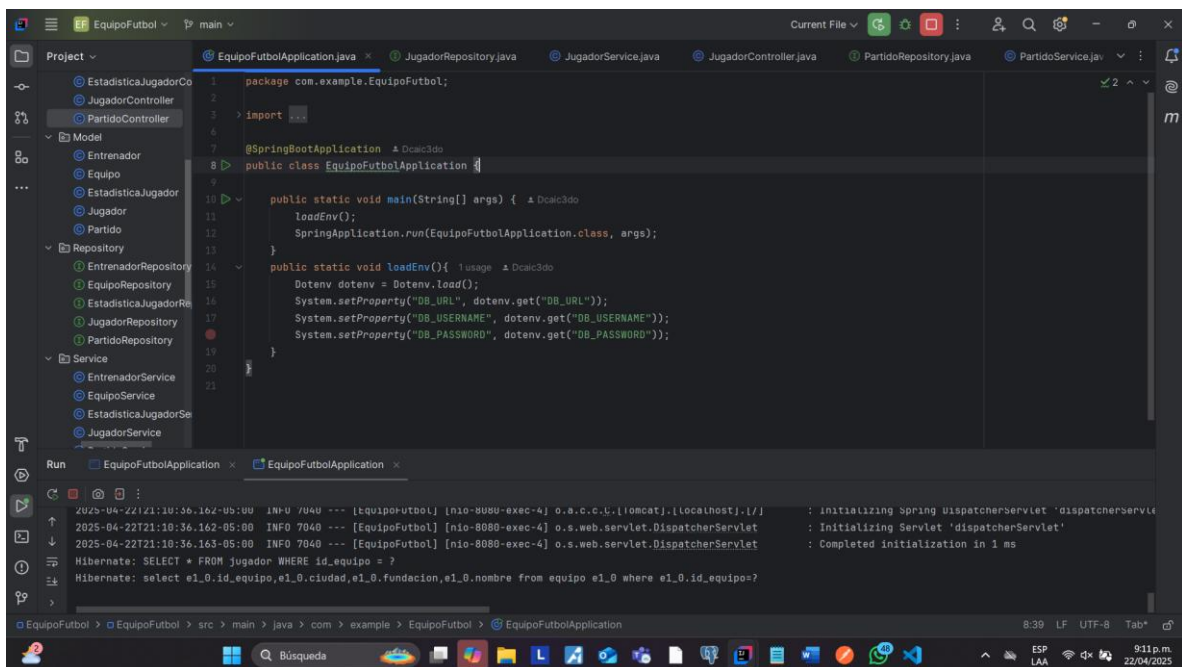
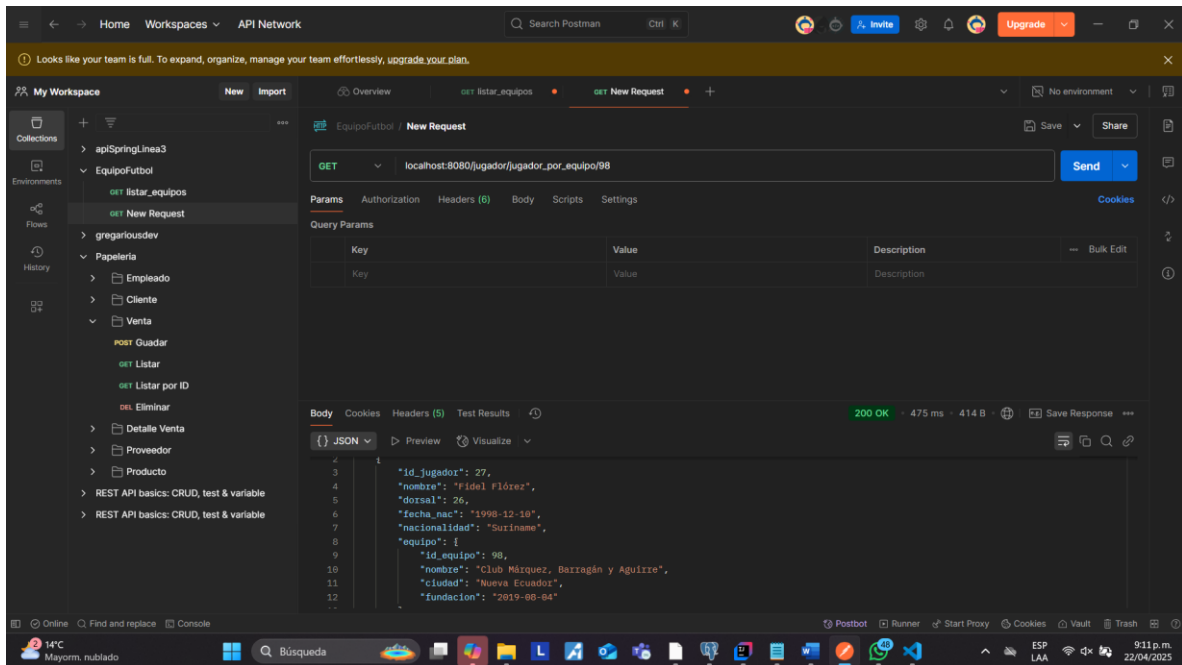
Schemas

15°C Mayorm, nublado

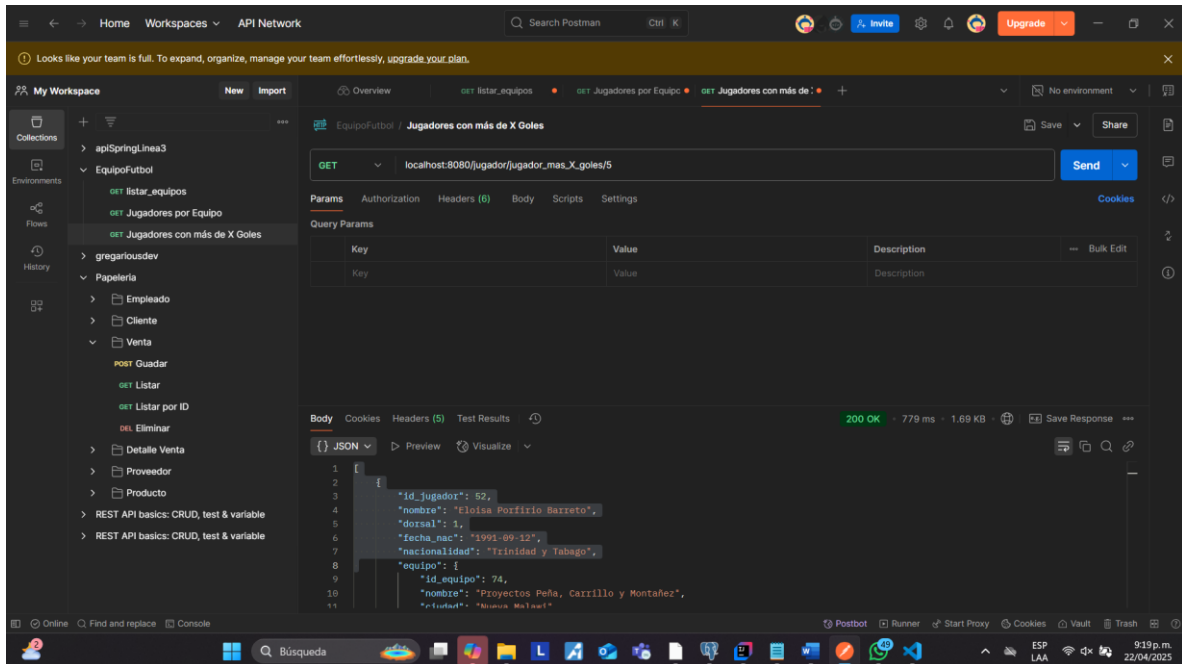
15°C Mayorm, nublado

4. Consultas nativas

Obtener todos los jugadores de un equipo específico:

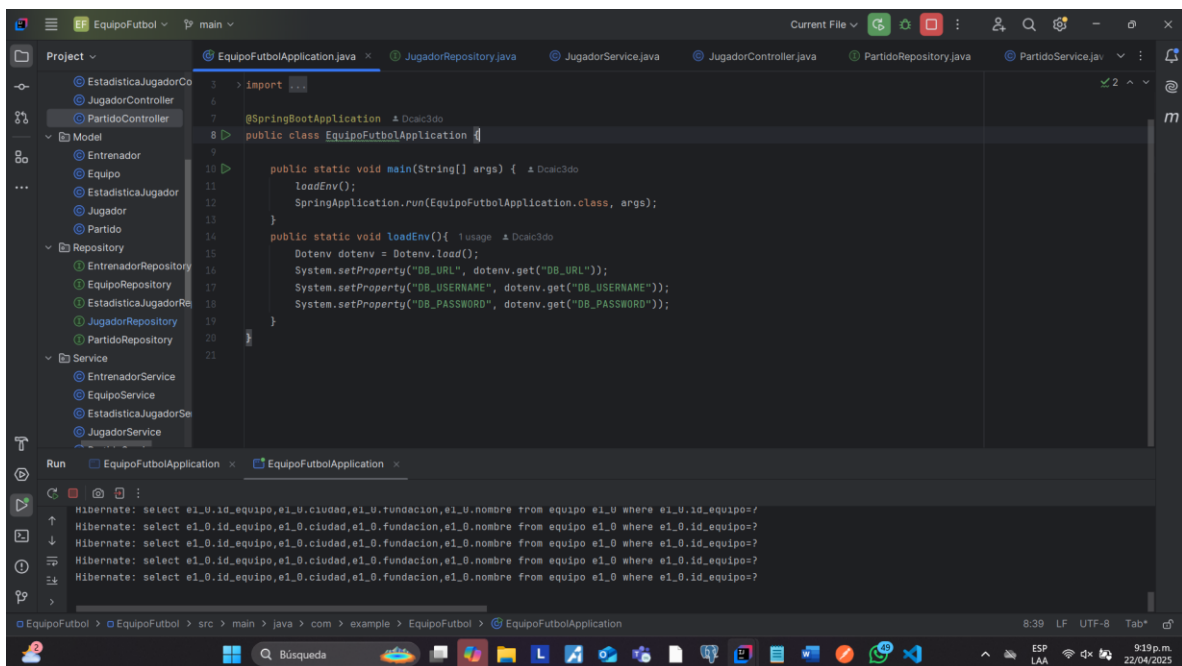


Obtener los jugadores que han marcado más de X goles



The screenshot shows the API Network interface. The selected endpoint is `GET localhost:8080/jugador/jugador_mas_X_goles/5`. The response status is **200 OK** with a response time of 779 ms and a body size of 1.69 KB. The response body is a JSON array containing one player object.

```
1 [
2   {
3     "id_jugador": 52,
4     "nombre": "Eloisa Porfirio Barreto",
5     "dorsal": 1,
6     "fecha_nac": "1991-09-12",
7     "nacionalidad": "Trinidad y Tabago",
8     "equipo": {
9       "id_equipo": 74,
10      "nombre": "Proyectos Peña, Carrillo y Montañez",
11      "pais": "Nicaragua"
12    }
13   }
14 ]
```



The screenshot shows the source code of the `EquipoFutbolApplication.java` file in the IntelliJ IDEA IDE. The code is a Spring Boot application that loads environment variables and sets database properties.

```
1 import org.springframework.boot.SpringApplication;
2 import org.springframework.boot.autoconfigure.SpringBootApplication;
3
4 @SpringBootApplication
5 public class EquipoFutbolApplication {
6
7     public static void main(String[] args) {
8         SpringApplication.run(EquipoFutbolApplication.class, args);
9     }
10
11     public static void loadEnv() {
12         Dotenv dotenv = Dotenv.load();
13         System.setProperty("DB_URL", dotenv.get("DB_URL"));
14         System.setProperty("DB_USERNAME", dotenv.get("DB_USERNAME"));
15         System.setProperty("DB_PASSWORD", dotenv.get("DB_PASSWORD"));
16     }
17 }
```

Obtener el número total de goles marcados por un equipo en todos sus partidos

