

PARCIAL 2 LINEA DE PROFUNDIZACION 3



Joan Felipe Nuñez Angarita

521000121

William Alexander Matayano Porras

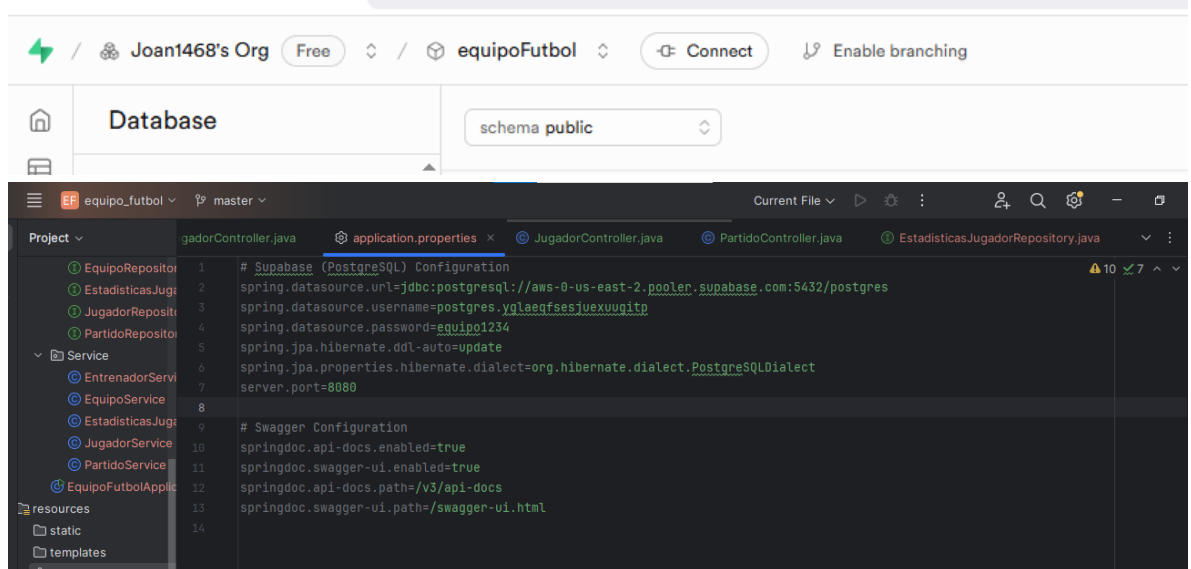
Universidad de Cundinamarca

Extension chia- Facultad de Ingenieria

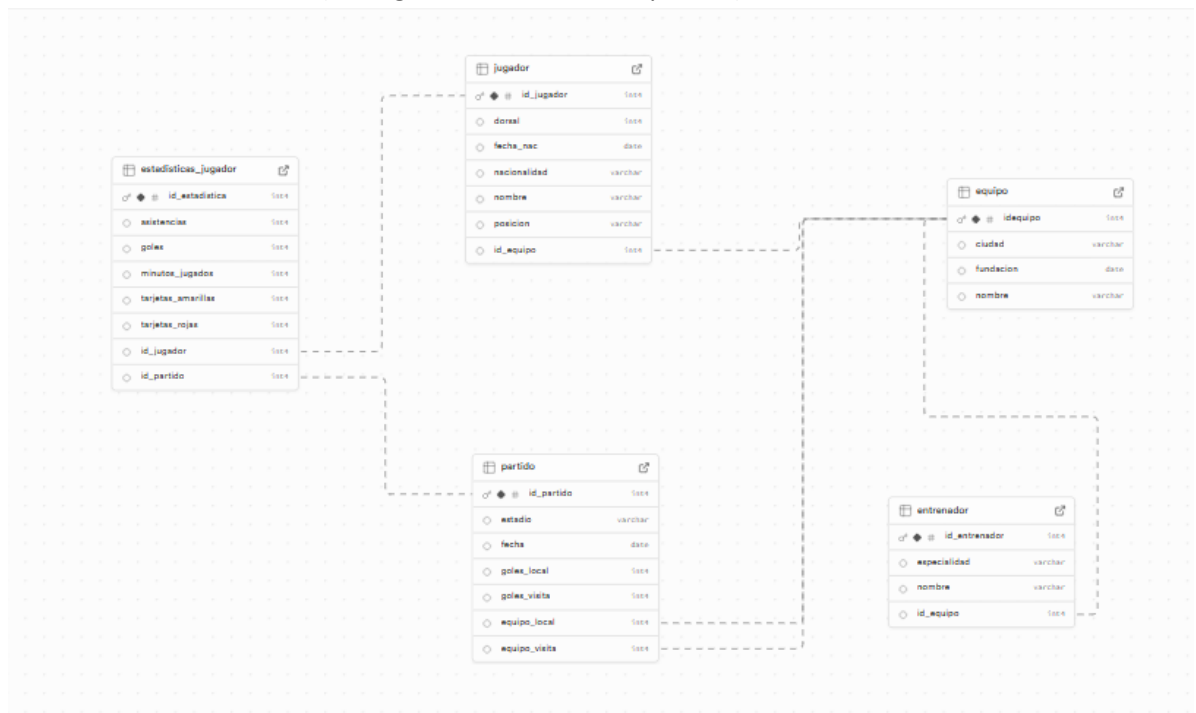
2025

Desarrollar una API REST en Spring Boot que permita gestionar la información de un equipo de fútbol: jugadores, entrenadores, partidos y estadísticas. Se documentará con Swagger y usará Supabase como base de datos PostgreSQL. **Acciones y Etapas del Proyecto**

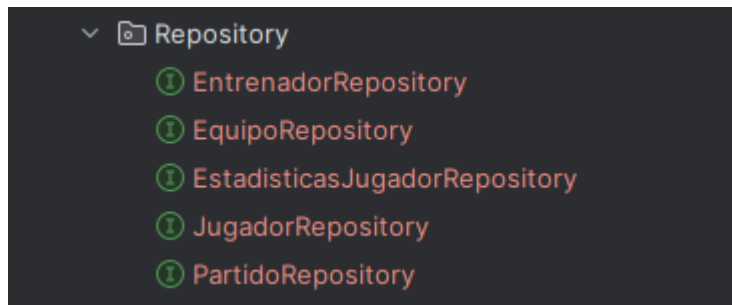
1. Conexión con Supabase



2. Crear el Modelo de Datos (Entregar screen modelo Supabase)



3. Crear los Repositorios



Entrenador Repository

```
1 package com.example.equipo_futbol.Repository;
2
3 import com.example.equipo_futbol.Model.Entrenador;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository 2 usages
8 public interface EntrenadorRepository extends JpaRepository<Entrenador, Integer> {
9     // Consultas básicas heredadas de JpaRepository
10 }
```

EquipoRepository

```
import com.example.equipo_futbol.Model.Equipo;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

@Repository 2 usages
public interface EquipoRepository extends JpaRepository<Equipo, Integer> {

    // Consulta nativa: Total de goles por equipo
    @Query(value = "" no usages
        SELECT SUM(ej.goles) FROM estadisticas_jugador ej
        JOIN jugadores j ON ej.id_jugador = j.id_jugador
        WHERE j.id_equipo = :equipoId
        "", nativeQuery = true)
    Integer sumGolesByEquipoId(@Param("equipoId") Integer equipoId);
}
```

EstadisticasJugadorRepository

```
1 package com.example.equipo_futbol.Repository;
2
3 import com.example.equipo_futbol.Model.EstadisticasJugador;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository 2 usages
8 public interface EstadisticasJugadorRepository extends JpaRepository<EstadisticasJugador, Integer> {
9     // Consultas básicas heredadas de JpaRepository
10 }
```

JugadorRepository

```
5 import org.springframework.data.jpa.repository.Query;
6 import org.springframework.data.repository.query.Param;
7 import org.springframework.stereotype.Repository;
8 import java.util.List;
9
10 @Repository 2 usages
11 public interface JugadorRepository extends JpaRepository<Jugador, Integer> {
12
13     // Consulta nativa: Obtener todos los jugadores de un equipo específico
14     @Query(value = "SELECT * FROM jugadores WHERE id_equipo = :equipoId", nativeQuery = true) no usages
15     List<Jugador> findJugadoresByEquipoId(@Param("equipoId") Integer equipoId);
16
17     // Consulta nativa: Jugadores con más de X goles
18     @Query(value = "" no usages
19         SELECT j.* FROM jugadores j
20         JOIN estadisticas_jugador ej ON j.id_jugador = ej.id_jugador
21         GROUP BY j.id_jugador
22         HAVING SUM(ej.goles) > :minGoles
23         "", nativeQuery = true)
24     List<Jugador> findJugadoresConMasDeXGoles(@Param("minGoles") Integer minGoles);
25 }
```

PartidoRepository

```

4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.data.jpa.repository.Query;
6 import org.springframework.stereotype.Repository;
7 import java.util.List;
8
9 @Repository 2 usages
10 public interface PartidoRepository extends JpaRepository<Partido, Integer> {
11
12     // Consulta nativa: Resultados con nombres de equipos
13     @Query(value = """ no usages
14         SELECT p.id_partido, p.fecha, p.estadio,
15                p.goles_local, p.goles_visita,
16                el.nombre as nombre_local,
17                ev.nombre as nombre_visita
18         FROM partidos p
19         JOIN equipos el ON p.equipo_local = el.id_equipo
20         JOIN equipos ev ON p.equipo_visita = ev.id_equipo
21         """, nativeQuery = true)
22     List<Object[]> findAllPartidosConNombresEquipos();
23 }

```

4. Crear los Servicios

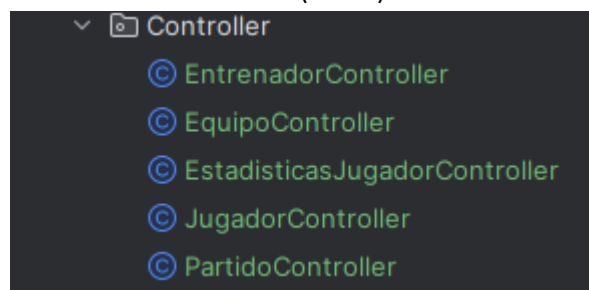
- © EntrenadorService
- © EquipoService
- © EstadisticasJugadorService
- © JugadorService
- © PartidoService

```

1 package com.example.equipo_futbol.Service;
2
3 import com.example.equipo_futbol.Model.Entrenador;
4 import com.example.equipo_futbol.Repository.EntrenadorRepository;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7
8 import java.util.List;
9 import java.util.Optional;
10
11 @Service
12 public class EntrenadorService {
13
14     @Autowired
15     private EntrenadorRepository entrenadorRepository;
16
17     //buscar todos
18     public List<Entrenador> findAll() {
19         return entrenadorRepository.findAll();
20     }
21
22     //buscar por id
23     public Optional<Entrenador> findById(Integer id) {
24         return entrenadorRepository.findById(id);
25     }
26

```

5. Crear los Controladores (CRUD)



6. Documentar la API con Swagger

7. Consultas nativas

- Obtener todos los jugadores de un equipo específico
- Obtener los jugadores que han marcado más de X goles
- Obtener el número total de goles marcados por un equipo en todos sus partidos
- Obtener los resultados de todos los partidos indicando los nombres de los equipos

Requisitos para la Entrega

1. Implementar cada consulta nativa en un método de repositorio

- Usar @Query con nativeQuery = true en el Repository.

2. Exponer las consultas a través del controlador (@RestController)

- Pueden usar métodos GET o POST según el tipo de consulta.

3. Probar cada consulta en Postman

- Usar la URL correcta (http://localhost:8080/...).
- Enviar parámetros (si aplica) por path o query.
- Mostrar la respuesta que entrega el backend.

4. Entregar capturas de pantalla

- Mostrar la solicitud con su método, parámetros y resultado.
- Si es posible, exportar la colección en formato .json

Modelo Relacional

1. Equipo

| Campo | Tipo de Dato | Clave | Descripción |
|-----------|--------------|-------|--------------------------|
| id_equipo | INT | PK | Identificador del equipo |
| nombre | VARCHAR(100) | | Nombre del equipo |
| ciudad | VARCHAR(100) | | Ciudad base del equipo |
| fundacion | DATE | | Fecha de fundación |

2. Jugador

| Campo | Tipo de Dato | Clave | Descripción |
|--------------|--------------|-------|-----------------------------------|
| id_jugador | INT | PK | Identificador del jugador |
| nombre | VARCHAR(100) | | Nombre completo del jugador |
| posicion | VARCHAR(50) | | Posición (portero, defensa, etc.) |
| dorsal | INT | | Número de camiseta |
| fecha_nac | DATE | | Fecha de nacimiento |
| nacionalidad | VARCHAR(100) | | Nacionalidad |
| id_equipo | INT | FK | Equipo al que pertenece |

3. Entrenador

| Campo | Tipo de Dato | Clave | Descripción |
|---------------|--------------|-------|----------------------------------|
| id_entrenador | INT | PK | Identificador del entrenador |
| nombre | VARCHAR(100) | | Nombre del entrenador |
| especialidad | VARCHAR(100) | | Rol (Principal, asistente, etc.) |
| id_equipo | INT | FK | Equipo al que pertenece |

4. Partido

| Campo | Tipo de Dato | Clave | Descripción |
|---------------|--------------|-------|----------------------------|
| id_partido | INT | PK | Identificador del partido |
| fecha | DATE | | Fecha del partido |
| estadio | VARCHAR(100) | | Nombre del estadio |
| equipo_local | INT | FK | ID del equipo local |
| equipo_visita | INT | FK | ID del equipo visitante |
| goles_local | INT | | Goles del equipo local |
| goles_visita | INT | | Goles del equipo visitante |

5. Estadísticas Jugador

| Campo | Tipo de Dato | Clave | Descripción |
|-----------------|--------------|-------|---------------------------------|
| id_estadistica | INT | PK | Identificador de la estadística |
| id_jugador | INT | FK | Jugador relacionado |
| id_partido | INT | FK | Partido relacionado |
| minutos_jugados | INT | | Minutos jugados |
| goles | INT | | Goles anotados |
| asistencias | INT | | Asistencias |

| | | | |
|--------------------|-----|--|------------------------------|
| tarjetas_amarillas | INT | | Número de tarjetas amarillas |
| tarjetas_rojas | INT | | Número de tarjetas rojas |

Relaciones

Equipo — Jugador

- **Relación:** Un equipo **tiene muchos jugadores**
- **Cardinalidad:** 1 a N

Equipo — Entrenador

- **Relación:** Un equipo **tiene uno o varios entrenadores**
- **Cardinalidad:** 1 a N

Equipo — Partido (como local y visitante)

- **Relación:** Un equipo puede jugar **muchos partidos como local y muchos como visitante**
- **Cardinalidad:** 1 a N (dos veces)

Jugador — EstadísticasJugador

- **Relación:** Un jugador puede tener **muchas estadísticas en diferentes partidos**
- **Cardinalidad:** 1 a N

Partido — EstadísticasJugador

- **Relación:** Un partido puede tener **estadísticas para muchos jugadores**
- **Cardinalidad:** 1 a N