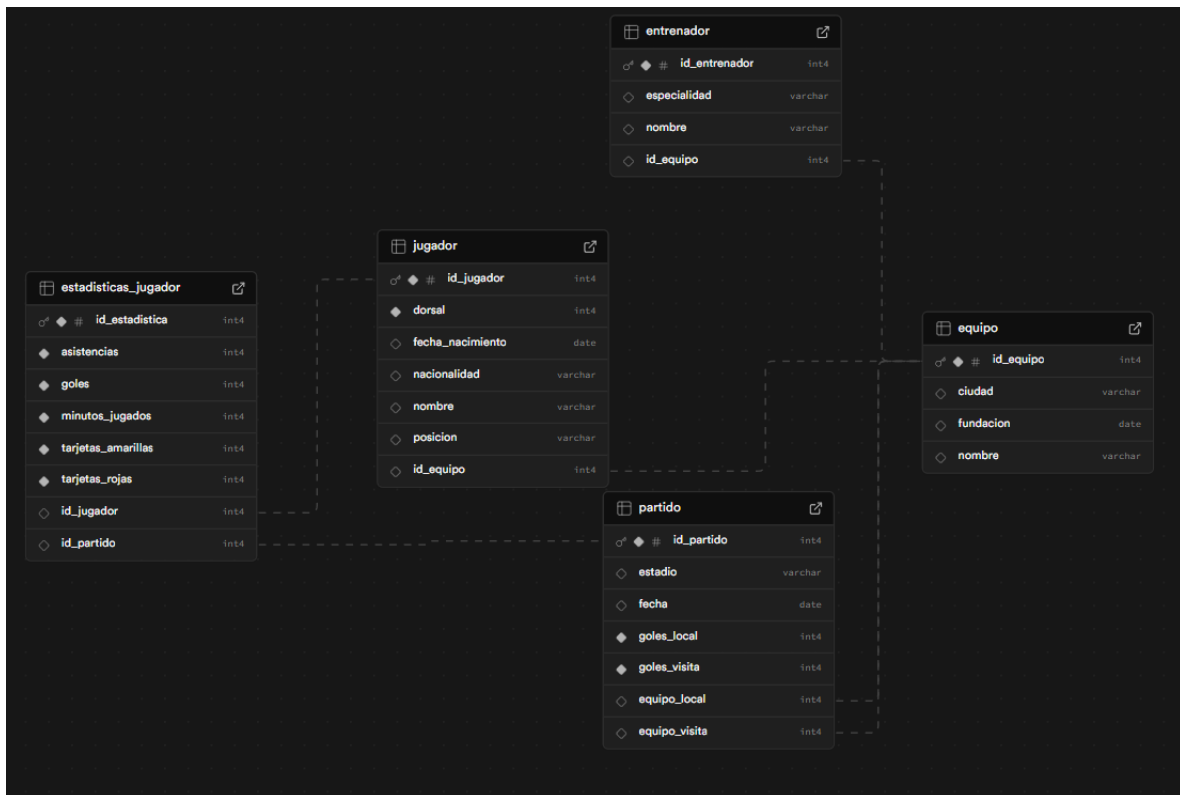


Joan Sebastian Novoa Hernández

1. Conexión con Supabase



2. Implementar cada consulta nativa en un método de repositorio

```
@Repository 3 usages ± S3B4S *
public interface JugadorRepository extends JpaRepository<Jugador, Integer> {

    @Query(value = "SELECT * FROM jugador WHERE id_equipo = :idEquipo", nativeQuery = true) 1 usage new *
    List<Jugador> listarJugadoresPorEquipo(@Param("idEquipo") int idEquipo);
}
```

Esta es una consulta nativa en SQL (es decir, escrita directamente como se usaría en PostgreSQL, no en JPQL), que se ejecuta desde un repositorio JPA en Spring Boot.

Obtener todos los jugadores de una tabla jugador que pertenezcan a un equipo específico.

```

public List<Jugador> listarJugadoresPorEquipo(int id){ 1 usage
    try{
        if(!equipoRepository.existsById(id)){
            throw new IllegalArgumentException("No se encontro el equipo con ID: " + id);
        }
        return jugadorRepository.listarJugadoresPorEquipo(id);
    }catch(Exception e){
        System.out.println("Error al listar los jugadores del equipo " + e.getMessage());
        return new ArrayList<>();
    }
}

```

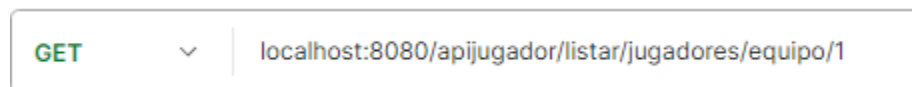
El servicio valida primero que el equipo exista y después invoca al método del repositorio pasando como parámetro el id, si el equipo no existe retorna una lista vacía.

```

@GetMapping(Ⓜ"/listar/jugadores/equipo/{id}")
public List<Jugador> listarJugadoresPorEquipo(@PathVariable int id){
    return jugadorService.listarJugadoresPorEquipo(id);
}

```

Este método del controlador expone una ruta HTTP GET (/listar/jugadores/equipo/{id}) que permite obtener, en formato JSON, la lista de jugadores asociados a un equipo específico según su ID. Usa @PathVariable para capturar el ID desde la URL y llama al servicio jugadorService.listarJugadoresPorEquipo(id), que valida la existencia del equipo y retorna los jugadores o una lista vacía si ocurre un error.



Con postman por medio de la url se pasa el parámetro del id 1, lo cual nos devuelve todos los jugadores asociados al equipo con id 1

```

1  [
2    {
3      "id_jugador": 1,
4      "nombre": "Olegario Casanovas Perello",
5      "posicion": "Centrocampista",
6      "dorsal": 7,
7      "fecha_nacimiento": "1991-08-10",
8      "nacionalidad": "Colombiano",
9      "equipo": {
10        "id_equipo": 1,
11        "nombre": "Marino LLC",
12        "ciudad": "Cantabria",
13        "fundacion": "1999-05-23"
14      }
15    },
16    {
17      "id_jugador": 2,
18      "nombre": "Javier Llopis Rodriguez",
19      "posicion": "Portero",
20      "dorsal": 11,
21      "fecha_nacimiento": "2001-11-14",
22      "nacionalidad": "Colombiano",
23      "equipo": {
24        "id_equipo": 1,
25        "nombre": "Marino LLC",
26        "ciudad": "Cantabria",
27        "fundacion": "1999-05-23"
28      }
29    },
30  ],
  
```

```

1  {
2    "id_jugador": 3,
3    "nombre": "Lope del Tellez",
4    "posicion": "Centrocampista",
5    "dorsal": 16,
6    "fecha_nacimiento": "1990-07-09",
7    "nacionalidad": "Colombiano",
8    "equipo": {
9      "id_equipo": 1,
10     "nombre": "Marino LLC",
11     "ciudad": "Cantabria",
12     "fundacion": "1999-05-23"
13   }
14 },
15 {
16   "id_jugador": 4,
17   "nombre": "Valerio Heraclio Morcillo Alba",
18   "posicion": "Defensa",
19   "dorsal": 18,
20   "fecha_nacimiento": "2004-03-21",
21   "nacionalidad": "Colombiano",
22   "equipo": {
23     "id_equipo": 1,
24     "nombre": "Marino LLC",
25     "ciudad": "Cantabria",
26     "fundacion": "1999-05-23"
27   }
28 }
29 ]
  
```

En este caso, 4 jugadores pertenecientes al equipo 1. La prueba igualmente se puede realizar con swagger.

id * required

integer(\$int32)

(path)

1

Execute

Responses

Curl

curl -X 'GET' \
'http://localhost:8080/apijugador/listar/jugadores/equipo/1' \
-H 'accept: */*'

Request URL

http://localhost:8080/apijugador/listar/jugadores/equipo/1

Server response

Code

Details

200

Response body

[
{
"id_jugador": 1,
"nombre": "Olegario Casanovas Perello",
"posicion": "Centrocampista",
"dorsal": 7,
"fecha_nacimiento": "1991-08-10",
"nacionalidad": "Colombiano",
"equipo": {
"id_equipo": 1,
"nombre": "Marino LLC",
"ciudad": "Cantabria",

```
@Query(value = "SELECT j.* FROM jugador j " + 1 usage new *
        "JOIN estadisticas_jugador ej ON j.id_jugador = ej.id_jugador " +
        "GROUP BY j.id_jugador " +
        "HAVING SUM(ej.goles) > :minGoles", nativeQuery = true)
List<Jugador> listarJugadoresConMasDeXGoles(@Param("minGoles") int minGoles);
```

Esta consulta nativa retorna una lista de jugadores que han anotado más de una cantidad específica de goles. Realiza un JOIN entre las tablas jugador y estadisticas_jugador usando el ID del jugador, agrupa los resultados por jugador y utiliza HAVING para filtrar solo aquellos cuya suma de goles sea mayor al valor recibido como parámetro (:minGoles). Es útil para listar jugadores destacados por su rendimiento goleador.

```
public List<Jugador> listarJugadoresConMasDeXGoles(int minGoles) { 1 usage
    return jugadorRepository.listarJugadoresConMasDeXGoles(minGoles);
}
```

Este método del servicio simplemente llama al repositorio jugadorRepository para ejecutar la consulta que retorna los jugadores cuya suma de goles supera un valor mínimo (minGoles). Actúa como intermediario entre el controlador y el repositorio, delegando la lógica de búsqueda sin realizar validaciones adicionales

```
@GetMapping("Ⓜ"/listar/jugadores/goles/{minGoles}")
public List<Jugador> listarJugadoresConMasDeXGoles(@PathVariable int minGoles) {
    return jugadorService.listarJugadoresConMasDeXGoles(minGoles);
}
```

Este método del controlador expone una ruta HTTP GET (/listar/jugadores/goles/{minGoles}) que permite obtener la lista de jugadores que han marcado más de una cierta cantidad de goles. Usa @PathVariable para recibir el valor mínimo de goles desde la URL y delega la lógica al servicio jugadorService.listarJugadoresConMasDeXGoles(minGoles), devolviendo el resultado en formato JSON.

GET

/apijugador/listar/jugadores/goles/{minGoles}

Parameters

Name	Description
<div>minGoles * required</div> <div>integer(\$int32)</div> <div>(path)</div>	<div>5</div>

Request URL

http://localhost:8080/apijugador/listar/jugadores/goles/5

Server response

Code	Details
200	<div>Response body</div> <div> <pre>[{ "id_jugador": 15, "nombre": "Alcides Grau Pullol", "posicion": "Centrocampista", "dorsal": 22, "fecha_nacimiento": "1992-01-23", "nacionalidad": "Colombiano", "equipo": { "id_equipo": 4, "nombre": "Blanch-Correa", "ciudad": "Pontevedra", "fundacion": "2009-08-24" } }]</pre> </div>

```

Hibernate: select j1_0.id_jugador,j1_0.dorsal,j1_0.id_equipo,j1_0.fecha_nacimiento,j1_0.nacionalidad,j1_0.nombre
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select count(*) from equipo e1_0 where e1_0.id_equipo=?
Hibernate: SELECT * FROM jugador WHERE id_equipo = ?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select count(*) from equipo e1_0 where e1_0.id_equipo=?
Hibernate: SELECT * FROM jugador WHERE id_equipo = ?

```

Tercera consulta

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

```
@Repository 3 usages S3B4S *
public interface PartidoRepository extends JpaRepository<Partido, Integer> {

    @Query(value = "SELECT SUM(CASE WHEN p.equipo_local = :id THEN p.goles_local " +
        "WHEN p.equipo_visita = :id THEN p.goles_visita ELSE 0 END) " +
        "FROM partido p", nativeQuery = true)
    Integer totalGolesPorEquipo(@Param("id") int idEquipo);
}
```

Esta consulta nativa calcula el **total de goles anotados por un equipo específico**, ya sea como local o visitante. Usa una expresión CASE WHEN para sumar los goles del equipo dependiendo de si jugó como **local** (goles_local) o **visitante** (goles_visita). Si no fue ninguno, suma 0. Luego, aplica SUM(...) sobre esos valores para obtener el total y lo devuelve como un Integer. Ideal para mostrar estadísticas generales del equipo en la temporada.

```
public int obtenerTotalGolesEquipo(int idEquipo) { 1 usage
    return partidoRepository.totalGolesPorEquipo(idEquipo);
}
```

```
@GetMapping("/goles/equipo/{id}")
public int obtenerTotalGolesEquipo(@PathVariable int id) {
    return partidoService.obtenerTotalGolesEquipo(id);
}
```

GET

/apipartido/goles/equipo/{id}

Parameters

Name	Description
id * required	
integer(\$int32)	1
(path)	

Execute

Curl

```
curl -X 'GET' \
'http://localhost:8080/apipartido/goles/equipo/1' \
-H 'accept: */*'
```

Request URL

```
http://localhost:8080/apipartido/goles/equipo/1
```

Server response

Code	Details
200	<div>Response body</div> <div>13</div>

Para el equipo con id 2

Curl

```
curl -X 'GET' \
'http://localhost:8080/apipartido/goles/equipo/2' \
-H 'accept: */*'
```

Request URL

```
http://localhost:8080/apipartido/goles/equipo/2
```

Server response

Code	Details
200	<div>Response body</div> <div>4</div>

Ultima consulta nativa

Obtener los resultados de todos los partidos indicando los nombres de los equipos

```
@Query(value = "SELECT p.id_partido, p.fecha, p.estadio, el.nombre AS nombre_local, ev.nombre AS nombre_visita, " +  
    "p.goles_local, p.goles_visita " +  
    "FROM partido p " +  
    "JOIN equipo el ON p.equipo_local = el.id_equipo " +  
    "JOIN equipo ev ON p.equipo_visita = ev.id_equipo", nativeQuery = true)  
List<Object[]> obtenerResultadosConNombresEquipos();
```

Esta consulta nativa recupera una lista de partidos incluyendo detalles del encuentro y los **nombres de los equipos** que participaron. Realiza un JOIN con la tabla equipo dos veces: una para obtener el nombre del equipo local (el) y otra para el equipo visitante (ev). Devuelve columnas como el ID del partido, fecha, estadio, goles y nombres de ambos equipos. El resultado es una lista de arreglos de objetos (List<Object[]>), donde cada posición representa una columna seleccionada. Es útil para mostrar resultados detallados de partidos con nombres de los equipos en lugar de solo sus IDs.

```
public List<Object[]> obtenerResultadosPartidos() { 1 usage  
    return partidoRepository.obtenerResultadosConNombresEquipos();  
}
```

```
@GetMapping("/resultados")  
public List<Object[]> obtenerResultadosPartidos() {  
    return partidoService.obtenerResultadosPartidos();  
}
```

Esta consulta no recibe parámetros y devuelve los resultados de los partidos con el nombre de los equipos

Curl

```
curl -X 'GET' \
'http://localhost:8080/apipartido/resultados' \
-H 'accept: */*'
```

Request URL

```
http://localhost:8080/apipartido/resultados
```

Server response

Code	Details
200	<p>Response body</p> <pre>[[1, "2025-03-20", "Calle de Claudio Zamora", "Blanch-Correa", "Batalla, Martinez and Vives", 5, 1], [2, "2024-10-13", "Canada de Goyo Marquez", "Carbo-Losa", "Blanch-Correa", 4, 5], [3, "2025-02-19", "Canada Vidal Lumbreras", "Carbo-Losa", "Canellas Group", 3, 4],]</pre>

```
curl -X 'GET' \
'http://localhost:8080/apipartido/resultados' \
-H 'accept: */*'
```

Request URL

```
http://localhost:8080/apipartido/resultados
```

Server response

Code	Details
200	<p>Response body</p> <pre>[4, "2024-07-03", "Callejon de Aaron Jorda", "Canellas Group", "Marino LLC", 3, 0], [5, "2025-01-16", "Avenida Florentino Solano", "Canellas Group", "Blanch-Correa", 0, 2], [6, "2024-11-25", "Pasaje de Marisol Losa", "Blanch-Correa", "Santana-Machado", 0, 5], [7, </pre>

Respuesta del backend sin errores para las consultas

```

↑
↓
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: SELECT j.* FROM jugador j JOIN estadisticas_jugador ej ON j.id_jugador = ej.id_jugador GROUP BY j.id_jugador HAVING SUM(ej.goles) > ?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: SELECT j.* FROM jugador j JOIN estadisticas_jugador ej ON j.id_jugador = ej.id_jugador GROUP BY j.id_jugador HAVING SUM(ej.goles) > ?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: SELECT j.* FROM jugador j JOIN estadisticas_jugador ej ON j.id_jugador = ej.id_jugador GROUP BY j.id_jugador HAVING SUM(ej.goles) > ?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: select e1_0.id_equipo,e1_0.ciudad,e1_0.fundacion,e1_0.nombre from equipo e1_0 where e1_0.id_equipo=?
Hibernate: SELECT SUM(CASE WHEN p.equipo_local = ? THEN p.goles_local WHEN p.equipo_visita = ? THEN p.goles_visita ELSE 0 END) FROM partido p
Hibernate: SELECT SUM(CASE WHEN p.equipo_local = ? THEN p.goles_local WHEN p.equipo_visita = ? THEN p.goles_visita ELSE 0 END) FROM partido p
Hibernate: SELECT SUM(CASE WHEN p.equipo_local = ? THEN p.goles_local WHEN p.equipo_visita = ? THEN p.goles_visita ELSE 0 END) FROM partido p
Hibernate: SELECT p.id_partido, p.fecha, p.estadio, el.nombre AS nombre_local, ev.nombre AS nombre_visita, p.goles_local, p.goles_visita FROM partido p

```

Documentacion con swagger

partido-controller

PUT /apipartido/actualizar

POST /apipartido/guardar

GET /apipartido/resultados

GET /apipartido/listar

GET /apipartido/goles/equipo/{id}

GET /apipartido/buscar/{id}

DELETE /apipartido/eliminar/{id}

jugador-controller

PUT /apijugador/actualizar

POST /apijugador/guardar

GET /apijugador/listar

GET /apijugador/listar/jugadores/goles/{minGoles}

GET /apijugador/listar/jugadores/equipo/{id}

GET /apijugador/buscar/{id}

DELETE /apijugador/eliminar/{id}

estadisticas-jugador-controller

PUT /apiestadisticasjugador/actualizar

POST /apiestadisticasjugador/guardar

GET /apiestadisticasjugador/listar

GET /apiestadisticasjugador/buscar/{id}

DELETE /apiestadisticasjugador/eliminar/{id}

equipo-controller

PUT /apiequipo/actualizar

POST /apiequipo/guardar

GET /apiequipo/listar

GET /apiequipo/buscar/{id}

DELETE /apiequipo/eliminar/{id}

entrenador-controller

PUT /apientrenador/actualizar

POST /apientrenador/guardar

GET /apientrenador/listar

GET /apientrenador/buscar/{id}

DELETE /apientrenador/eliminar/{id}

Conexión con pgadmin

