

## 1. Conexión con Supabase

The diagram illustrates a Supabase database schema with five tables: **entrenador**, **equipo**, **partido**, **estadisticas\_jugador**, and **jugador**. The relationships are as follows:

- entrenador** (id\_entrenador: int4, especialidad: varchar, nombre: varchar, id\_equipo: int4) is linked to **equipo** via **id\_equipo**.
- equipo** (id\_equipo: int4, ciudad: varchar, fundacion: timestamp, nombre: varchar) is linked to **partido** via **equipo\_local** and **equipo\_visitante**.
- partido** (id\_partido: int4, estadio: varchar, fecha: timestamp, goles\_local: int4, goles\_visitante: int4, equipo\_local: int4, equipo\_visitante: int4) is linked to **estadisticas\_jugador** via **id\_partido**.
- estadisticas\_jugador** (id\_estadistica: int4, asistencias: int4, goles: int4, minutos\_jugados: int4, tarjetas\_amarillas: int4, tarjetas\_rojas: int4, id\_jugador: int4, id\_partido: int4) is linked to **jugador** via **id\_jugador**.
- jugador** (id\_jugador: int4, dorsal: int4, fecha\_nac: timestamp, nacionalidad: varchar, nombre: varchar, posicion: varchar, id\_equipo: int4) is linked to **equipo** via **id\_equipo**.

Below the diagram, two code snippets are shown:

**Snippet 1: .env file configuration**

```
DB_URL=jdbc:postgresql://aws-0-us-east-1.pooler.supabase.com:5432/postgres
DB_USERNAME=postgres.uojwhdtaderydsdwpbrm
DB_PASSWORD=XKmmnmMuCGlkIgQS
```

**Snippet 2: Spring Boot configuration files**

```
spring.application.name=Parcial2
spring.datasource.url=${DB_URL}
spring.datasource.username=${DB_USERNAME}
spring.datasource.password=${DB_PASSWORD}
spring.datasource.driver-class-name=org.postgresql.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

## 2. Creación de Modelos

```
Entrenador.java x
1 package com.example.Parcial2.Model;
2
3 import jakarta.persistence.*;
4 import lombok.*;
5
6 @Entity 15 usages  LuisaRom
7 @Data
8 @NoArgsConstructor
9 @AllArgsConstructor
10 public class Entrenador {
11
12     @Id no usages
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     private Integer id_entrenador;
15
16     private String nombre; no usages
17     private String especialidad; no usages
18
19     @ManyToOne no usages
20     @JoinColumn(name = "id_equipo")
21     private Equipo equipo;
22 }
```

**Clase Entrenador** tiene relación con la **clase Equipo** ya que un equipo puede tener varios entrenadores

Se usa la dependencia Lombok para **reducir líneas de código** y se usan las etiquetas: @Entity - @Data - @NoArgsConstructor y @AllArgsConstructor.

```
Equipo.java x
1 package com.example.Parcial2.Model;
2
3 > import ...
7
8 @Entity 18 usages LuisaRom
9 @Data
10 @NoArgsConstructor
11 @AllArgsConstructor
12 public class Equipo {
13
14     @Id no usages
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Integer id_equipo;
17
18     private String nombre; no usages
19     private String ciudad; no usages
20     private Date fundacion; no usages
21
22     @OneToMany(mappedBy = "equipo") no usages
23     private List<Jugador> jugadores;
24
25     @OneToMany(mappedBy = "equipo") no usages
26     private List<Entrenador> entrenadores;
27
28     @OneToMany(mappedBy = "equipoLocal") no usages
29     private List<Partido> partidosLocal;
30
31     @OneToMany(mappedBy = "equipoVisita") no usages
32     private List<Partido> partidosVisita;
33 }
```

**Clase Equipo** tiene relación con la **clase Entrenador** ya que un equipo tiene uno o varios entrenadores, relación con la **clase Jugador** ya que un equipo tiene muchos jugadores, relación con la **clase Partido** ya que un equipo puede jugar muchos partidos como visitante o local.

Se usa la dependencia Lombok para **reducir líneas de código** y se usan las etiquetas: @Entity - @Data - @NoArgsConstructor y @AllArgsConstructor.

```

1 package com.example.Parcial2.Model;
2
3
4 import jakarta.persistence.*;
5 import lombok.*;
6
7 @Entity 16 usages  🧑 LuisaRom
8 @Data
9 @NoArgsConstructor
10 @AllArgsConstructor
11 public class EstadisticasJugador {
12
13     @Id no usages
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Integer id_estadistica;
16
17     private Integer minutos_jugados; no usages
18     private Integer goles; no usages
19     private Integer asistencias; no usages
20     private Integer tarjetas_amarillas; no usages
21     private Integer tarjetas_rojas; no usages
22
23     @ManyToOne no usages
24     @JoinColumn(name = "id_jugador")
25     private Jugador jugador;
26
27     @ManyToOne no usages
28     @JoinColumn(name = "id_partido")
29     private Partido partido;
30 }
31

```

**Clase EstadisticasJugador** tiene relación con la **clase Jugador** ya que el jugador puede tener muchas estadísticas en los partidos y relación con **clase Partido** ya que un partido tiene muchas estadísticas para los jugadores.

Se usa la dependencia Lombok para **reducir líneas de código** y se usan las etiquetas: @Entity - @Data - @NoArgsConstructor y @AllArgsConstructor.

```
© Jugador.java ×
1 package com.example.Parcial2.Model;
2
3 > import ...
4
5
6
7
8
9 @Entity 24 usages  👤 LuisaRom
10 @Data
11 @NoArgsConstructor
12 @AllArgsConstructor
13 public class Jugador {
14
15     @Id no usages
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private Integer id_jugador;
18
19     private String nombre; no usages
20     private String posicion; no usages
21     private Integer dorsal; no usages
22     private Date fecha_nac; no usages
23     private String nacionalidad; no usages
24
25     @ManyToOne no usages
26     @JoinColumn(name = "id_equipo")
27     private Equipo equipo;
28
29     @OneToMany(mappedBy = "jugador") no usages
30     private List<EstadisticasJugador> estadisticas;
31 }
32
```

**Clase Jugador** tiene relación con la **clase Equipo** ya que un equipo tiene muchos jugadores, relación con la **clase EstadisticasJugador** ya que un jugador tiene muchas estadísticas.

Se usa la dependencia Lombok para **reducir líneas de código** y se usan las etiquetas: @Entity - @Data - @NoArgsConstructor y @AllArgsConstructor.

```
Partido.java x
1 package com.example.Parcial2.Model;
2
3
4 > import ...
9
10 @Entity 17 usages LuisaRom
11 @Data
12 @NoArgsConstructor
13 @AllArgsConstructor
14 public class Partido {
15
16     @Id no usages
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private Integer id_partido;
19
20     private Date fecha; no usages
21     private String estadio; no usages
22     private Integer goles_local; no usages
23     private Integer goles_visita; no usages
24
25     @ManyToOne no usages
26     @JoinColumn(name = "equipo_local")
27     private Equipo equipoLocal;
28
29     @ManyToOne no usages
30     @JoinColumn(name = "equipo_visita")
31     private Equipo equipoVisita;
32
33     @OneToMany(mappedBy = "partido") no usages
34     private List<EstadisticasJugador> estadisticas;
35 }
36
```

**Clase Partido** tiene relación con la **clase Equipo** ya que un equipo puede jugar muchos partidos como local o visitante y relación con la **clase EstadisticasJugador** ya que un partido tiene muchas estadísticas en sus jugadores.

### 3. Creación de Repositorios

```
EntrenadorRepository.java x
1 package com.example.Parcial2.Repository;
2
3 import com.example.Parcial2.Model.Entrenador;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface EntrenadorRepository extends JpaRepository<Entrenador, Integer> { 6 usages
7 }
8

EquipoRepository.java x
1 package com.example.Parcial2.Repository;
2
3 import com.example.Parcial2.Model.Equipo;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface EquipoRepository extends JpaRepository<Equipo, Integer> { 6 usages
7 }
8

EstadisticasJugadorRepository.java x
1 package com.example.Parcial2.Repository;
2
3 import com.example.Parcial2.Model.EstadisticasJugador;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface EstadisticasJugadorRepository extends JpaRepository<EstadisticasJugador, Integer> { 6 usages
7 }
8

JugadorRepository.java x
1 package com.example.Parcial2.Repository;
2
3 import com.example.Parcial2.Model.Jugador;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.data.jpa.repository.Query;
6
7 import java.util.List;
8
9 public interface JugadorRepository extends JpaRepository<Jugador, Integer> { 6 usages
10
11
12 @Query(value = "SELECT * FROM jugador WHERE id_equipo = ?1", nativeQuery = true) 2 usages
13 List<Jugador> findJugadoresByEquipoId(int idEquipo);
14
15
16 @Query(value = "" 2 usages
17 SELECT j.* FROM jugador j_
18 JOIN estadistica_jugador ej ON j.id_jugador = ej.id_jugador
19 GROUP BY j.id_jugador
20 HAVING SUM(ej.goles) > ?1
21 "", nativeQuery = true)
22 List<Jugador> findJugadoresConMasDeXGoles(int goles);
23 }
24
25
```

Se usa las **@Query** ya que esta clase se una con otras para llamar informacion, se pueden usar agregaciones o filtrar datos.

```
PartidoRepository.java x
1 package com.example.Parcial2.Repository;
2
3 > import ...
4
5
6
7
8
9 public interface PartidoRepository extends JpaRepository<Partido, Integer> { 6 usages 1 LuisaRom
10
11
12     @Query(value = "" 2 usages 1 LuisaRom
13         SELECT p.id_partido, e.nombre AS equipo_local, ev.nombre AS equipo_visita,
14             p.goles_local, p.goles_visita, p.fecha
15         FROM partido p
16         JOIN equipo el ON p.equipo_local = el.id_equipo
17         JOIN equipo ev ON p.equipo_visita = ev.id_equipo
18         "", nativeQuery = true)
19     List<Object[]> findResultadosPartidosConNombres();
20
21
22     @Query(value = "" 2 usages 1 LuisaRom
23         SELECT e.nombre,
24             SUM(CASE WHEN p.equipo_local = e.id_equipo THEN p.goles_local ELSE 0 END +
25                 CASE WHEN p.equipo_visita = e.id_equipo THEN p.goles_visita ELSE 0 END) AS total_g
26         FROM equipo e
27         LEFT JOIN partido p ON e.id_equipo = p.equipo_local OR e.id_equipo = p.equipo_visita
28         GROUP BY e.nombre
29         "", nativeQuery = true)
30     List<Object[]> findTotalGolesPorEquipo();
31 }
32
```

Se usa las **@Query** ya que esta clase se una con otras para llamar informacion, se pueden usar agregaciones o filtrar datos.

#### 4. Creación de Servicios

##### Service Impl

```
ServiceImpl
├── EntrenadorServiceImpl
├── EquipoServiceImpl
├── EstadisticasJugadorServiceImpl
├── JugadorServiceImpl
└── PartidoServiceImpl
```

Implementación real de los métodos usando repositorios u otra lógica



```

EntrenadorServiceImpl.java x
2
3 import com.example.Parcial2.Model.Entrenador;
4 import com.example.Parcial2.Repository.EntrenadorRepository;
5 import com.example.Parcial2.Service.EntrenadorService;
6 import org.springframework.stereotype.Service;
7
8 import java.util.List;
9
10 @Service no usages LuisaRom
11 public class EntrenadorServiceImpl implements EntrenadorService {
12
13     private final EntrenadorRepository entrenadorRepository; 3 usages
14
15     public EntrenadorServiceImpl(EntrenadorRepository entrenadorRepository) { no usages LuisaRom
16         this.entrenadorRepository = entrenadorRepository;
17     }
18
19     @Override no usages LuisaRom
20     public List<Entrenador> getAll() {
21         return entrenadorRepository.findAll();
22     }
23
24     @Override no usages LuisaRom
25     public Entrenador crear(Entrenador entrenador) {
26         return entrenadorRepository.save(entrenador);
27     }
28 }
29

```

```

EquipoServiceImpl.java x
1 package com.example.Parcial2.Service.ServiceImpl;
2
3 > import ...
9
10 @Service no usages LuisaRom
11 public class EquipoServiceImpl implements EquipoService {
12
13     private final EquipoRepository equipoRepository; 3 usages
14
15     public EquipoServiceImpl(EquipoRepository equipoRepository) { no usages LuisaRom
16         this.equipoRepository = equipoRepository;
17     }
18
19     @Override no usages LuisaRom
20     public List<Equipo> getAllEquipos() {
21         return equipoRepository.findAll();
22     }
23
24     @Override no usages LuisaRom
25     public Equipo crearEquipo(Equipo equipo) {
26         return equipoRepository.save(equipo);
27     }
28 }
29

```

```

1 package com.example.Parcial2.Service.ServiceImpl;
2
3 > import ...
4
5
6
7
8
9
10 @Service no usages 1 LuisaRom
11 public class EstadisticasJugadorServiceImpl implements EstadisticasJugadorService {
12
13     private final EstadisticasJugadorRepository estadisticasRepository; 3 usages
14
15     public EstadisticasJugadorServiceImpl(EstadisticasJugadorRepository estadisticasRepository) { no
16         this.estadisticasRepository = estadisticasRepository;
17     }
18
19
20 @Override no usages 1 LuisaRom
21 public List<EstadisticasJugador> getAll() {
22     return estadisticasRepository.findAll();
23 }
24
25 @Override no usages 1 LuisaRom
26 public EstadisticasJugador crear(EstadisticasJugador estadisticas) {
27     return estadisticasRepository.save(estadisticas);
28 }
29 }

```

```

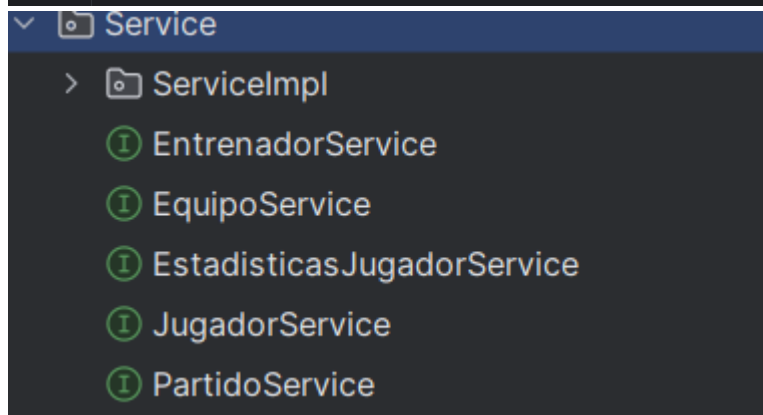
1 package com.example.Parcial2.Service.ServiceImpl;
2 > import ...
3
4
5
6
7
8
9 @Service no usages 1 LuisaRom
10 public class JugadorServiceImpl implements JugadorService {
11
12     private final JugadorRepository jugadorRepository; 5 usages
13
14     public JugadorServiceImpl(JugadorRepository jugadorRepository) { no usages 1 LuisaRom
15         this.jugadorRepository = jugadorRepository;
16     }
17
18
19 @Override no usages 1 LuisaRom
20 public List<Jugador> getAll() {
21     return jugadorRepository.findAll();
22 }
23
24 @Override no usages 1 LuisaRom
25 public Jugador crear(Jugador jugador) {
26     return jugadorRepository.save(jugador);
27 }
28
29 @Override no usages 1 LuisaRom
30 public List<Jugador> getPorEquipo(int idEquipo) {
31     return jugadorRepository.findJugadoresByEquipoId(idEquipo);
32 }
33
34 @Override no usages 1 LuisaRom
35 public List<Jugador> getConMasDe(int goles) {
36     return jugadorRepository.findJugadoresConMasDeXGoles(goles);
37 }
38 }

```

```

2
3 > import ...
9
10 @Service no usages LuisaRom
11 public class PartidoServiceImpl implements PartidoService {
12
13     private final PartidoRepository partidoRepository; 5 usages
14
15     public PartidoServiceImpl(PartidoRepository partidoRepository) { no usages LuisaRom
16         this.partidoRepository = partidoRepository;
17     }
18
19     @Override no usages LuisaRom
20     public List<Partido> getAll() {
21         return partidoRepository.findAll();
22     }
23
24     @Override no usages LuisaRom
25     public Partido crear(Partido partido) {
26         return partidoRepository.save(partido);
27     }
28
29     @Override no usages LuisaRom
30     public List<Object[]> getResultados() {
31         return partidoRepository.findResultadosPartidosConNombres();
32     }
33
34     @Override no usages LuisaRom
35     public List<Object[]> getGolesPorEquipo() {
36         return partidoRepository.findTotalGolesPorEquipo();
37     }
38 }

```



**Nombres de métodos (sin lógica)**

EntrenadorService.java
1
package com.example.Parcial2.Service;
2
3
> import ...
6
7
public interface EntrenadorService { 2 usages 1 implementation LuisaRom
8
9
List<Entrenador> getAll(); no usages 1 implementation LuisaRom
10
Entrenador crear(Entrenador entrenador); no usages 1 implementation LuisaRom
11
}
12

EquipoService.java
1
package com.example.Parcial2.Service;
2
3
> import ...
6
7
public interface EquipoService { 2 usages 1 implementation LuisaRom
8
9
List<Equipo> getAllEquipos(); no usages 1 implementation LuisaRom
10
Equipo crearEquipo(Equipo equipo); no usages 1 implementation LuisaRom
11
}
12

EstadisticasJugadorService.java
1
package com.example.Parcial2.Service;
2
3
> import ...
6
7
public interface EstadisticasJugadorService { 2 usages 1 implementation LuisaRom
8
9
List<EstadisticasJugador> getAll(); no usages 1 implementation LuisaRom
10
EstadisticasJugador crear(EstadisticasJugador estadisticas); no usages 1 imp
11
}
12
13

JugadorService.java
1
package com.example.Parcial2.Service;
2
3
> import ...
6
7
public interface JugadorService { 2 usages 1 implementation LuisaRom
8
9
List<Jugador> getAll(); no usages 1 implementation LuisaRom
10
Jugador crear(Jugador jugador); no usages 1 implementation LuisaRom
11
List<Jugador> getPorEquipo(int idEquipo); no usages 1 implementation LuisaRom
12
List<Jugador> getConMasDe(int goles); no usages 1 implementation LuisaRom
13
}

```

PartidoService.java x
1 package com.example.Parcial2.Service;
2
3 > import ...
4
5
6
7 public interface PartidoService { 2 usages 1 implementation LuisaRom
8
9 List<Partido> getAll(); no usages 1 implementation LuisaRom
10 Partido crear(Partido partido); no usages 1 implementation LuisaRom
11 List<Object[]> getResultados(); no usages 1 implementation LuisaRom
12 List<Object[]> getGolesPorEquipo(); no usages 1 implementation LuisaRom
13 }

```

## 5. Creación Controllers

```

EntrenadorController.java x
1 package com.example.Parcial2.Controller;
2
3 > import ...
4
5
6
7
8
9 @RestController no usages LuisaRom
10 @RequestMapping("/entrenadores")
11 public class EntrenadorController {
12
13     private final EntrenadorRepository entrenadorRepository; 3 usages
14
15     public EntrenadorController(EntrenadorRepository entrenadorRepository) { no usages
16         this.entrenadorRepository = entrenadorRepository;
17     }
18
19     @GetMapping no usages LuisaRom
20     public List<Entrenador> getAllEntrenadores() {
21         return entrenadorRepository.findAll();
22     }
23
24     @PostMapping no usages LuisaRom
25     public Entrenador crearEntrenador(@RequestBody Entrenador entrenador) {
26         return entrenadorRepository.save(entrenador);
27     }
28 }
29

```

```
EquipoController.java x
1 package com.example.Parcial2.Controller;
2
3 > import ...
4
5
6
7
8
9 @RestController no usages LuisaRom
10 @RequestMapping("/equipos")
11 public class EquipoController {
12
13     private final EquipoRepository equipoRepository; 3 usages
14
15     public EquipoController(EquipoRepository equipoRepository) { no usages LuisaRom
16         this.equipoRepository = equipoRepository;
17     }
18
19     @GetMapping no usages LuisaRom
20     public List<Equipo> getAllEquipos() {
21         return equipoRepository.findAll();
22     }
23
24     @PostMapping no usages LuisaRom
25     public Equipo crearEquipo(@RequestBody Equipo equipo) {
26         return equipoRepository.save(equipo);
27     }
28 }
```

```
EstadisticasJugadorController.java x
1 package com.example.Parcial2.Controller;
2
3 > import ...
4
5
6
7
8
9 @RestController no usages LuisaRom
10 @RequestMapping("/estadisticas")
11 public class EstadisticasJugadorController {
12
13     private final EstadisticasJugadorRepository estadisticasRepository; 3 usages
14
15     public EstadisticasJugadorController(EstadisticasJugadorRepository estadisticasRepository) { no usages LuisaRom
16         this.estadisticasRepository = estadisticasRepository;
17     }
18
19     @GetMapping no usages LuisaRom
20     public List<EstadisticasJugador> getAllEstadisticas() {
21         return estadisticasRepository.findAll();
22     }
23
24     @PostMapping no usages LuisaRom
25     public EstadisticasJugador crearEstadistica(@RequestBody EstadisticasJugador estadisticas) {
26         return estadisticasRepository.save(estadisticas);
27     }
28 }
```

```
© JugadorController.java x
1 package com.example.Parcial2.Controller;
2 > import ...
7 @RestController no usages LuisaRom
8 @RequestMapping("/jugadores")
9 public class JugadorController {
10
11     private final JugadorRepository jugadorRepository; 5 usages
12
13     public JugadorController(JugadorRepository jugadorRepository) { no usages LuisaRom
14         this.jugadorRepository = jugadorRepository;
15     }
16
17     @GetMapping no usages LuisaRom
18     public List<Jugador> getAllJugadores() {
19         return jugadorRepository.findAll();
20     }
21
22     @GetMapping("/por-equipo/{idEquipo}") no usages LuisaRom
23     public List<Jugador> getJugadoresPorEquipo(@PathVariable int idEquipo) {
24         return jugadorRepository.findJugadoresByEquipoId(idEquipo);
25     }
26
27     @GetMapping("/con-mas-de/{goles}") no usages LuisaRom
28     public List<Jugador> getJugadoresConMasDe(@PathVariable int goles) {
29         return jugadorRepository.findJugadoresConMasDeXGoles(goles);
30     }
31
32     @PostMapping no usages LuisaRom
33     public Jugador crearJugador(@RequestBody Jugador jugador) {
34         return jugadorRepository.save(jugador);
35     }
36 }
```

```
© PartidoController.java ×
1 package com.example.Parcial2.Controller;
2 > import ...
3
4
5
6
7 @RestController no usages LuisaRom
8 @RequestMapping("/partidos")
9 public class PartidoController {
10
11     private final PartidoRepository partidoRepository; 5 usages
12
13     public PartidoController(PartidoRepository partidoRepository) { no usages LuisaRom
14         this.partidoRepository = partidoRepository;
15     }
16
17     @GetMapping no usages LuisaRom
18     public List<Partido> getAllPartidos() {
19         return partidoRepository.findAll();
20     }
21
22     @PostMapping no usages LuisaRom
23     public Partido crearPartido(@RequestBody Partido partido) {
24         return partidoRepository.save(partido);
25     }
26
27     @GetMapping("/resultados") no usages LuisaRom
28     public List<Object[]> getResultadosConNombresEquipos() {
29         return partidoRepository.findResultadosPartidosConNombres();
30     }
31
32     @GetMapping("/goles-por-equipo") no usages LuisaRom
33     public List<Object[]> getTotalGolesPorEquipo() {
34         return partidoRepository.findTotalGolesPorEquipo();
35     }
36 }
```