

PARCIAL2

Valeria Medina Ocampo

ID:834515

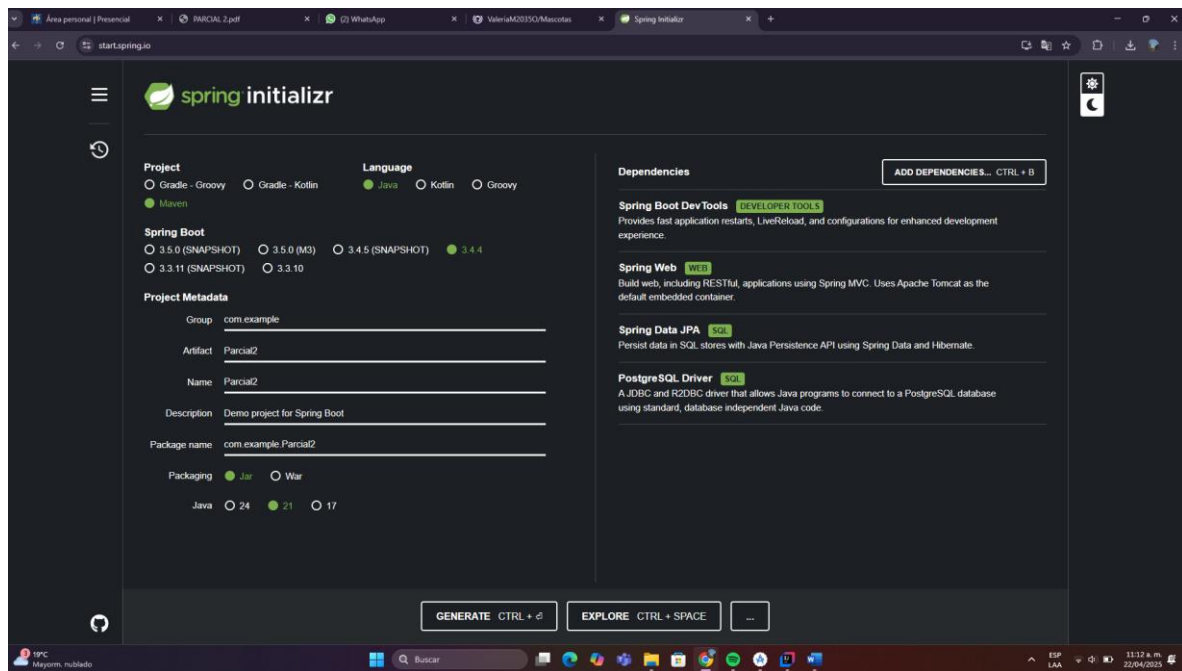
Ingeniería de sistemas, Facultad de ingeniería, Corporación Universitaria Minuto de Dios

NRC 60-66167: Arquitectura de software

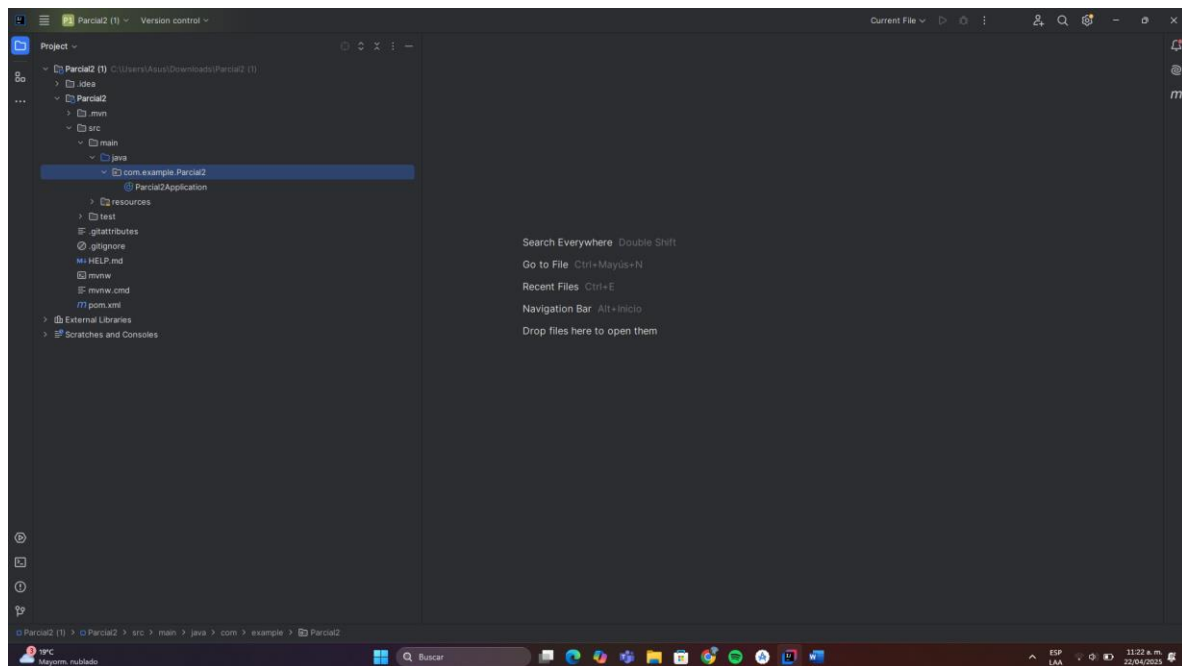
Prof. William Alexander Matallana Porras

Abril 24, 2025

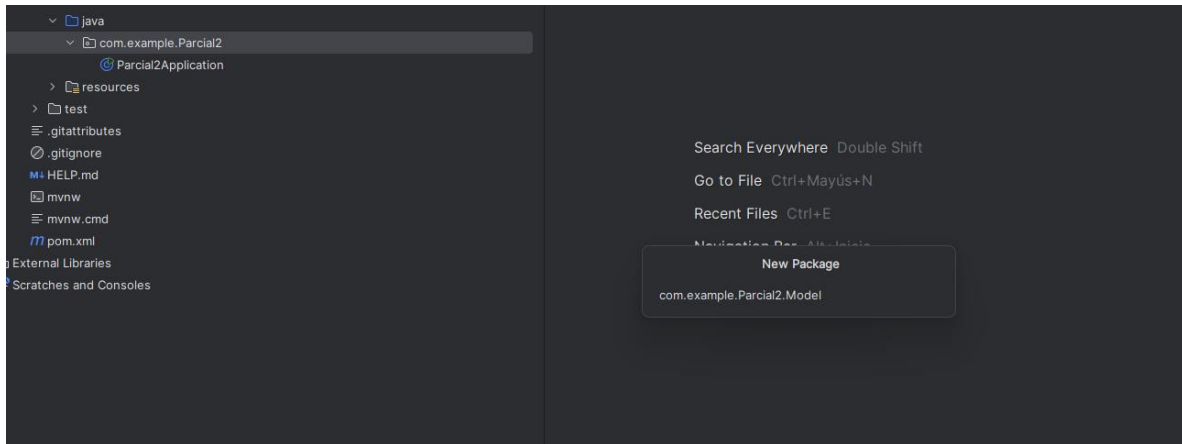
1 paso: Crear el spring boot del proyecto con sus respectivas dependencias.



2 paso: Se abre el proyecto ya descomprimido en intelliJ.



3 paso: Se crea un paquete con el nombre Model.



4 paso: Dentro de ese paquete se crean las siguientes clases:

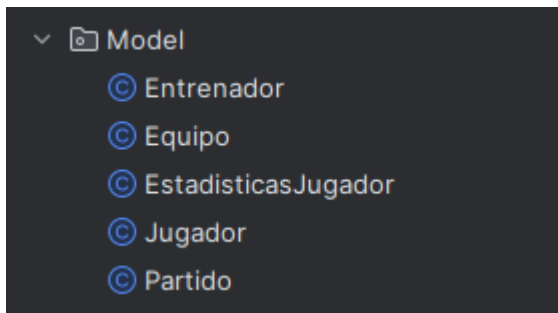
Equipo

Jugador

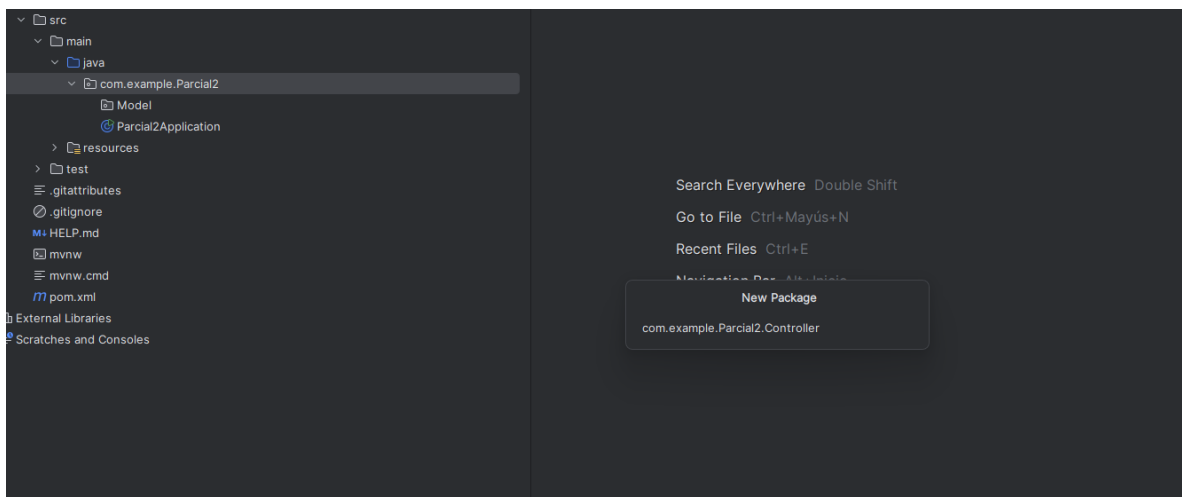
Entrenador

Partido

EstadisticasJugador



5 paso: Se crea un paquete con el nombre Controller.



6 paso: Dentro de ese paquete se crean las siguientes clases:

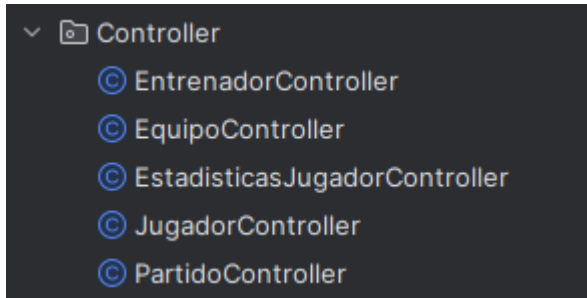
EquipoController

JugadorController

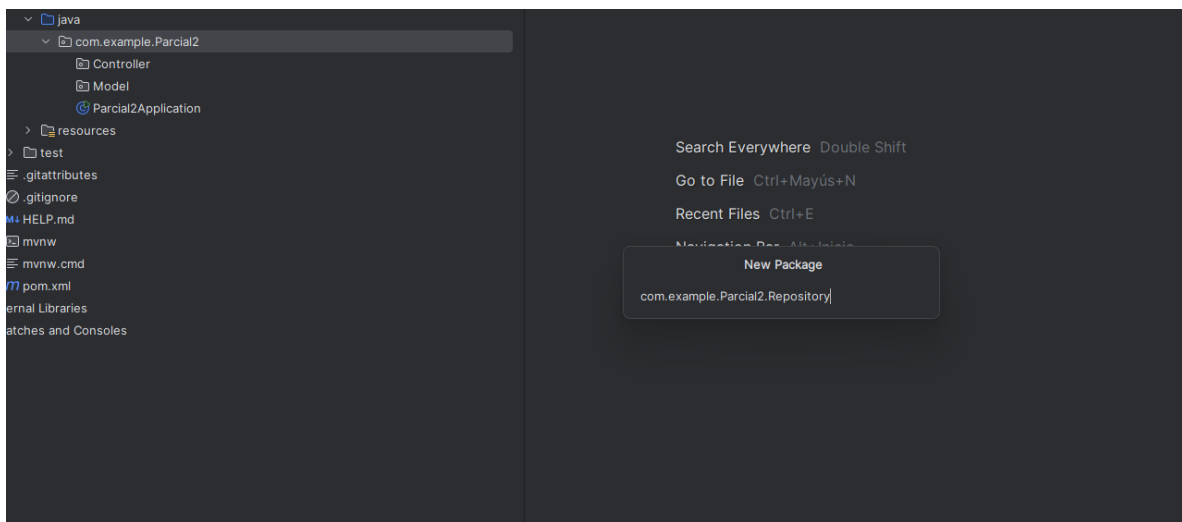
EntrenadorController

PartidoController

EstadisticasJugadorController



7 paso: Se crea un paquete con el nombre Repository.



8 paso: Dentro de ese paquete se crean los siguientes repositorios:

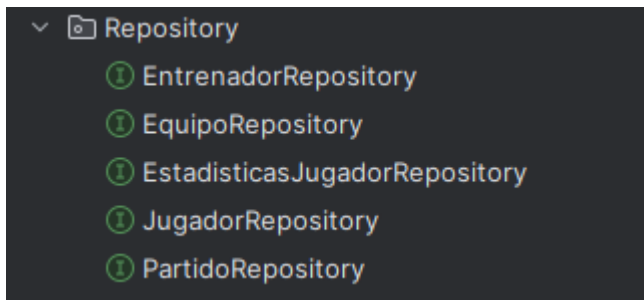
EquipoRepository

JugadorRepository

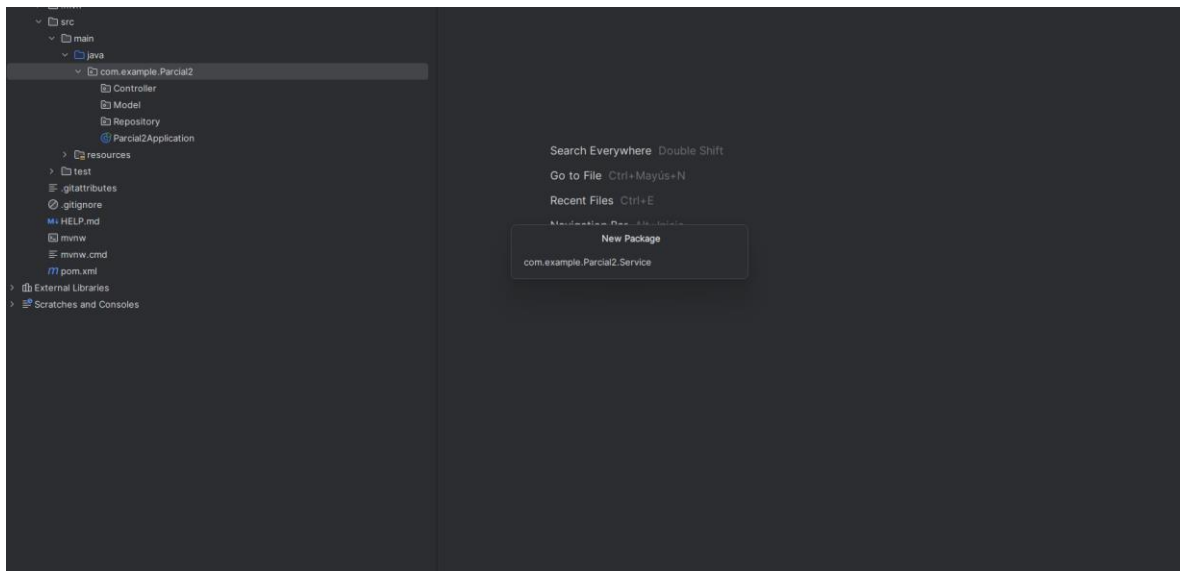
EntrenadorRepository

PartidoRepository

EstadisticasJugadorRepository



9 paso: Se crea un paquete con el nombre Service.



10 paso: Dentro de ese paquete se crean las siguientes clases:

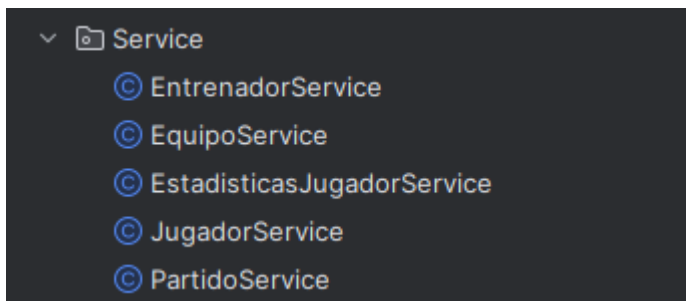
EquipoService

JugadorService

EntrenadorService

PartidoService

EstadisticasJugadorService



11 paso: Definimos cuales van a ser nuestras relaciones.

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

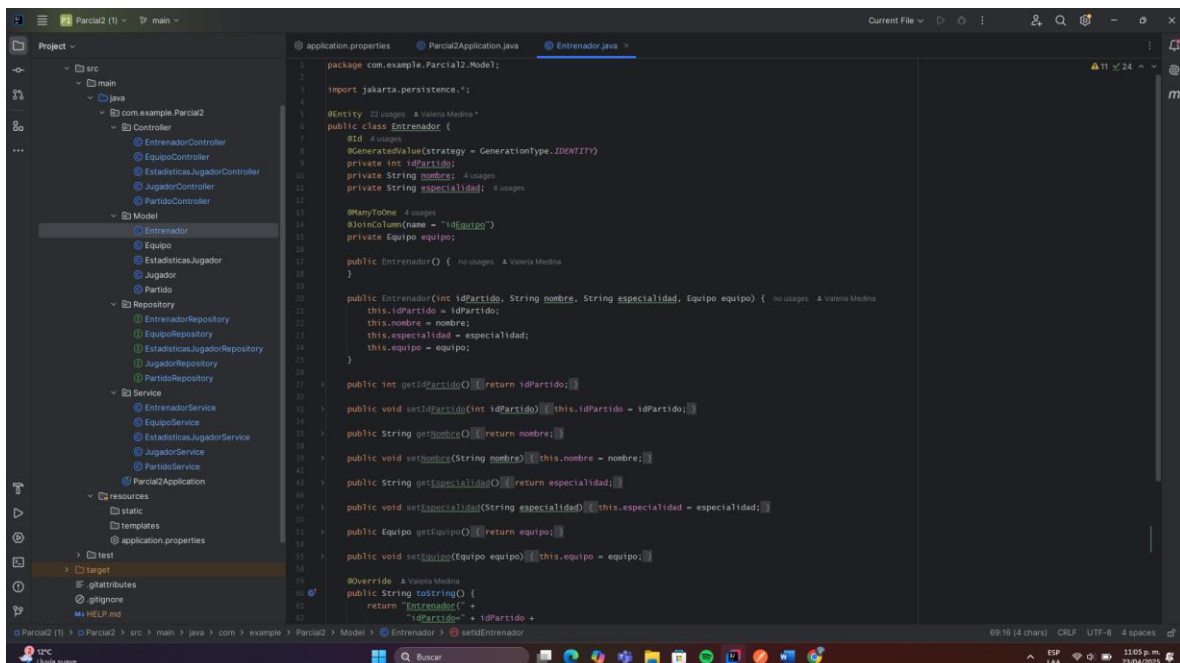
- **Relación:** Un jugador puede tener **muchas estadísticas en diferentes partidos**
- **Cardinalidad:** 1 a N

Partido — EstadísticasJugador

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

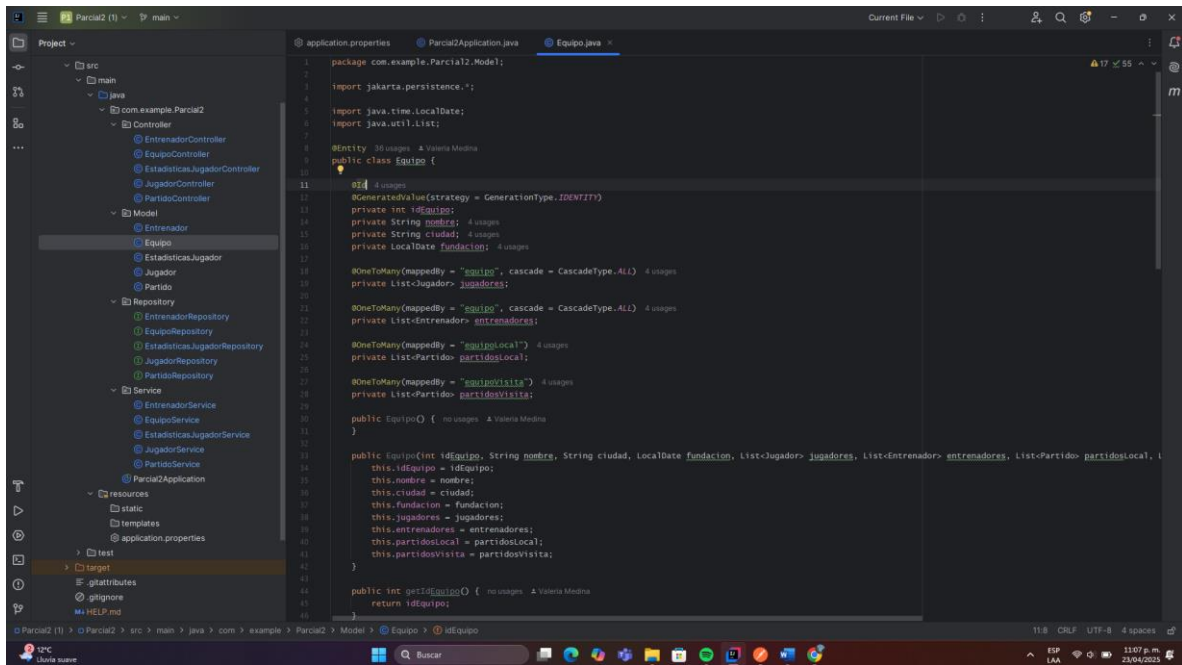
12 paso: En model creamos los atributos y las relaciones de cada clase además de generar los constructores, getters and setter y toString:

Clase Entrenador



```
1 package com.example.Parcial2.Model;
2
3 import jakarta.persistence.*;
4
5 @Entity @Table(name = "Entrenador")
6 public class Entrenador {
7     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
8     private int idPartido;
9     private String nombre;
10    private String especialidad;
11
12    @ManyToOne
13    @JoinColumn(name = "idPartido")
14    private Equipo equipo;
15
16    public Entrenador() {}
17
18    public Entrenador(int idPartido, String nombre, String especialidad, Equipo equipo) {
19        this.idPartido = idPartido;
20        this.nombre = nombre;
21        this.especialidad = especialidad;
22        this.equipo = equipo;
23    }
24
25    public int getIdPartido() { return idPartido; }
26
27    public void setIdPartido(int idPartido) { this.idPartido = idPartido; }
28
29    public String getNombre() { return nombre; }
30
31    public void setNombre(String nombre) { this.nombre = nombre; }
32
33    public String getEspecialidad() { return especialidad; }
34
35    public void setEspecialidad(String especialidad) { this.especialidad = especialidad; }
36
37    public Equipo getEquipo() { return equipo; }
38
39    public void setEquipo(Equipo equipo) { this.equipo = equipo; }
40
41    @Override
42    public String toString() {
43        return "Entrenador{" +
44            "idPartido=" + idPartido +
45            "nombre=" + nombre +
46            "especialidad=" + especialidad +
47            "equipo=" + equipo +
48            "}";
49    }
50}
```

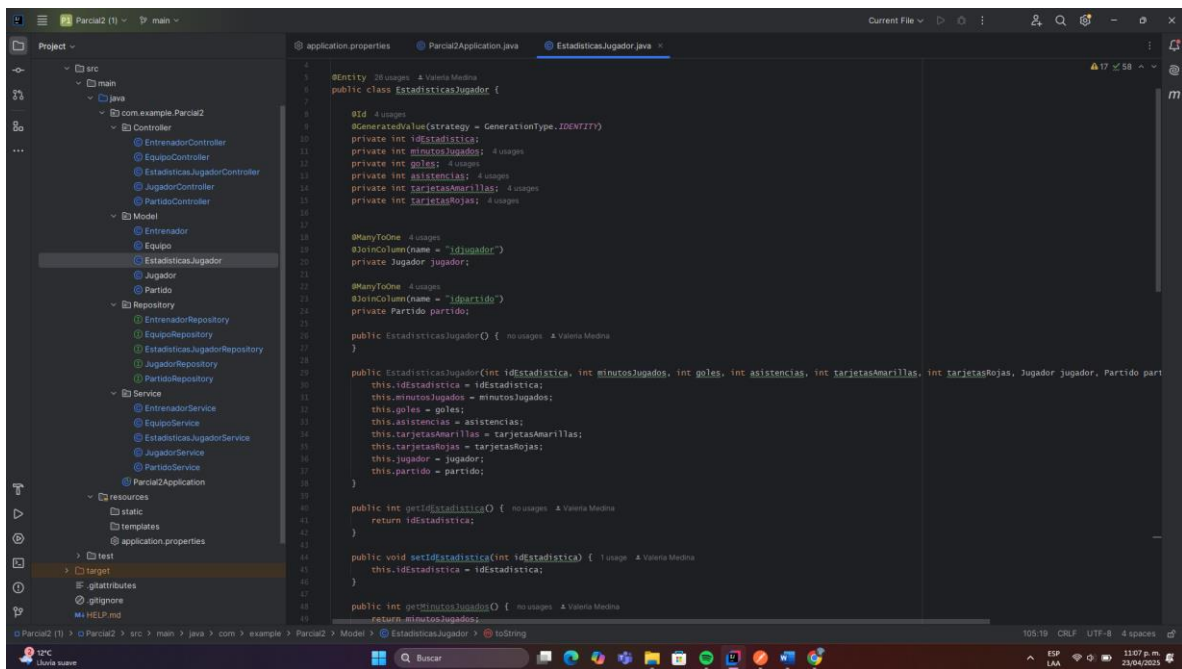
Clase Equipo



The screenshot shows an IDE with the 'Equipo.java' file open. The project structure on the left includes 'src', 'main', 'com.example.Parcial2', 'Controller', 'Model', 'Repository', 'Service', 'resources', 'static', 'templates', 'application.properties', 'test', 'target', '.gitattributes', '.gitignore', and 'HELP.md'. The 'Equipo.java' code is as follows:

```
1 package com.example.Parcial2.Model;
2
3 import jakarta.persistence.*;
4
5 import java.time.LocalDate;
6 import java.util.List;
7
8 @Entity
9 public class Equipo {
10
11     @Id
12     @GeneratedValue(strategy = GenerationType.IDENTITY)
13     private int idEquipo;
14     private String nombre;
15     private String ciudad;
16     private LocalDate fundacion;
17
18     @OneToMany(mappedBy = "equipo", cascade = CascadeType.ALL)
19     private List<Jugador> jugadores;
20
21     @OneToMany(mappedBy = "equipo", cascade = CascadeType.ALL)
22     private List<Entrenador> entrenadores;
23
24     @OneToMany(mappedBy = "equipoLocal", cascade = CascadeType.ALL)
25     private List<Partido> partidosLocal;
26
27     @OneToMany(mappedBy = "equipoVisita", cascade = CascadeType.ALL)
28     private List<Partido> partidosVisita;
29
30     public Equipo() {
31     }
32
33     public Equipo(int idEquipo, String nombre, String ciudad, LocalDate fundacion, List<Jugador> jugadores, List<Entrenador> entrenadores, List<Partido> partidosLocal, List<Partido> partidosVisita) {
34         this.idEquipo = idEquipo;
35         this.nombre = nombre;
36         this.ciudad = ciudad;
37         this.fundacion = fundacion;
38         this.jugadores = jugadores;
39         this.entrenadores = entrenadores;
40         this.partidosLocal = partidosLocal;
41         this.partidosVisita = partidosVisita;
42     }
43
44     public int getIdEquipo() {
45         return idEquipo;
46     }
47 }
```

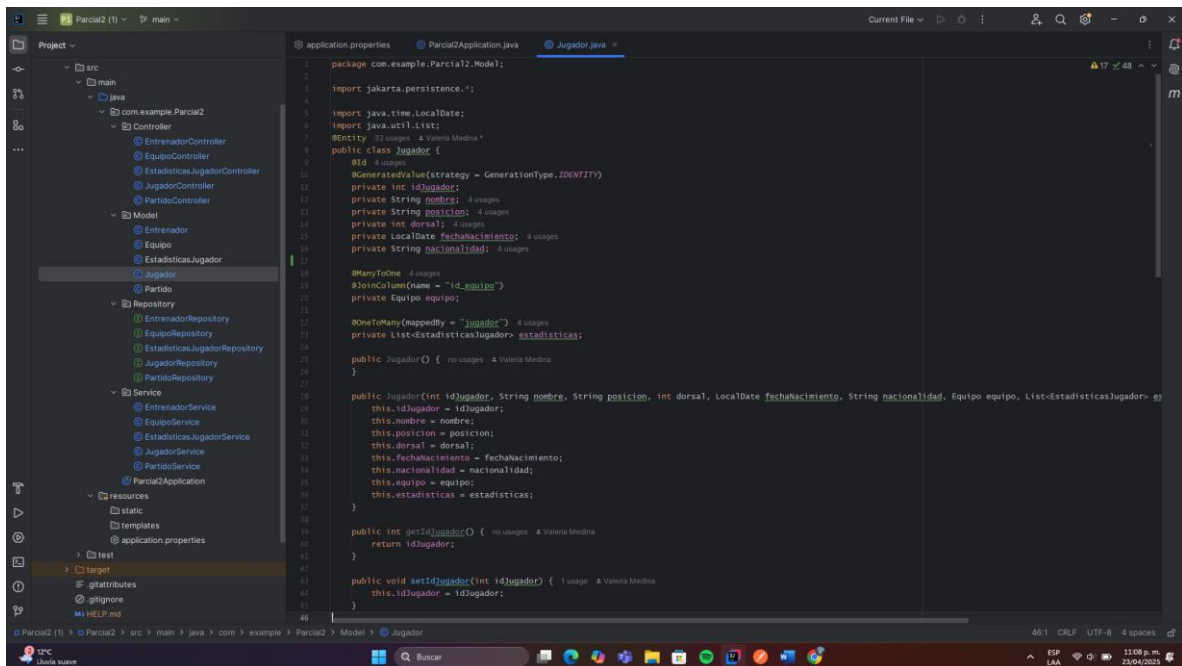
Clase EstadisticasJugador



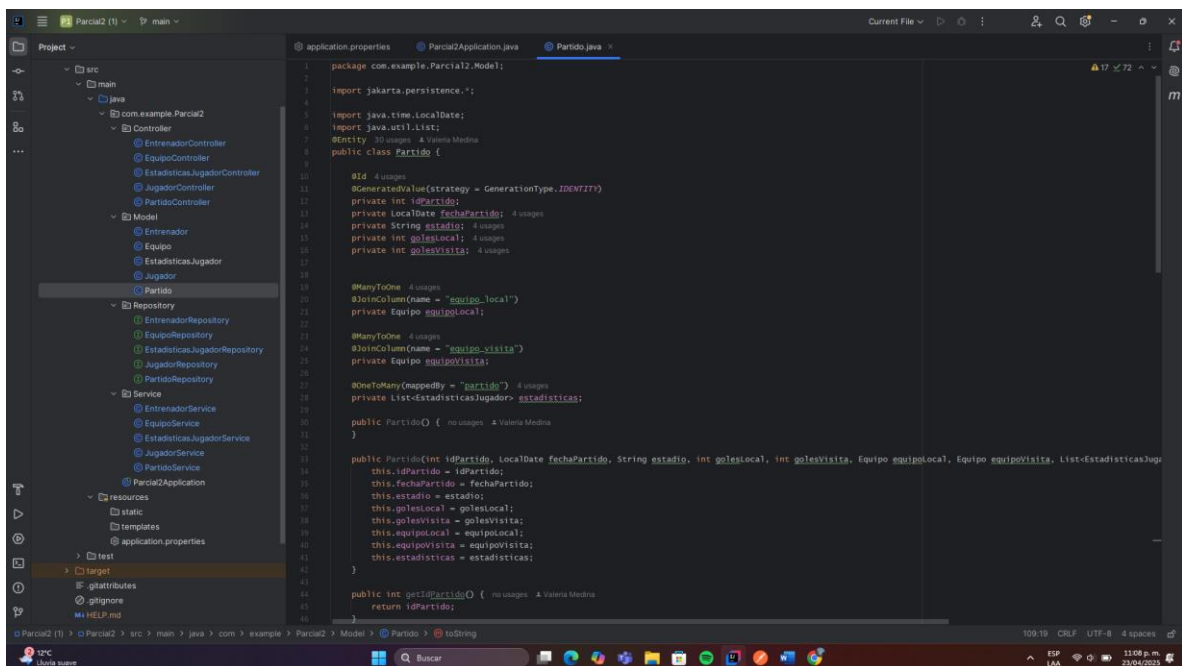
The screenshot shows an IDE with the 'EstadisticasJugador.java' file open. The project structure on the left is the same as the previous screenshot. The 'EstadisticasJugador.java' code is as follows:

```
1 @Entity
2 public class EstadisticasJugador {
3
4     @Id
5     @GeneratedValue(strategy = GenerationType.IDENTITY)
6     private int idEstadistica;
7     private int minutosJugados;
8     private int goles;
9     private int asistencias;
10    private int tarjetasAmarillas;
11    private int tarjetas Rojas;
12
13    @ManyToOne
14    @JoinColumn(name = "idJugador")
15    private Jugador jugador;
16
17    @ManyToOne
18    @JoinColumn(name = "idPartido")
19    private Partido partido;
20
21    public EstadisticasJugador() {
22    }
23
24    public EstadisticasJugador(int idEstadistica, int minutosJugados, int goles, int asistencias, int tarjetasAmarillas, int tarjetasRojas, Jugador jugador, Partido partido) {
25        this.idEstadistica = idEstadistica;
26        this.minutosJugados = minutosJugados;
27        this.goles = goles;
28        this.asistencias = asistencias;
29        this.tarjetasAmarillas = tarjetasAmarillas;
30        this.tarjetasRojas = tarjetasRojas;
31        this.jugador = jugador;
32        this.partido = partido;
33    }
34
35    public int getIdEstadistica() {
36        return idEstadistica;
37    }
38
39    public void setIdEstadistica(int idEstadistica) {
40        this.idEstadistica = idEstadistica;
41    }
42
43    public int getMinutosJugados() {
44        return minutosJugados;
45    }
46 }
```

Clase Jugador



Clase Partido



13 paso: En Controller creamos los métodos que se van a usar para las consultas como guardar, listar, actualizar, eliminar y las respectivas consultas nativas, los controller siempre van relacionados con los service de cada uno:

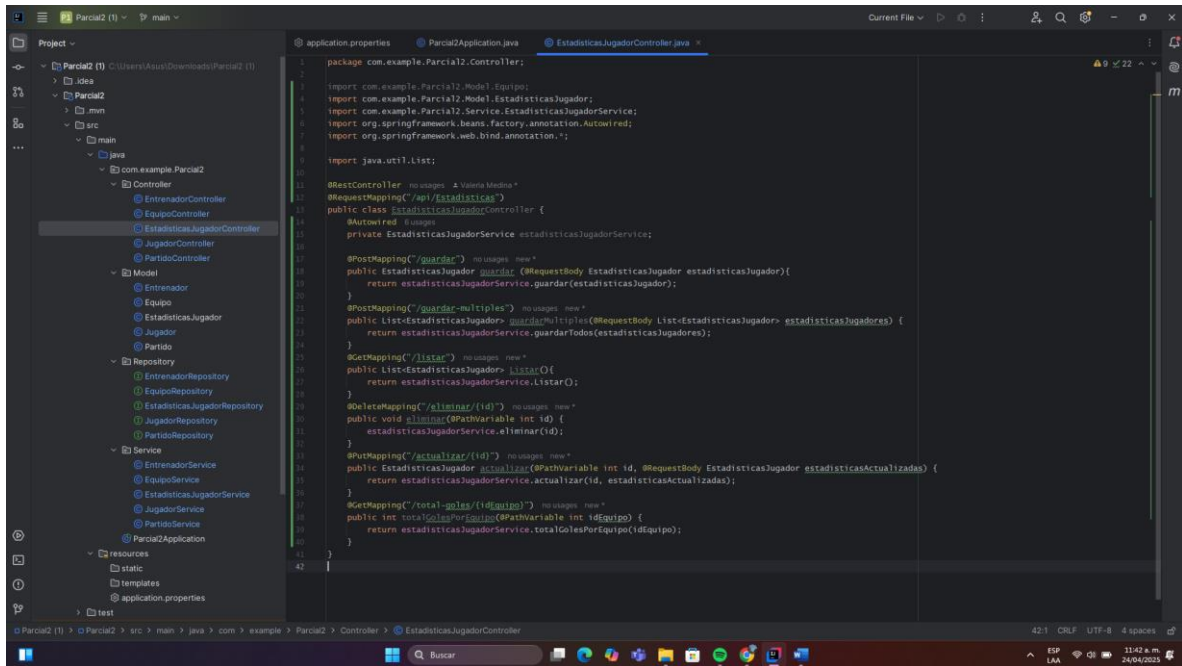
Clase EntrenadorController


```
1 package com.example.Parcial2.Controller;
2
3 import com.example.Parcial2.Model.Entrenador;
4 import com.example.Parcial2.Service.EntrenadorService;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.*;
7
8 import java.util.List;
9
10 @RestController
11 @RequestMapping("/api/entrenador")
12 public class EntrenadorController {
13
14     @Autowired
15     private EntrenadorService entrenadorService;
16
17     @PostMapping("/guardar")
18     public Entrenador guardar(@RequestBody Entrenador entrenador){
19         return entrenadorService.guardar(entrenador);
20     }
21
22     @PostMapping("/guardar-multiples")
23     public List<Entrenador> guardarMultiples(@RequestBody List<Entrenador> entrenadores){
24         return entrenadorService.guardarTodos(entrenadores);
25     }
26
27     @GetMapping("/listar")
28     public List<Entrenador> listar() {
29         return entrenadorService.listar();
30     }
31
32     @DeleteMapping("/eliminar/{id}")
33     public void eliminar(@PathVariable int id){
34         entrenadorService.eliminar(id);
35     }
36
37     @PutMapping("/actualizar/{id}")
38     public Entrenador actualizar(@PathVariable int id, @RequestBody Entrenador entrenadorActualizado){
39         return entrenadorService.actualizar(id, entrenadorActualizado);
40     }
41 }
```

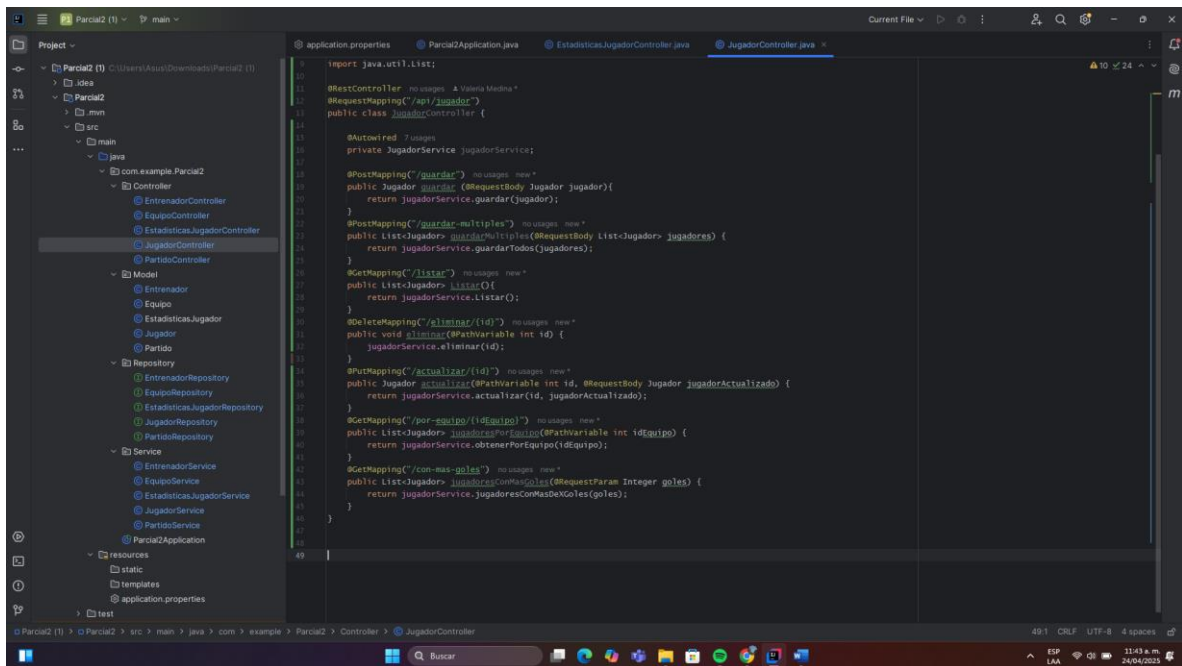
Clase EquipoController

```
1 package com.example.Parcial2.Controller;
2
3 import com.example.Parcial2.Model.Equipo;
4 import com.example.Parcial2.Service.EquipoService;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.*;
7
8 import java.util.List;
9
10 @RestController
11 @RequestMapping("/api/equipo")
12 public class EquipoController {
13
14     @Autowired
15     private EquipoService equipoService;
16
17     @PostMapping("/guardar")
18     public Equipo guardar(@RequestBody Equipo equipo){
19         return equipoService.guardar(equipo);
20     }
21
22     @PostMapping("/guardar-multiples")
23     public List<Equipo> guardarMultiples(@RequestBody List<Equipo> equipos){
24         return equipoService.guardarTodos(equipos);
25     }
26
27     @GetMapping("/listar")
28     public List<Equipo> listar() {
29         return equipoService.listar();
30     }
31
32     @DeleteMapping("/eliminar/{id}")
33     public void eliminar(@PathVariable int id){
34         equipoService.eliminar(id);
35     }
36
37     @PutMapping("/actualizar/{id}")
38     public Equipo actualizar(@PathVariable int id, @RequestBody Equipo equipoActualizado){
39         return equipoService.actualizar(id, equipoActualizado);
40     }
41 }
```

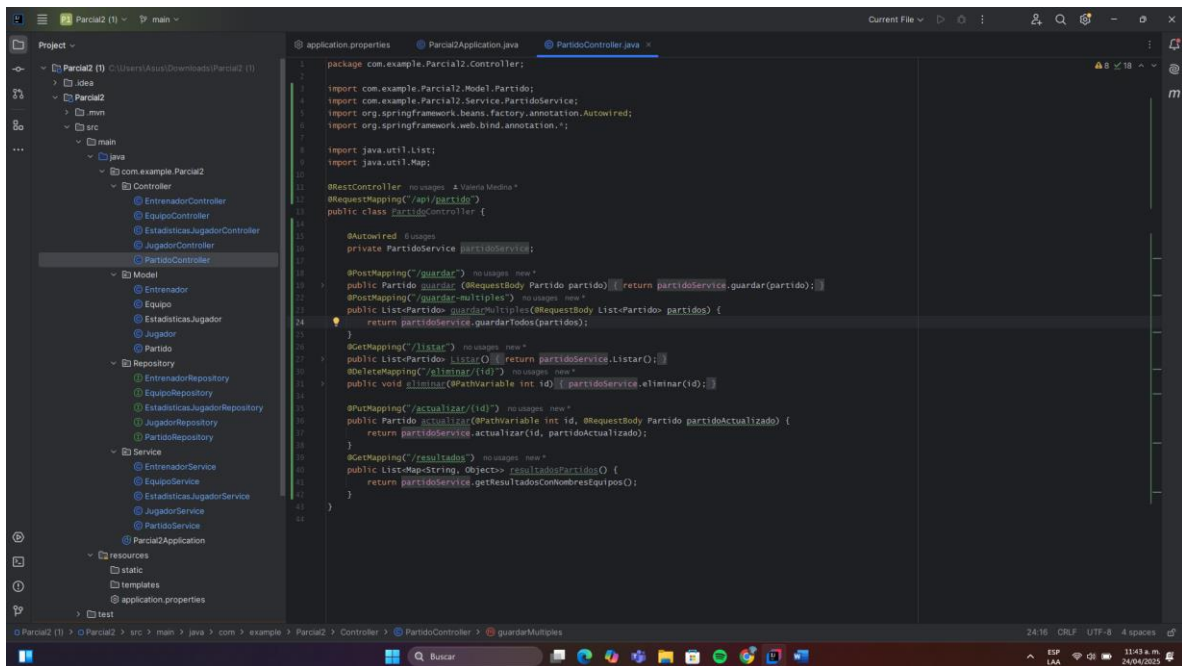
Clase EstadisticasJugadorController



Clase JugadorController

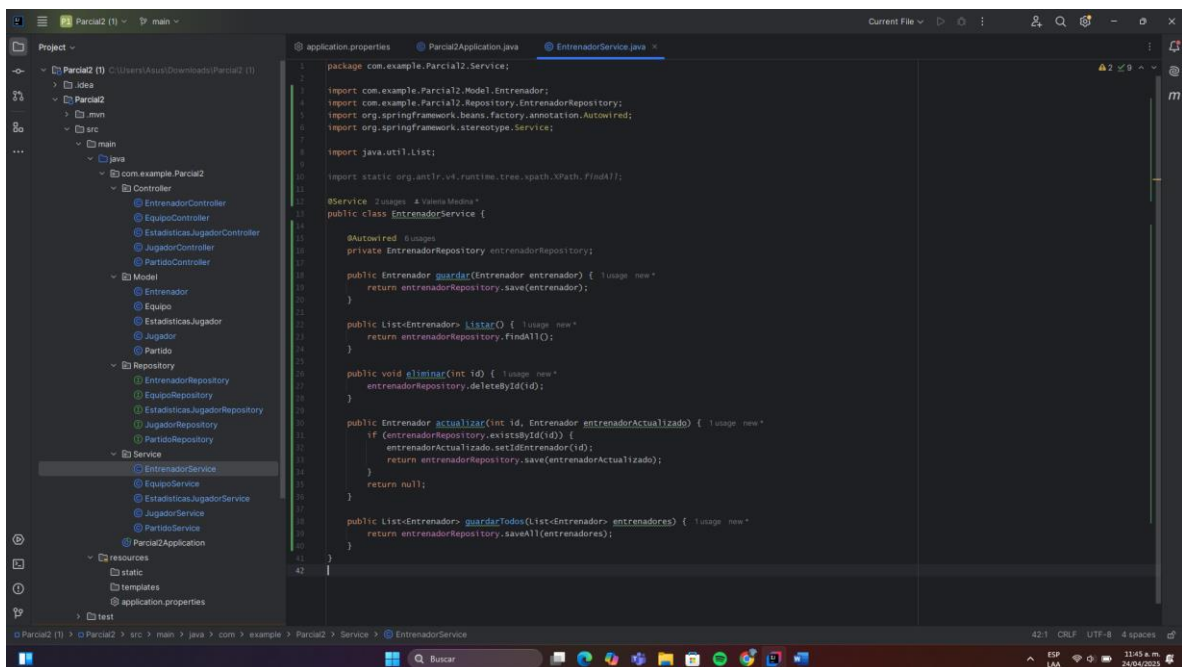


Clase PartidoController

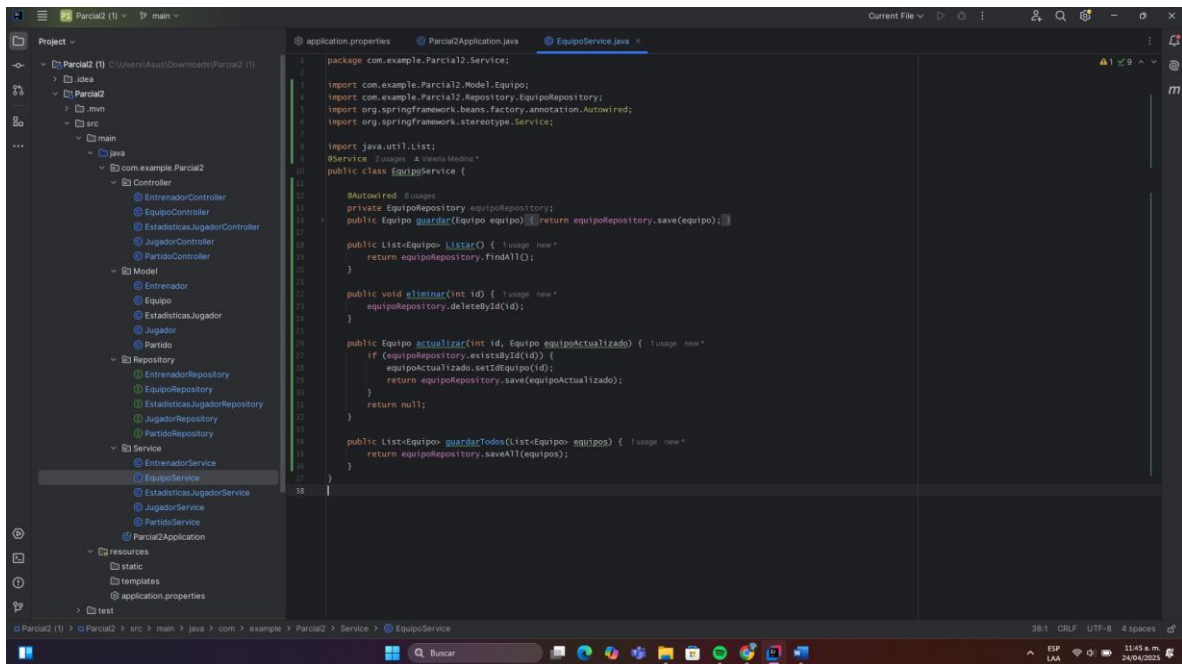


14 paso: En Service inicializamos los métodos que creamos en controller y las respectivas consultas nativas, los Service siempre van relacionados con los Repository de cada uno:

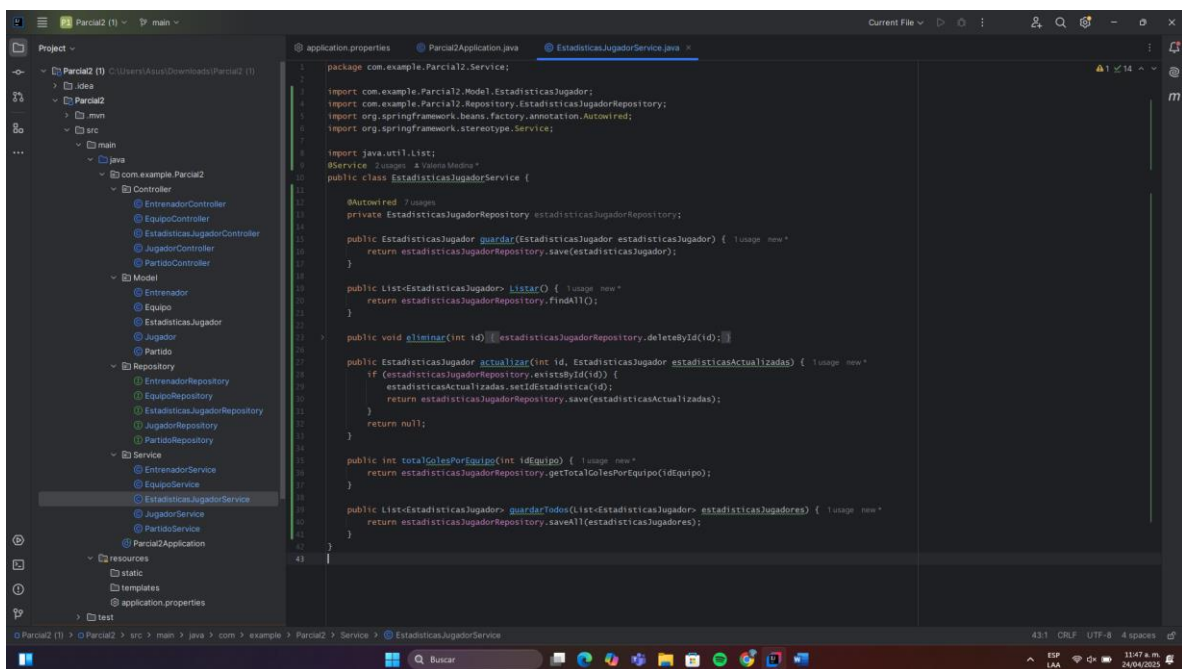
Clase EntrenadorService



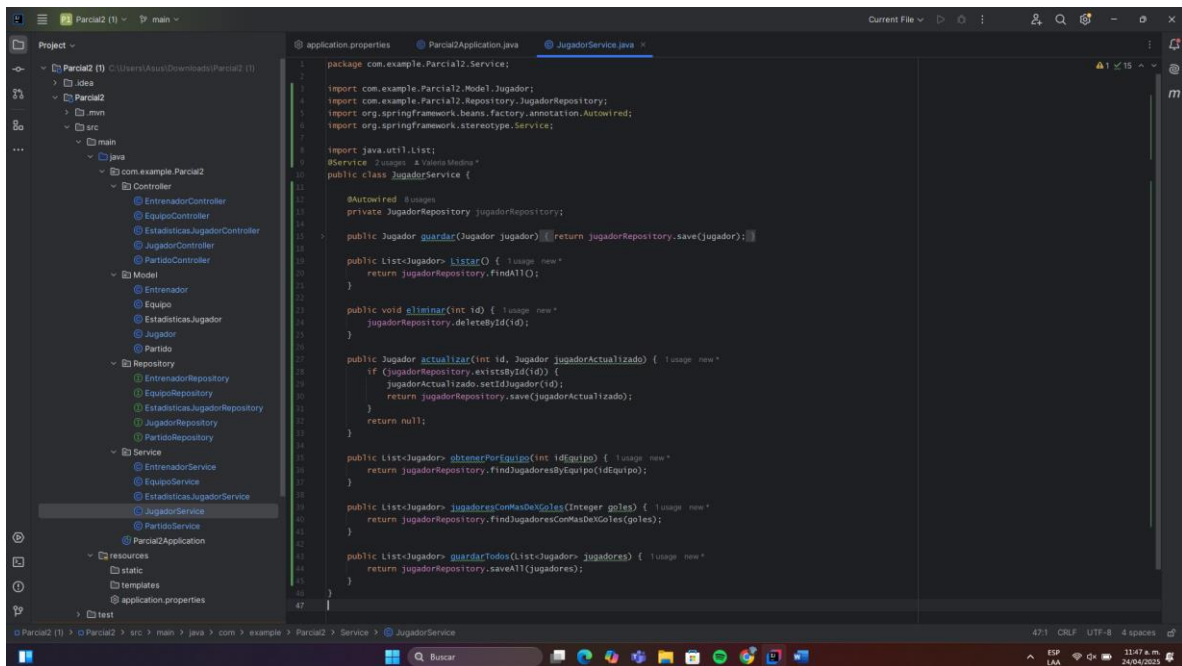
Clase EquipoService



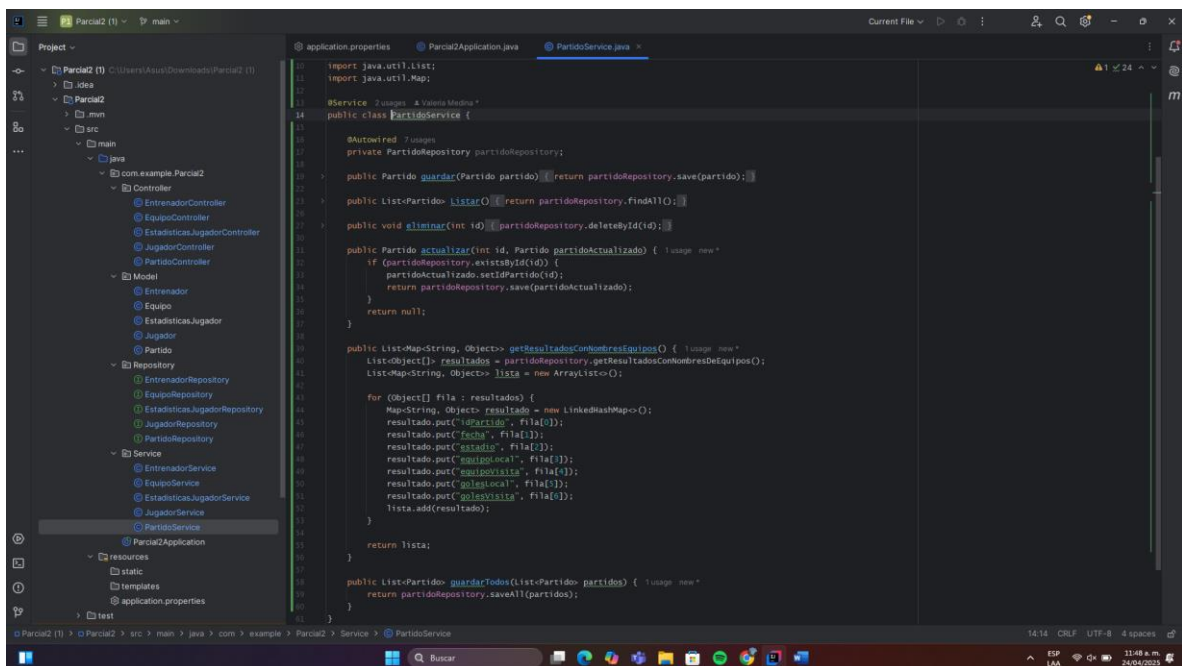
Clase EstadisticasJugadorService



Clase JugadorService

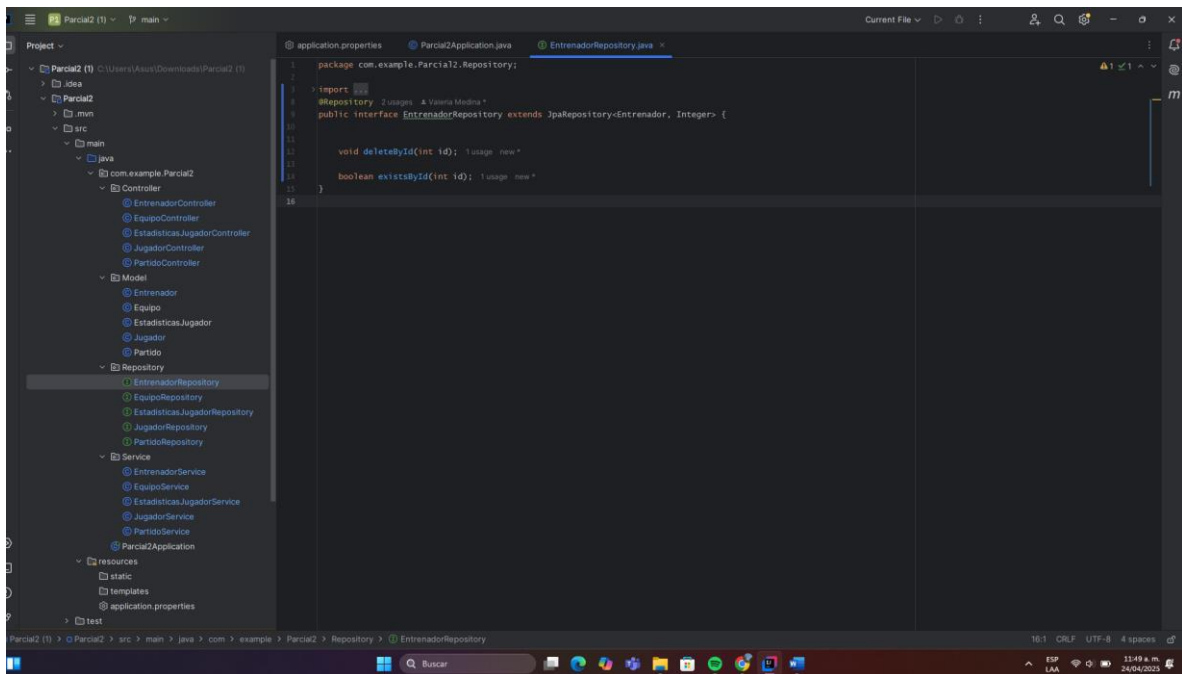


Clase PartidoService

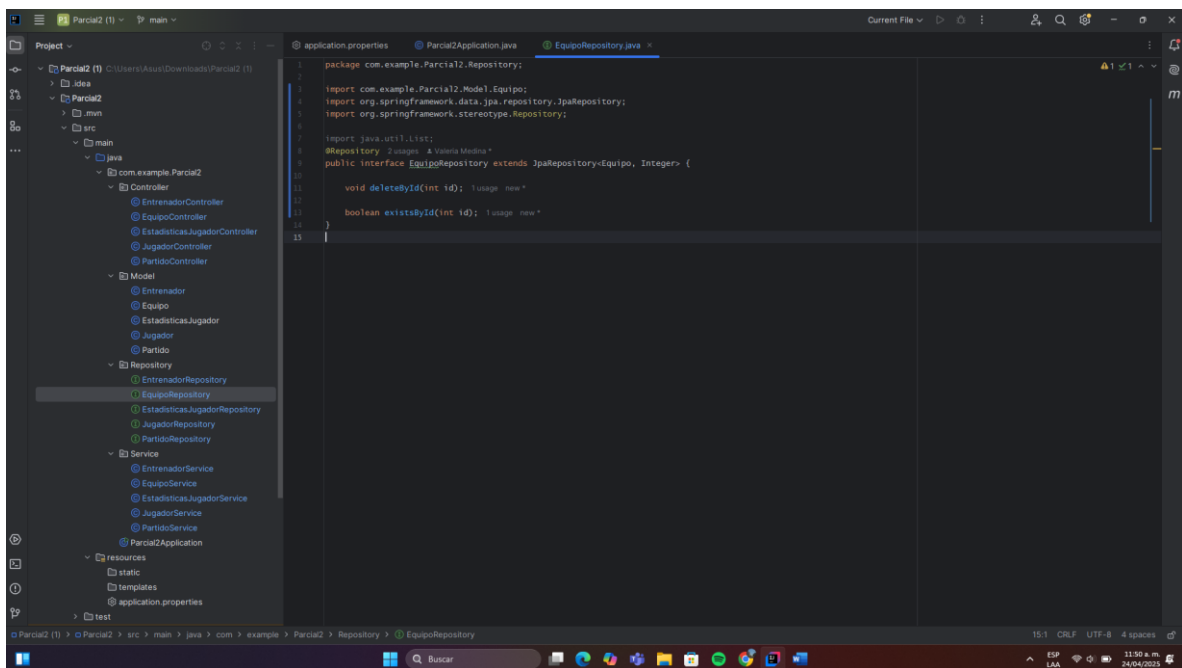


15 paso: Y por ultimo tenemos las interfaces de los repositorios donde están inicializadas también las 4 consultas nativas respectivamente.

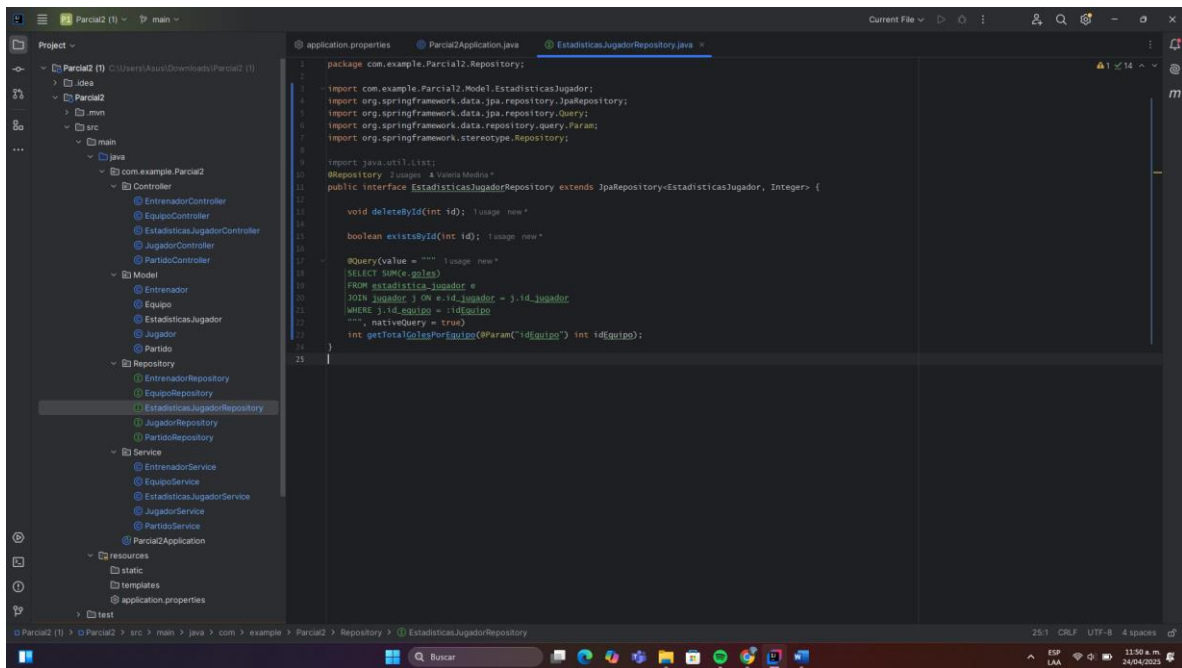
Clase EntrenadorRepository



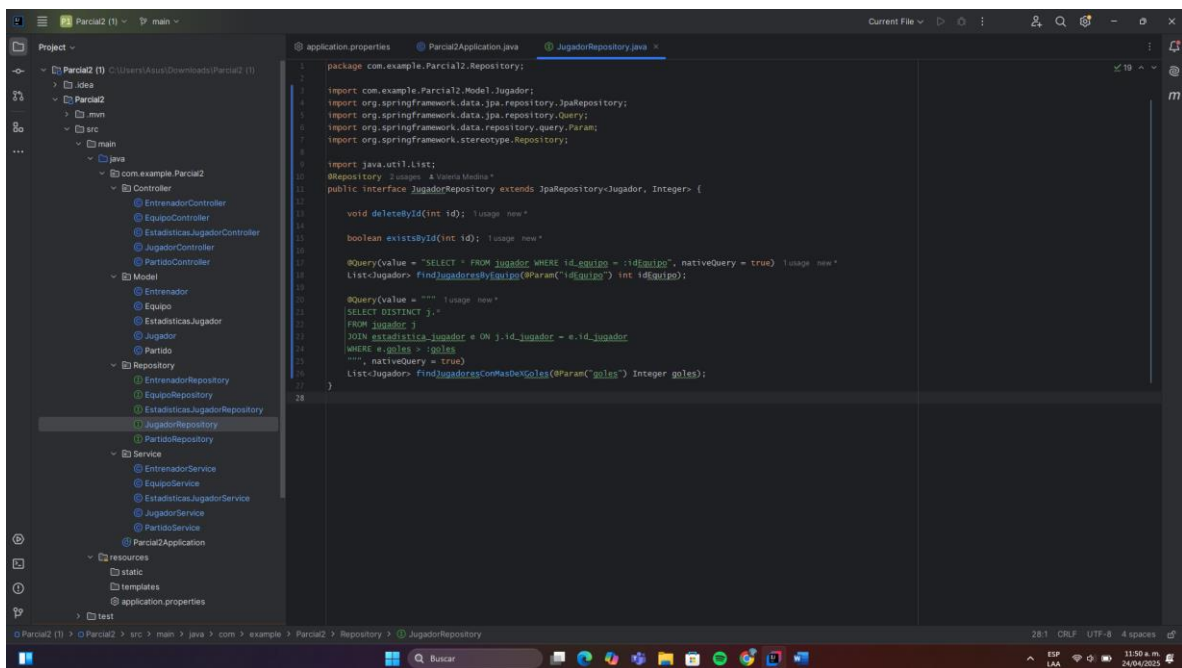
Clase EquipoRepository



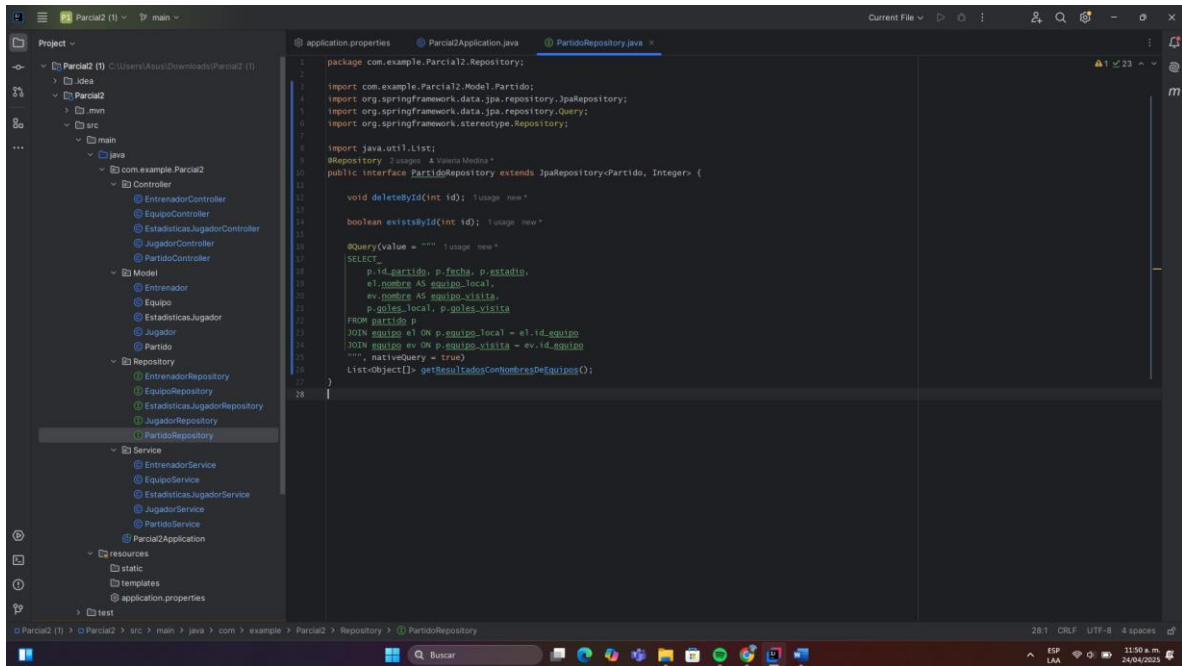
Clase EstadisticasJugadorRepository



Clase JugadorRepository



Clase PartidoRepository



SUPABASE

Filter	Sort	Insert	id	ciudad	fundacion	nombre
			2	Barcelona	1951-01-10	Barcelona FC
			3	Atlético	1976-12-06	Atlético de Madrid
			4	Valencia	1901-12-24	Valencia CF
			5	Sevilla	2015-12-30	Sevilla FC
			6	Manchester	1963-07-19	Manchester United
			7	Liverpool	1938-01-06	Liverpool FC
			8	Chelsea	1908-10-12	Chelsea FC
			9	Arsenal	1924-07-26	Arsenal FC
			10	Tottenham	1922-01-27	Tottenham Hotspur
			11	Bayern	1955-05-13	Bayern Munich
			12	Borussia	1953-12-21	Borussia Dortmund
			13	RB	1957-07-07	RB Leipzig
			14	Eintracht	1910-04-26	Eintracht Frankfurt
			15	Bayer	1940-03-01	Bayer Leverkusen
			16	Juventus	1990-11-01	Juventus
			17	AC	1999-11-08	AC Milan
			18	Inter	1968-07-14	Inter de Milán
			19	Napoli	1905-12-10	Napoli
			20	Roma	1961-11-04	Roma
			21	Paris	1938-03-19	Paris Saint-Germain
			22	Olympique	1940-01-27	Olympique de Marseille
			23	Lyon	2006-05-04	Lyon
		

Page 1 of 1 100 rows 99 records

Table Editor

schema: public

+ New table

Search tables...

entrenador

equipo

estadísticas_jugador

jugador

partido

	especialidad	nombre	id_equipo
1	Psicólogo	Diego Ramirez	18
3	Psicólogo	Manuel Torres	2
4	Preparador Físico	Julio López	3
5	Preparador Físico	Julio Jiménez	5
6	Asistente	David Navarro	4
7	Asistente	Luis Rodríguez	6
8	Principal	Javier Ramirez	7
9	Psicólogo	Oscar López	9
10	Analista	Javier Vargas	8
11	Principal	Eduardo Pérez	10
12	Asistente	Javier Díaz	11
13	Preparador Físico	Alejandro García	12
14	Preparador Físico	Eduardo Ramirez	13
15	Analista	Julio Torres	16
16	Psicólogo	Alejandro Cruz	15
17	Analista	Pablo Sánchez	14
18	Analista	Carlos Gómez	17
19	Preparador Físico	Oscar Pérez	18
20	Preparador Físico	Miguel García	19
21	Analista	Oscar Martínez	22
22	Preparador Físico	Julio García	20
23	Analista	Luis Torres	21

Page 1 of 1 100 rows 99 records

Refresh Data Definition

Table Editor

schema: public

+ New table

Search tables...

entrenador

equipo

estadísticas_jugador

jugador

partido

	id_estadist...	asistencias	goles	minutos_jugados	tarjetas_amarillas	tarjetas_rojas
1		0	0	54	0	0
2		1	0	30	0	0
3		1	0	71	0	0
4		0	0	79	1	0
5		0	0	23	1	0
6		0	2	22	0	0
7		0	1	65	0	0
8		0	0	47	0	0
9		0	2	37	0	0
10		0	1	11	0	0
11		0	0	23	1	0
12		1	0	62	1	0
13		0	0	23	0	0
14		0	0	5	0	0
15		0	1	65	0	0
16		0	2	48	1	0
17		0	0	56	1	0
18		1	1	86	0	1
19		0	0	88	0	0
20		0	1	80	0	1
21		0	0	72	0	0
22		2	0	55	0	0

ArchitectureFreeParallel2ConnectEnable branchingFeedbackRuntime offAPI Docs

Table Editor

Schema: public

New table

Search tables

entrenador

equipo

estadísticas_jugador

jugador

partido

FilterSortInsert

RLS disabledPostgresRuntime offAPI Docs

	id	dorsal	fecha_nacimiento	nacionalidad	nombre	posicion	id
62	96	1994-05-21	Brasileno	Miguel Vargas	Centrocampista	2	
63	19	2001-03-05	Mexicano	Pedro López	Portero	3	
64	8	1994-11-01	Colombiano	José Pérez	Portero	4	
65	70	1999-10-11	Chileno	Andrés Ramírez	Portero	5	
66	80	1991-08-31	Colombiano	Juan García	Defensa	6	
67	91	2006-01-27	Ecuatoriano	Andrés Castro	Defensa	7	
68	88	2001-12-21	Uruguayo	Javier García	Delantero	8	
69	99	1990-05-24	Ecuatoriano	Pedro Martínez	Defensa	9	
70	70	2003-11-05	Brasileno	José Pérez	Defensa	10	
71	78	2001-09-09	Colombiano	Luis Pérez	Portero	11	
72	97	2002-11-17	Ecuatoriano	Fernando Rodríguez	Portero	12	
73	25	2002-10-11	Ecuatoriano	Luis Martínez	Centrocampista	13	
74	83	1992-05-11	Mexicano	Diego Martínez	Defensa	14	
75	65	1992-10-09	Colombiano	Pedro Torres	Delantero	15	
76	66	1996-02-20	Ecuatoriano	Miguel Rodríguez	Delantero	16	
77	77	1992-05-16	Argentino	Javier Ramírez	Centrocampista	17	
78	79	1992-06-19	Argentino	Diego Ramírez	Delantero	18	
79	24	1991-10-22	Mexicano	José Torres	Portero	19	
80	11	1996-05-19	Peruano	Luis Castro	Delantero	20	
81	54	2000-08-22	Argentino	Miguel López	Delantero	21	
82	29	2002-12-29	Mexicano	Miguel Martínez	Delantero	22	
83	76	1997-10-19	Colombiano	Miguel Martínez	Portero	23	

Page 1 of 1100 rows98 records

RefreshDataDefinition

12:52 p.m.

Architectura

Free

Parcial2

Connect

Enable branching

Feedback

Runtime off

API Docs

Table Editor

schema: public

New table

Search tables

entrenador

equipo

estadísticas_jugador

jugador

partido

Filter

Sort

Insert

id_partido

estadio

fecha_partido

goles_local

goles_visitante

equipo_local

equipo_visitante

1

Estadio Nacional

NULL

0

0

NULL

NULL

2

Arena Olímpica

NULL

0

0

NULL

NULL

3

Arena Olímpica

NULL

0

0

NULL

NULL

4

Estadio Nacional

NULL

0

0

NULL

NULL

5

Estadio del Sol

NULL

0

0

NULL

NULL

6

Fortaleza Central

NULL

0

0

NULL

NULL

7

Coloso del Norte

NULL

0

0

NULL

NULL

8

Arena Olímpica

NULL

0

0

NULL

NULL

9

Fortaleza Central

NULL

0

0

NULL

NULL

10

Arena Olímpica

NULL

0

0

NULL

NULL

11

Arena Olímpica

NULL

0

0

NULL

NULL

12

Coloso del Norte

NULL

0

0

NULL

NULL

13

Arena Olímpica

NULL

0

0

NULL

NULL

14

Estadio Nacional

NULL

0

0

NULL

NULL

15

Fortaleza Central

NULL

0

0

NULL

NULL

16

Estadio del Sol

NULL

0

0

NULL

NULL

17

Fortaleza Central

NULL

0

0

NULL

NULL

18

Arena Olímpica

NULL

0

0

NULL

NULL

19

Arena Olímpica

NULL

0

0

NULL

NULL

20

Estadio del Sol

NULL

0

0

NULL

NULL

21

Fortaleza Central

NULL

0

0

NULL

NULL

22

Estadio del Sol

NULL

0

0

NULL

NULL

Page 1 of 1

100 rows

99 records

Refresh

Data

Definition

16 paso: Ya teniendo todo completo y corriendo correctamente vamos a postman donde iniciaremos ingresando 100 datos por cada tabla:

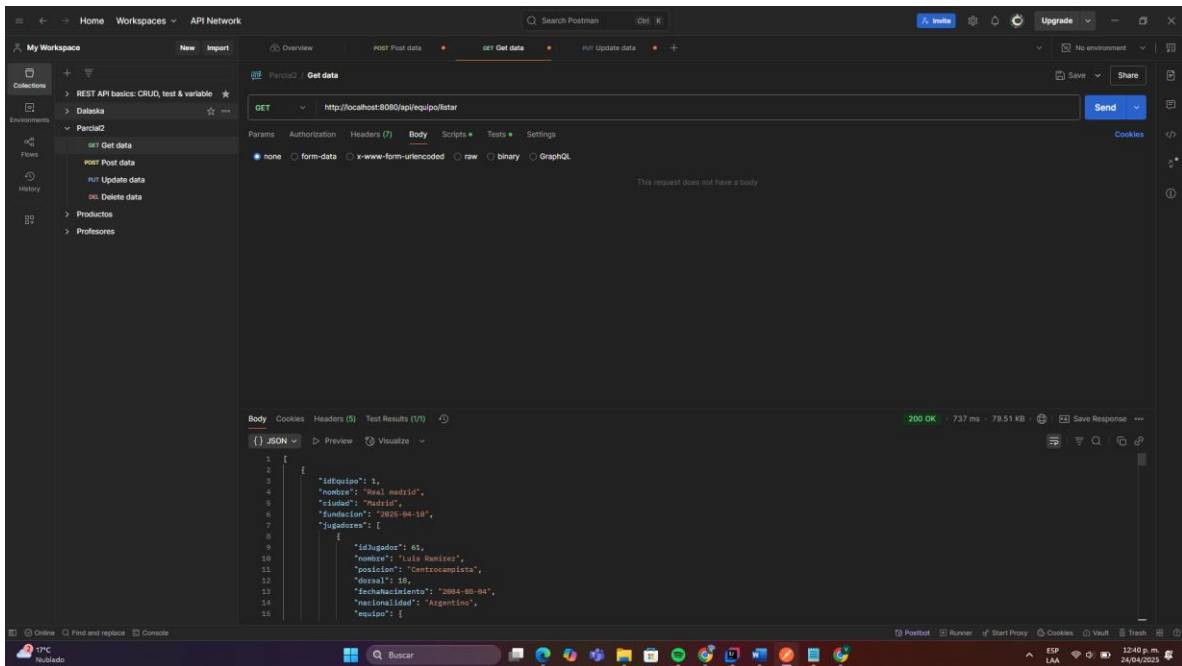
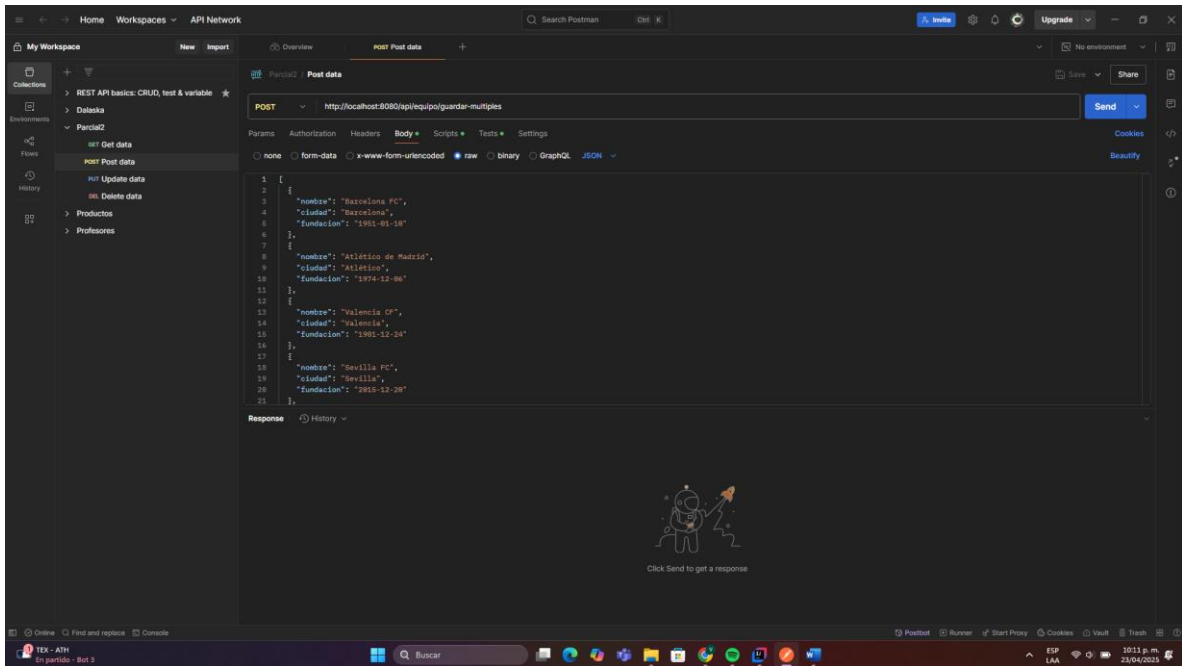
Datos Equipos

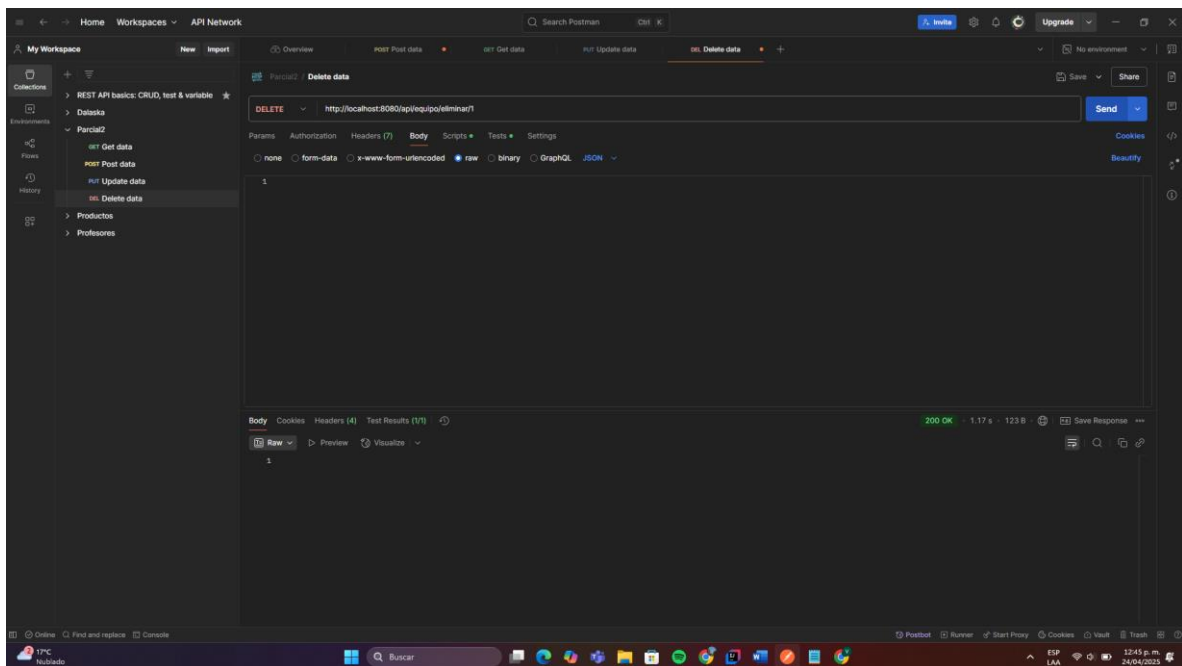
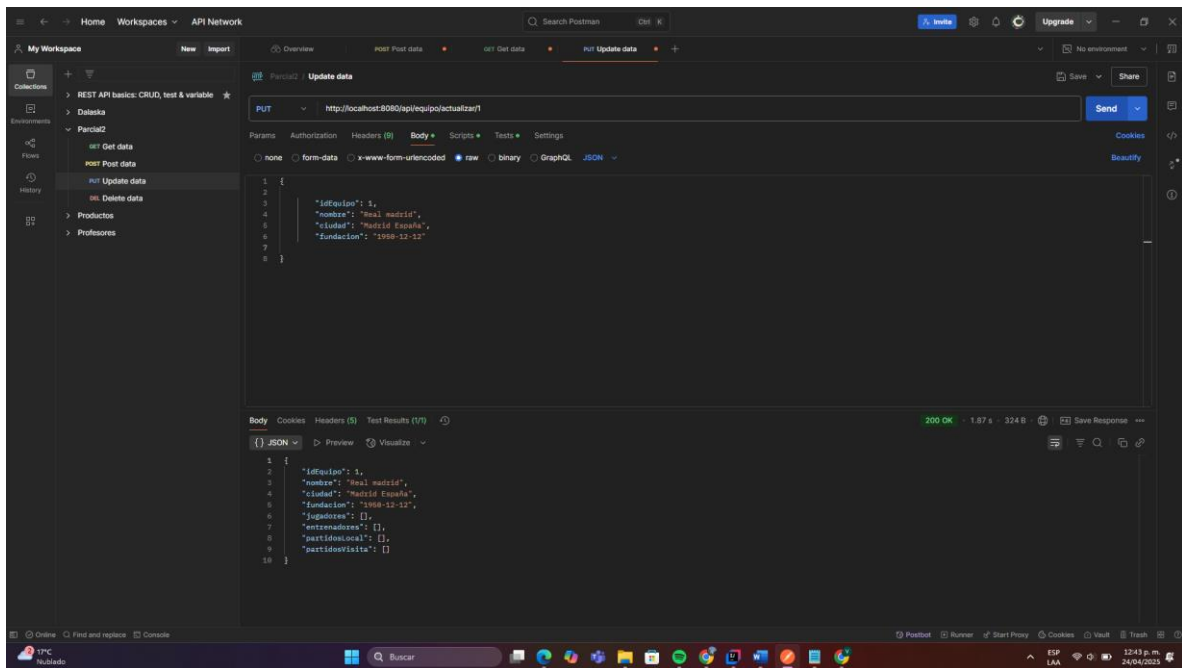
<http://localhost:8080/api/equipo/guardar-multiples>

<http://localhost:8080/api/equipo/eliminar/1>

<http://localhost:8080/api/equipo/actualizar/1>

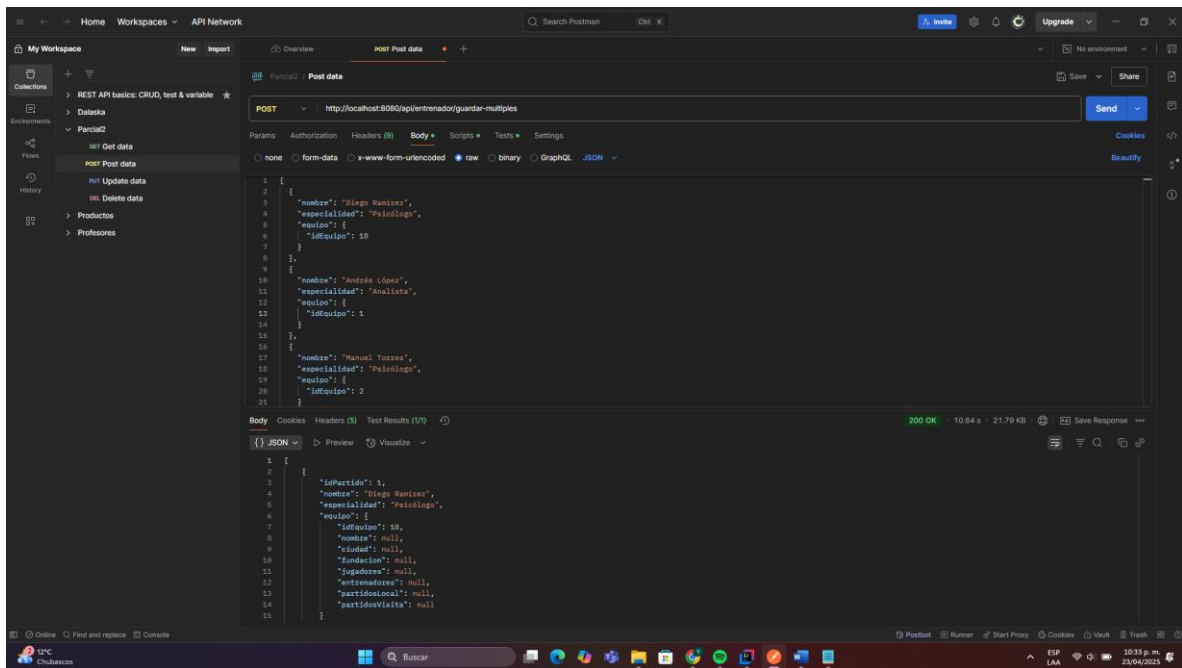
<http://localhost:8080/api/equipo/listar>





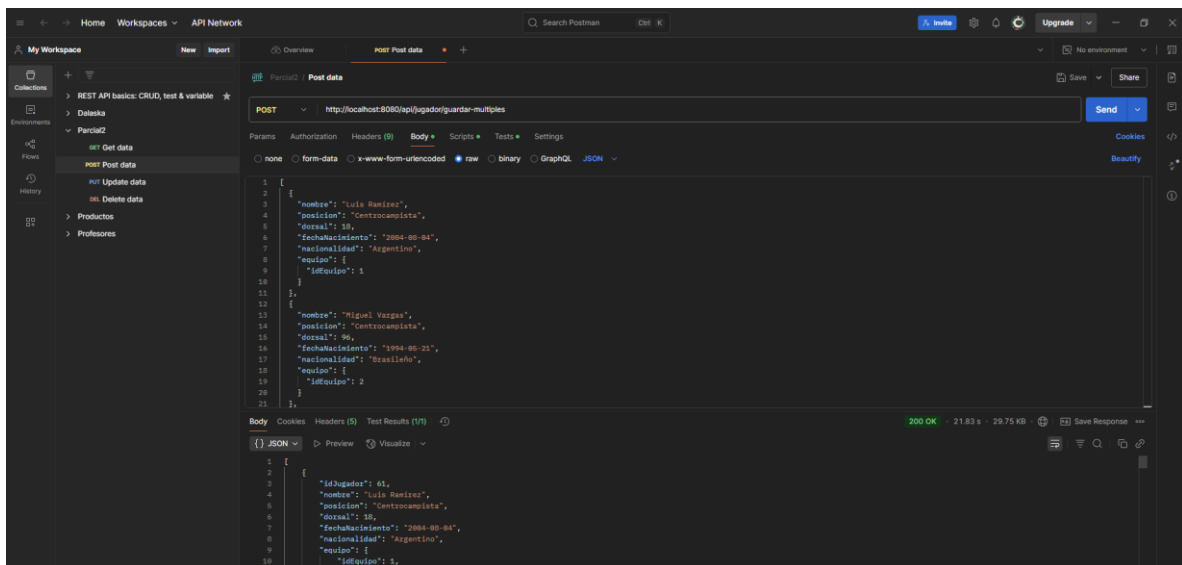
Datos Entrenadores

<http://localhost:8080/api/entrenador/guardar-multiples>



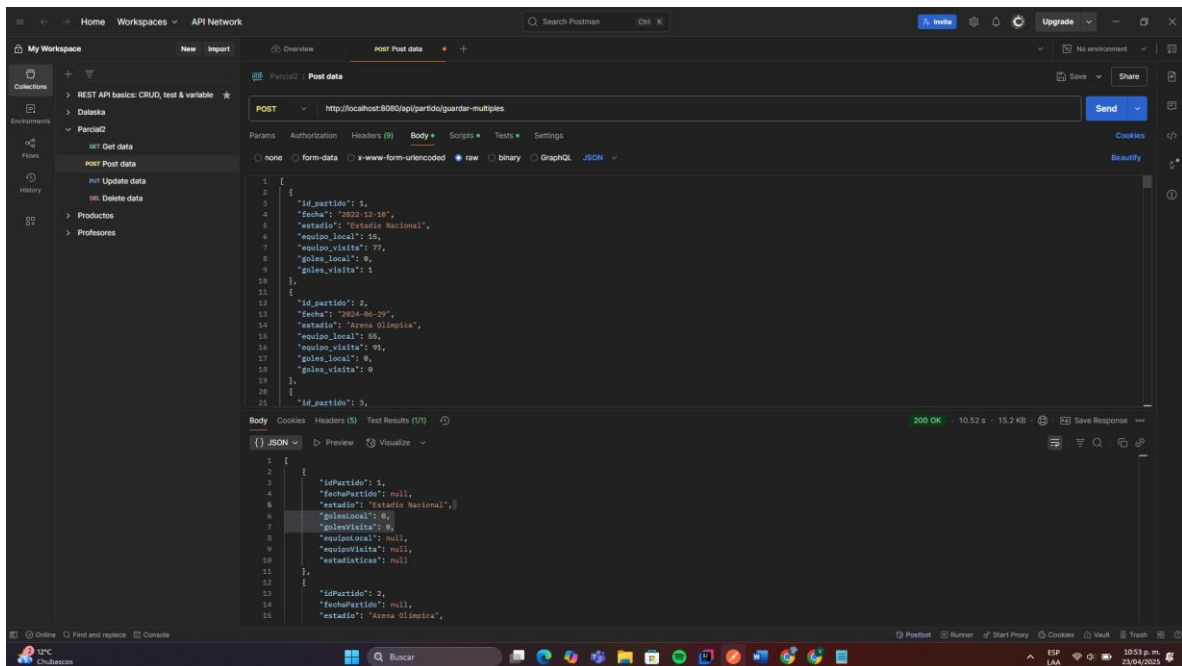
Datos Jugadores

`http://localhost:8080/api/jugador/guardar-multiples`



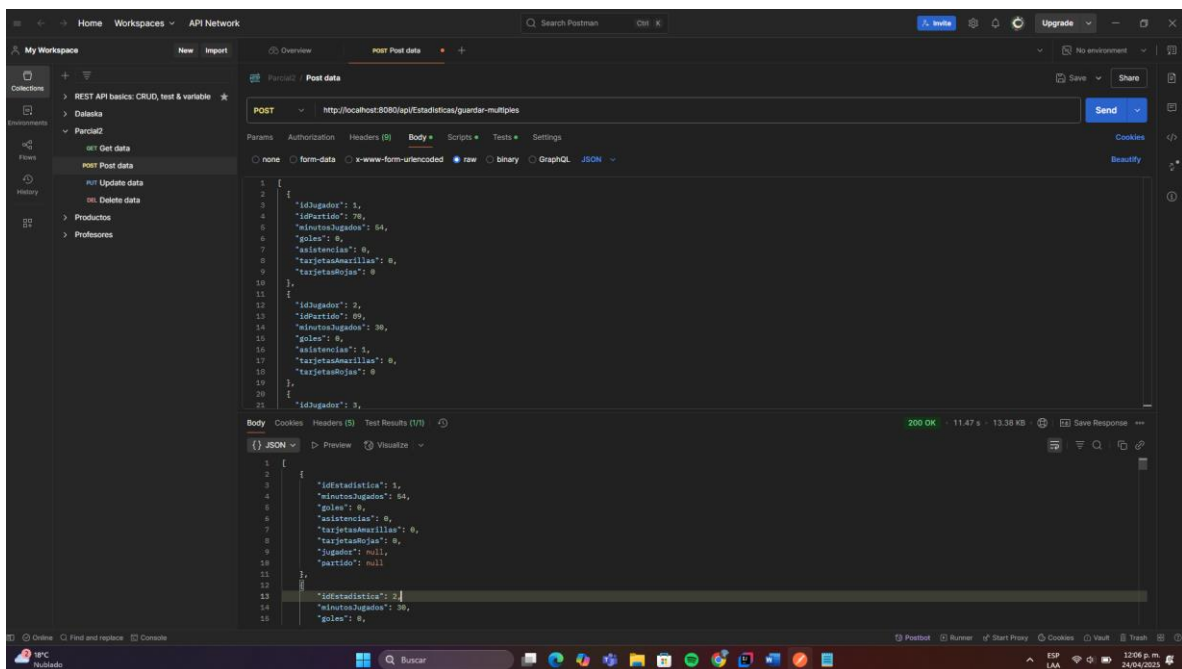
Datos Partidos

`http://localhost:8080/api/partido/guardar-multiples`



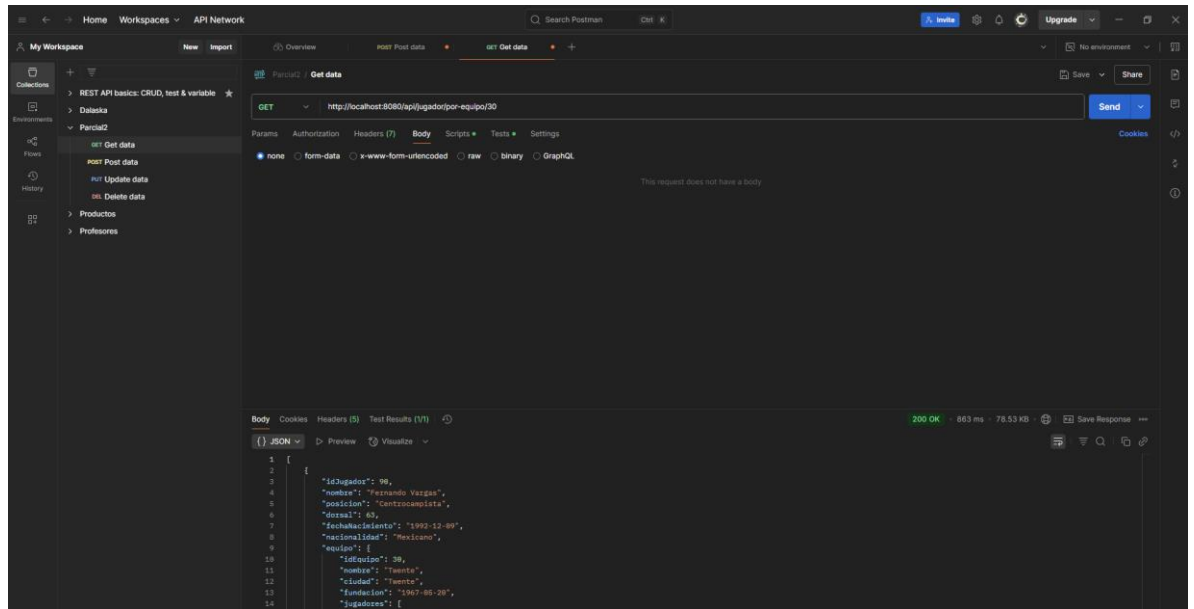
Datos Estadísticas

`http://localhost:8080/api/Estadisticas/guardar-multiples`

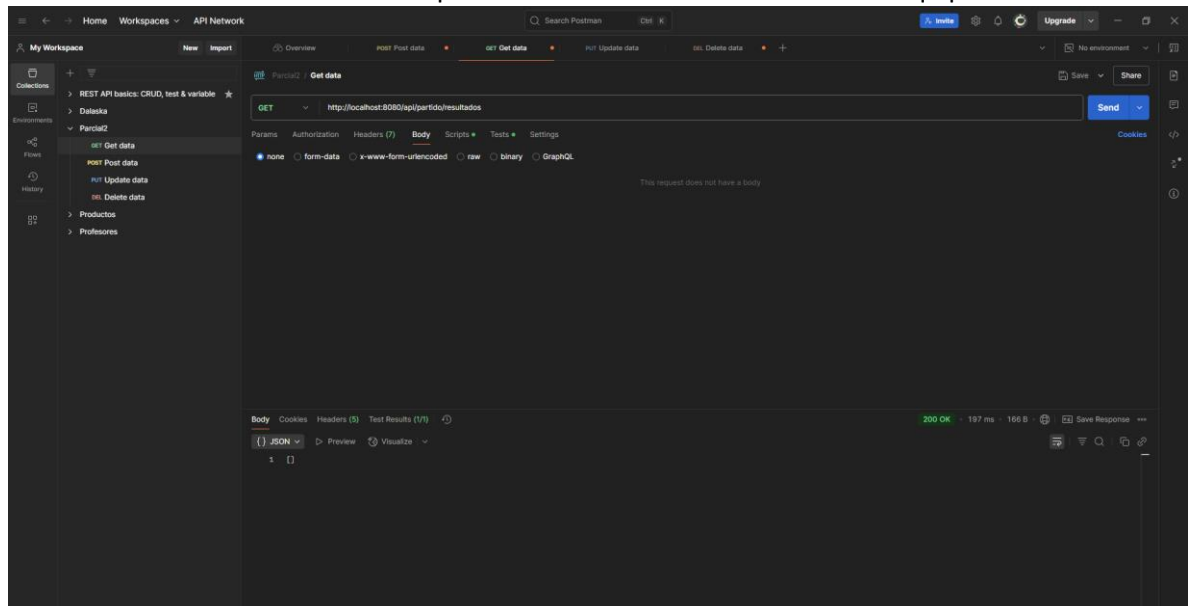


Consultas nativas

- 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



Link Github: <https://github.com/ValeriaM20350/Parcial2Corte.git>