

## **Parcial #2**

Joan Esneider Torres Gómez

Ingeniería de Sistemas, Corporación Universitaria Minuto de Dios

NRC 66167: Arquitectura de Software

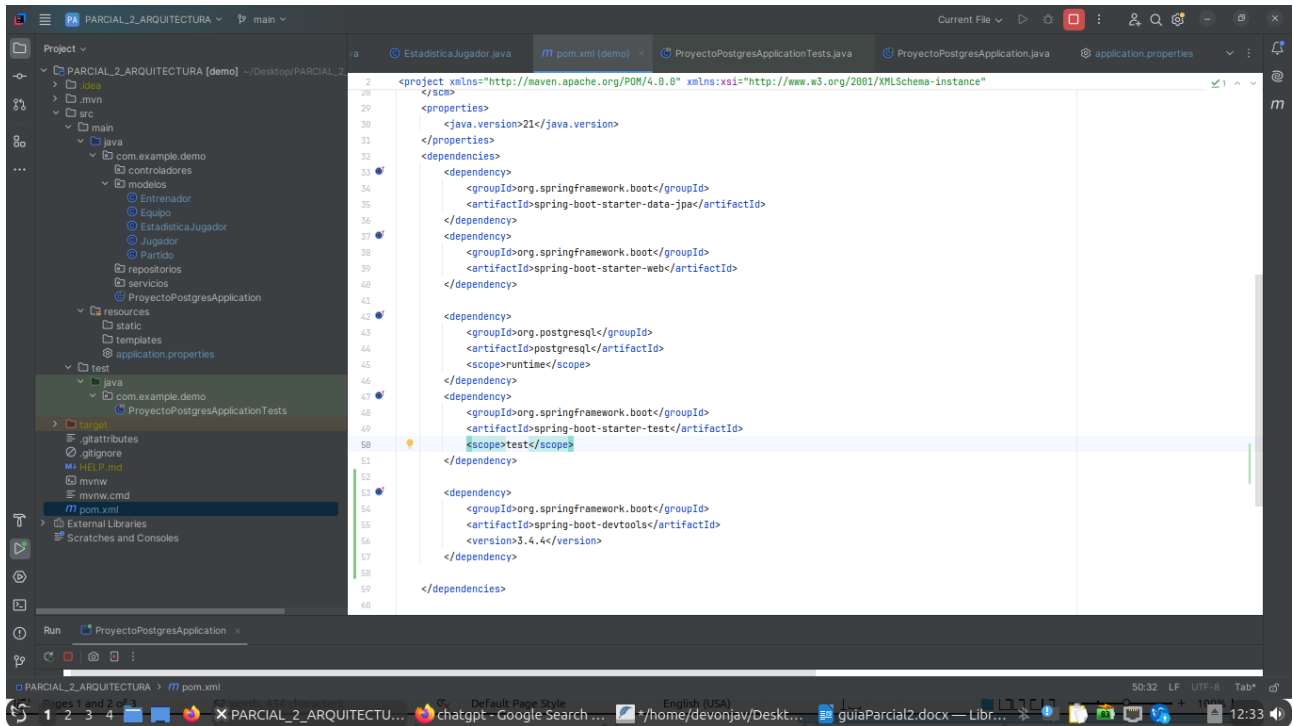
William Matallana Porras

Abril 24, 2025

## 1. Conexión con Supabase

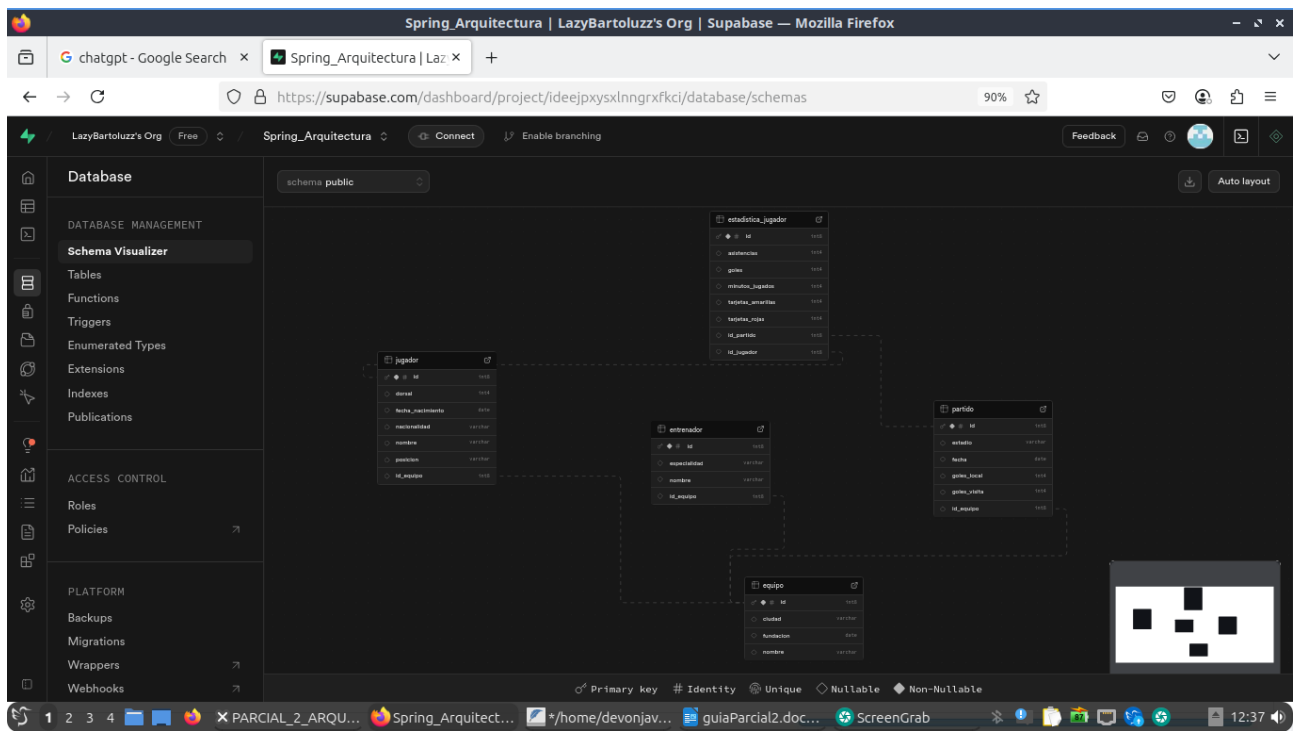
Antes que nada se necesita descargar un proyecto spring boot con las dependencias para web, devtools, jpa para poder iniciar.

Para hacer la conexión se debe como tal tener todas las dependencias al día



## 2. Crear el Modelo de Datos

### Pantallazo de modelo en Supabase



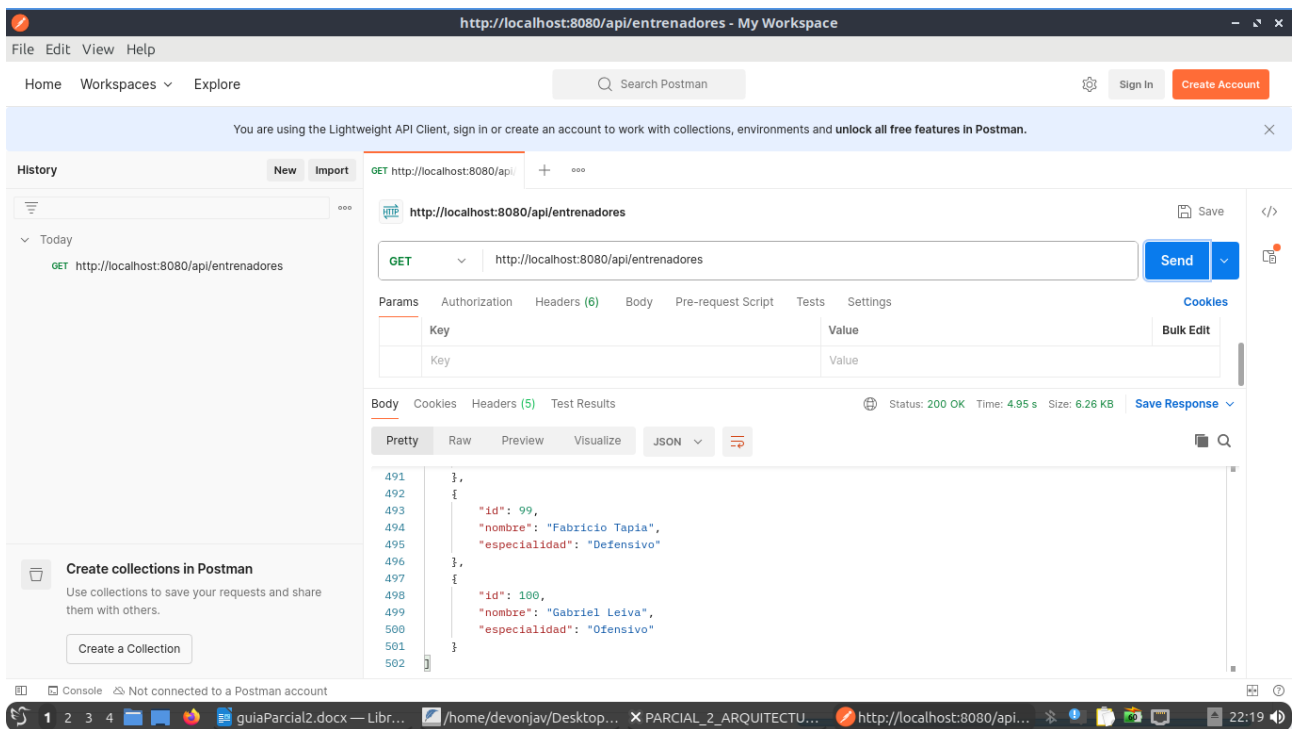
En la parte de la creación de los repositorios, servicios y controladores, si los modelos están mal desde el inicio todo lo demás en cascada también lo estará.

En este aspecto ocurre que si tenemos 2 campos para claves foráneas con el mismo nombre dado con el `@JoinColumn(elNombre)` se solapa el campo o algo así y arruina todo.

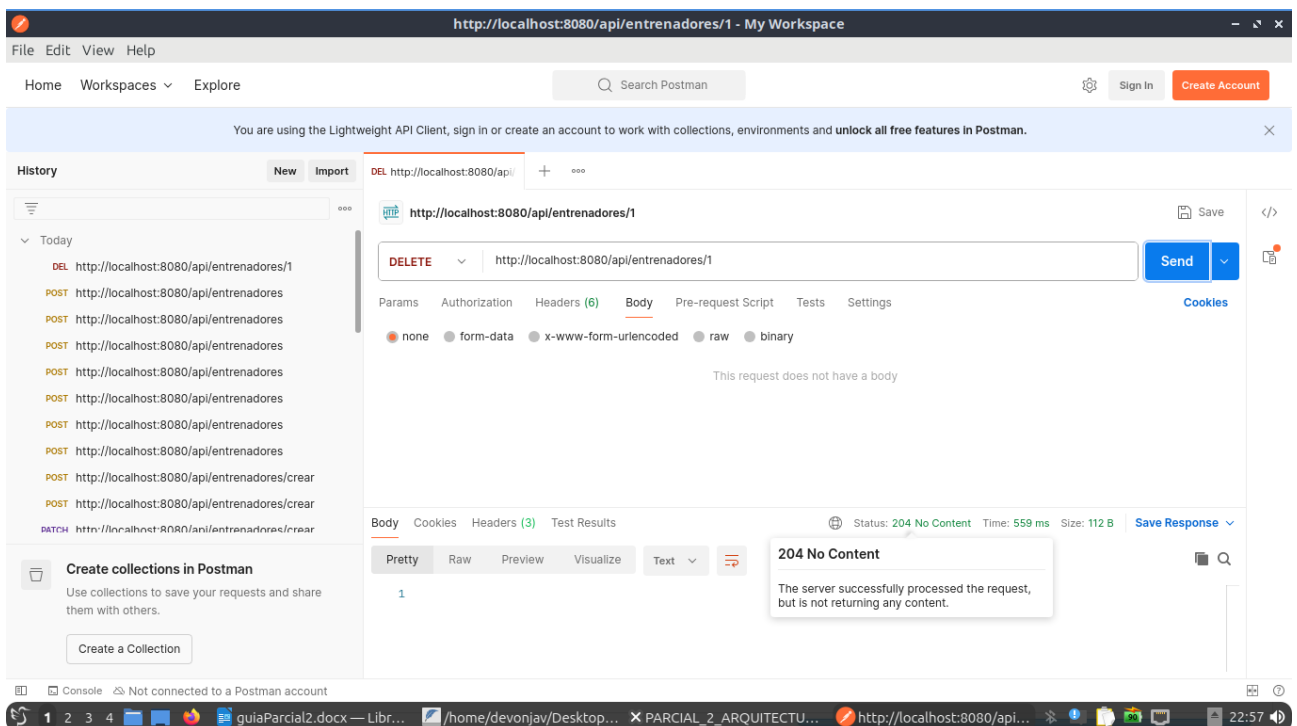
## Pantallazos

### Tabla Entrenadores

- Mostrar todos los entrenadores



- crear nuevo entrenador
- borra un entrenador con id = 1



- actualizar un entrenador id = 2

Postman interface showing a PUT request to `http://localhost:8080/api/entrenadores/2`. The request body is a JSON object:

```
{  "nombre": "Carlos Ruiz",  "especialidad": "Defensa",  "idEquipo": 1}
```

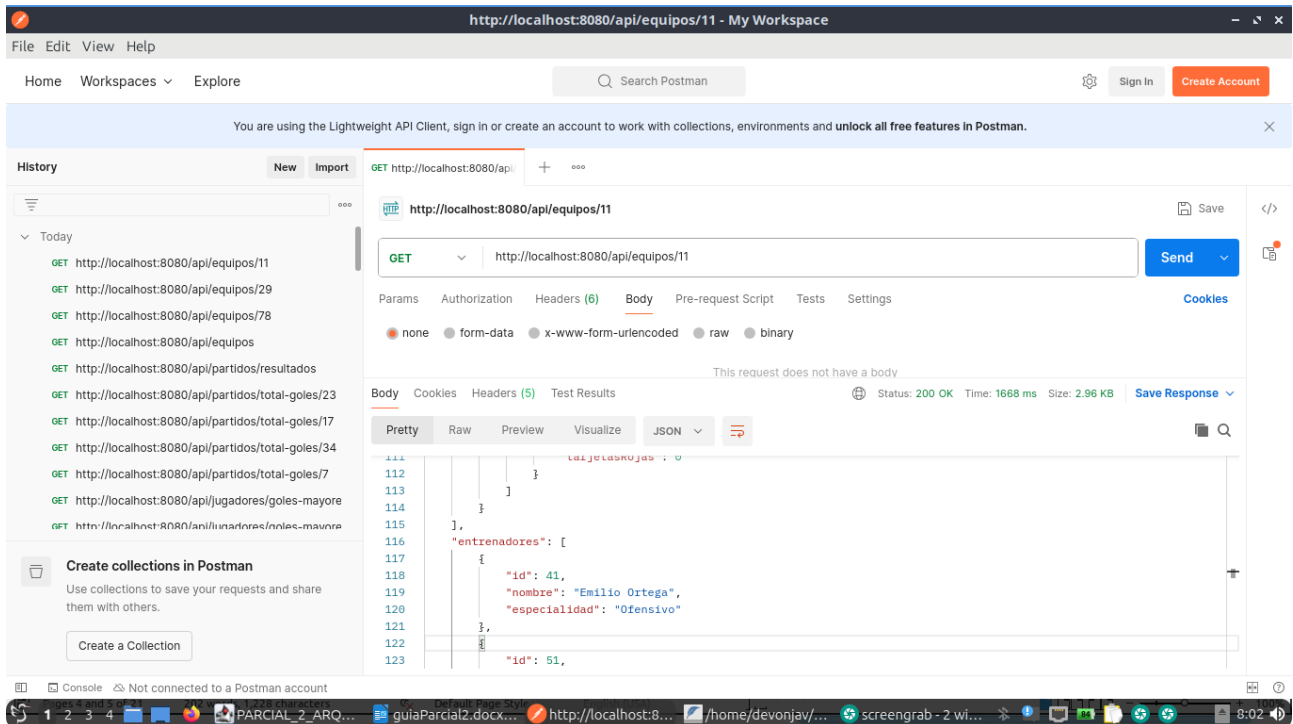
The response status is 200 OK, with a time of 1006 ms and a size of 220 B. The response body is:

```
{  "id": 2,  "nombre": "Carlos Ruiz",  "especialidad": "Defensa"}
```

The interface includes a History panel on the left, a main workspace for editing requests, and a bottom console area.

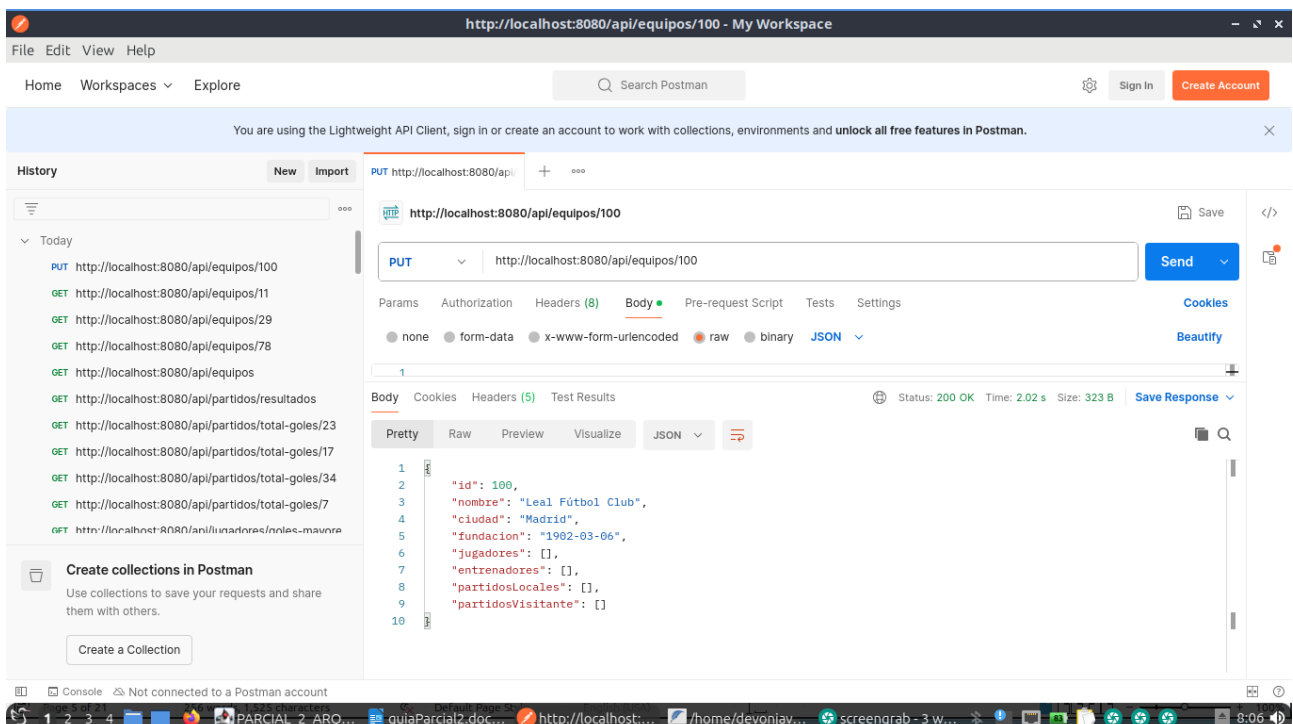
## Tabla Equipo

- Mostrar todos los equipos, son unos 100 equipos y por cada equipo debe mostrar una cantidad increíble de información debido a que no usa un objeto de transporte de info que permita limitar un poco lo que se recibe, así que hice un obtener equipo por identificador igual a 11.

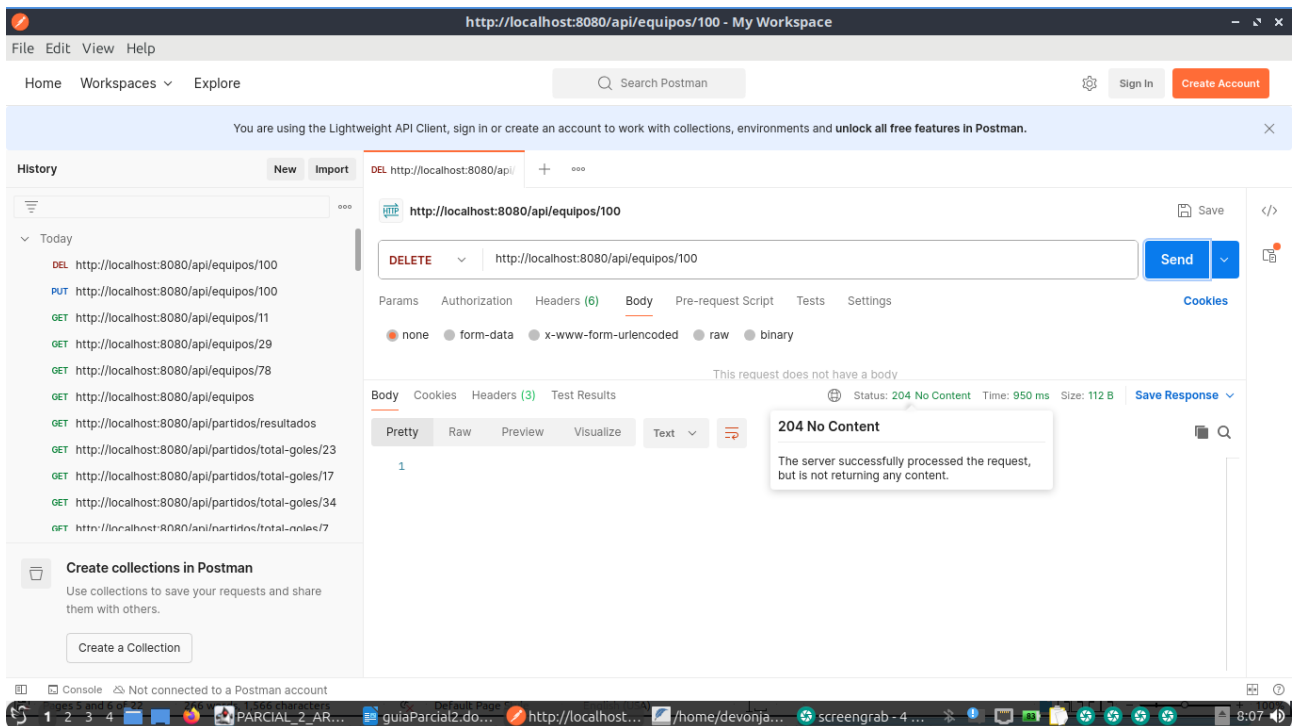


- Crear equipo

- Actualizar un equipo existente (id = 100)

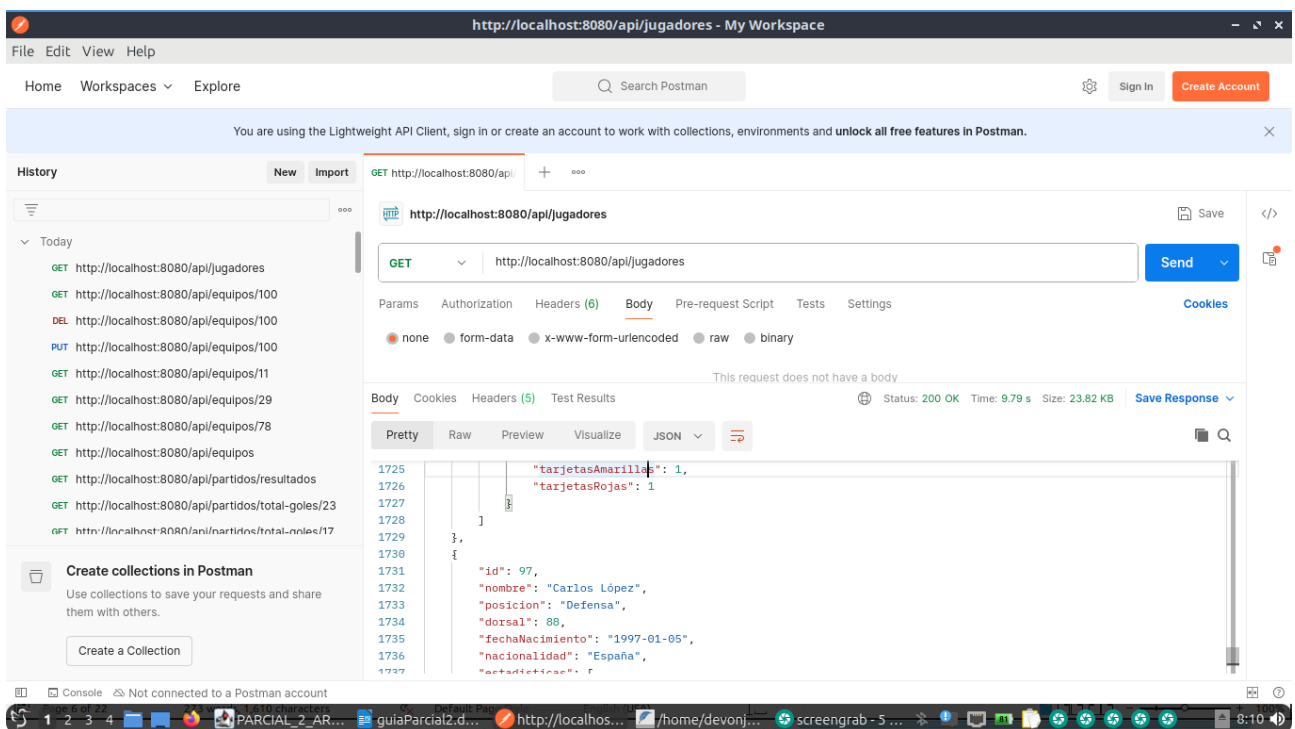


- Eliminar un equipo (id = 100)



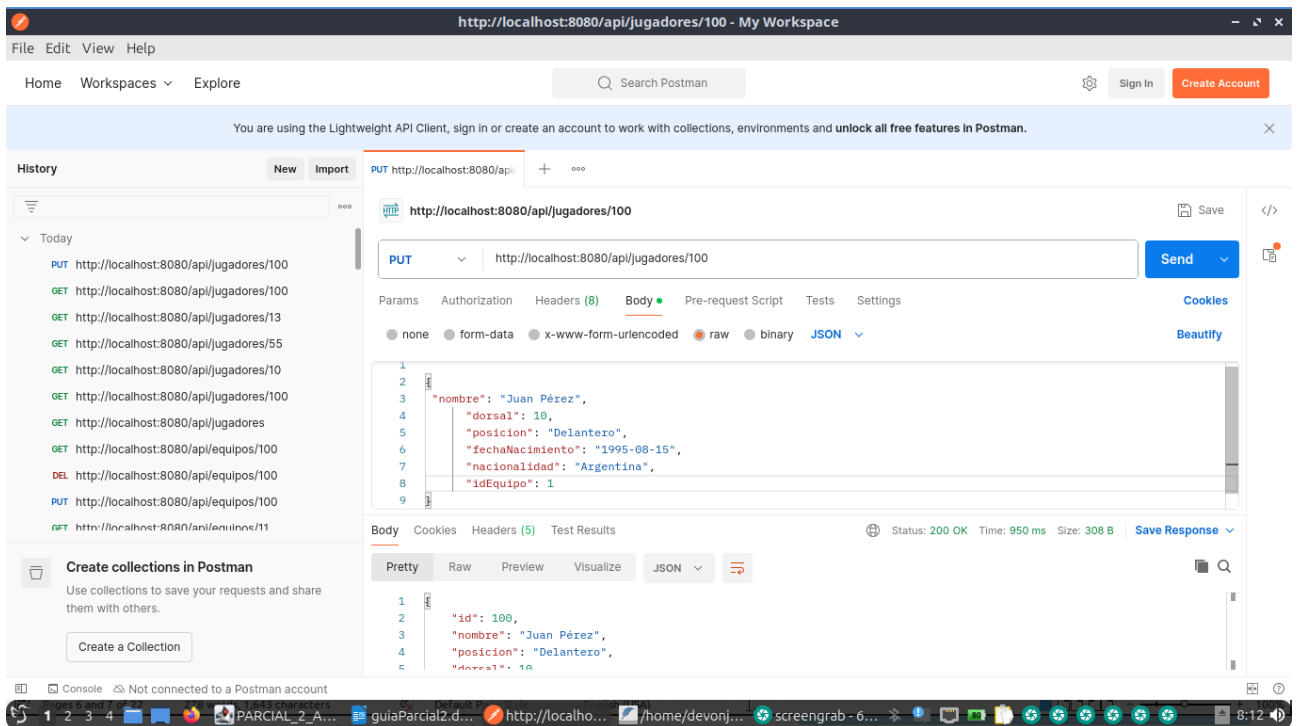
## Tabla Jugadores

- Obtener todos los jugadores

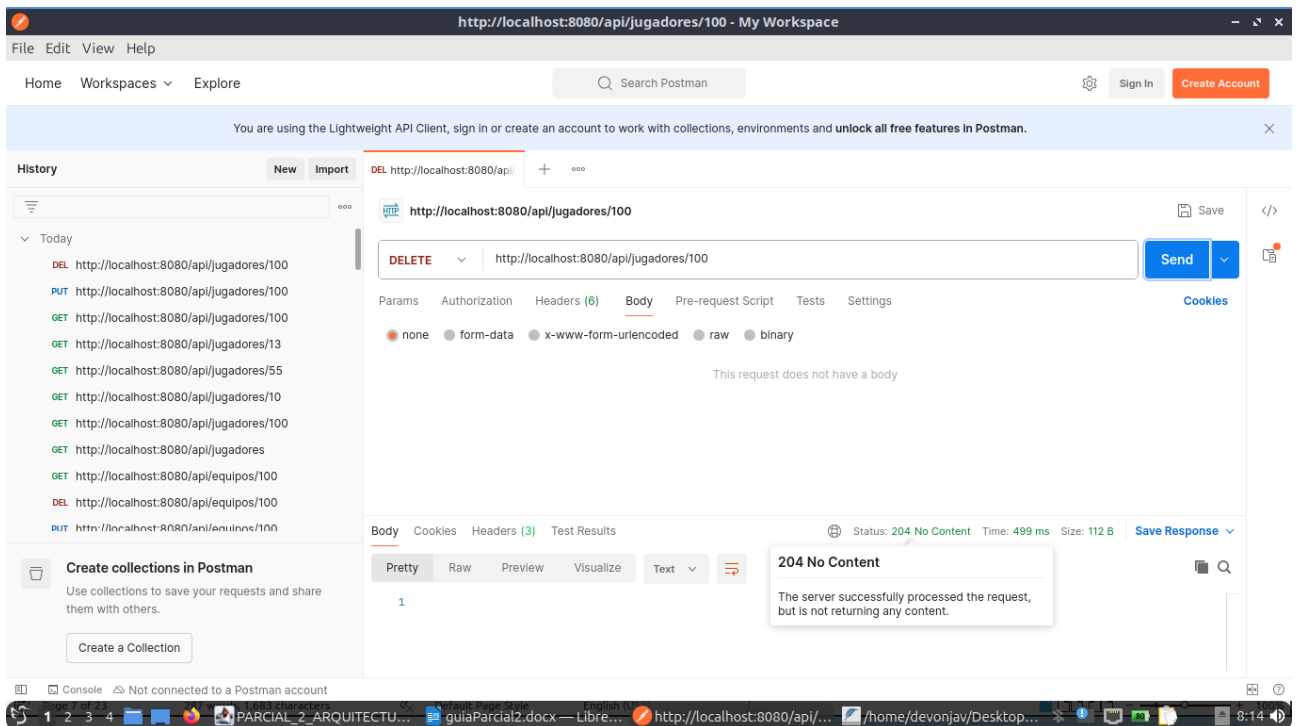


- Crear jugador

- Modificar jugador (id = 100)



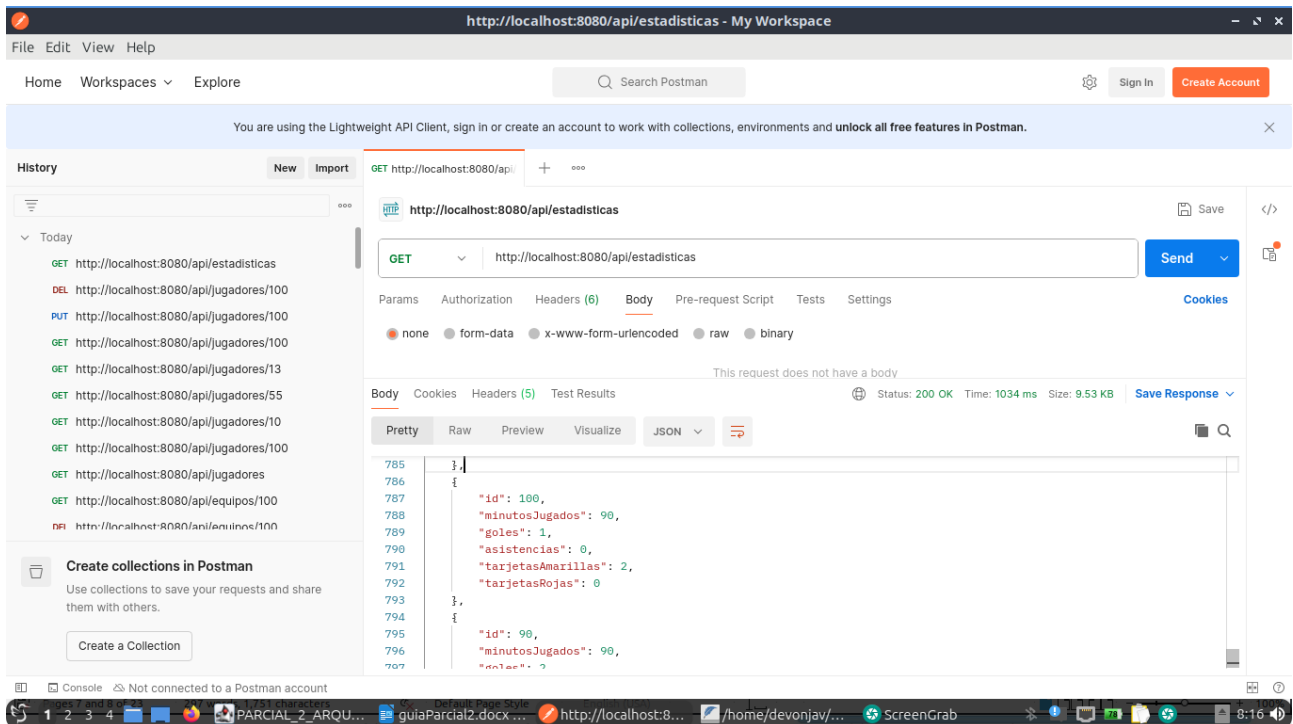
- Eliminar jugador (id = 100)



## Estadística Jugador

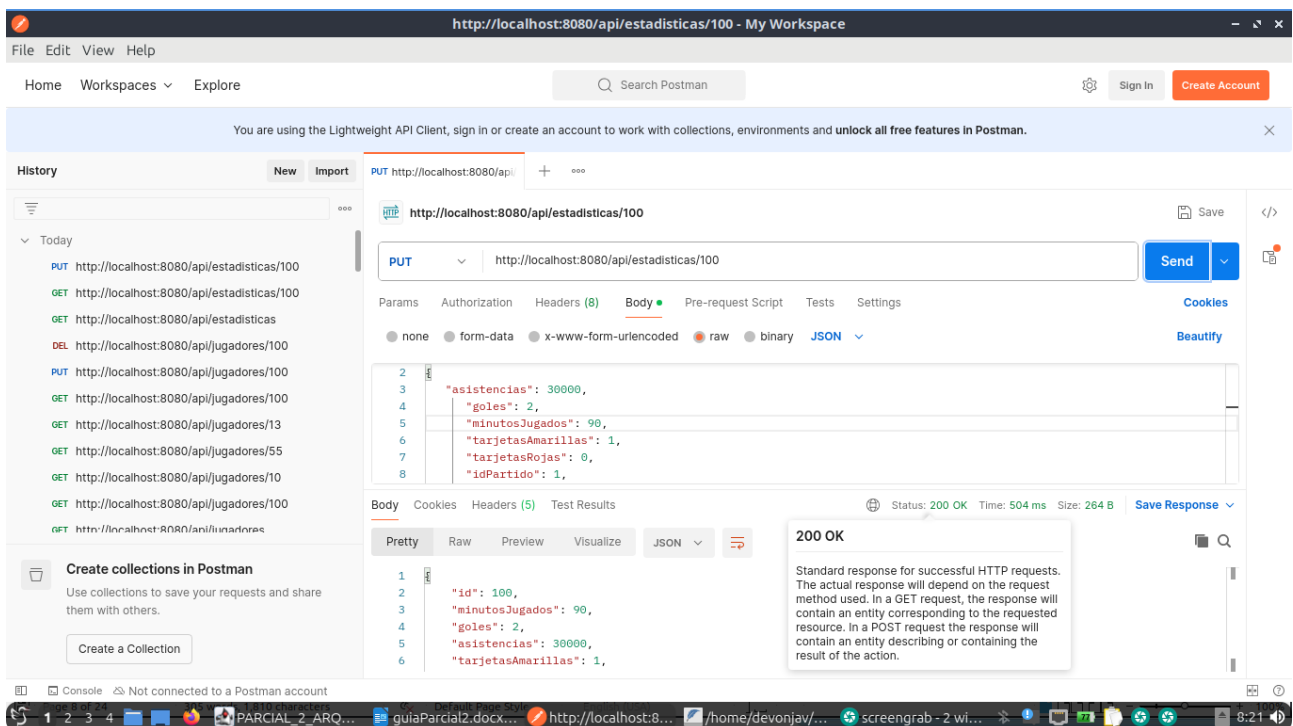


## - Obtener todas las estadísticas de los jugadores

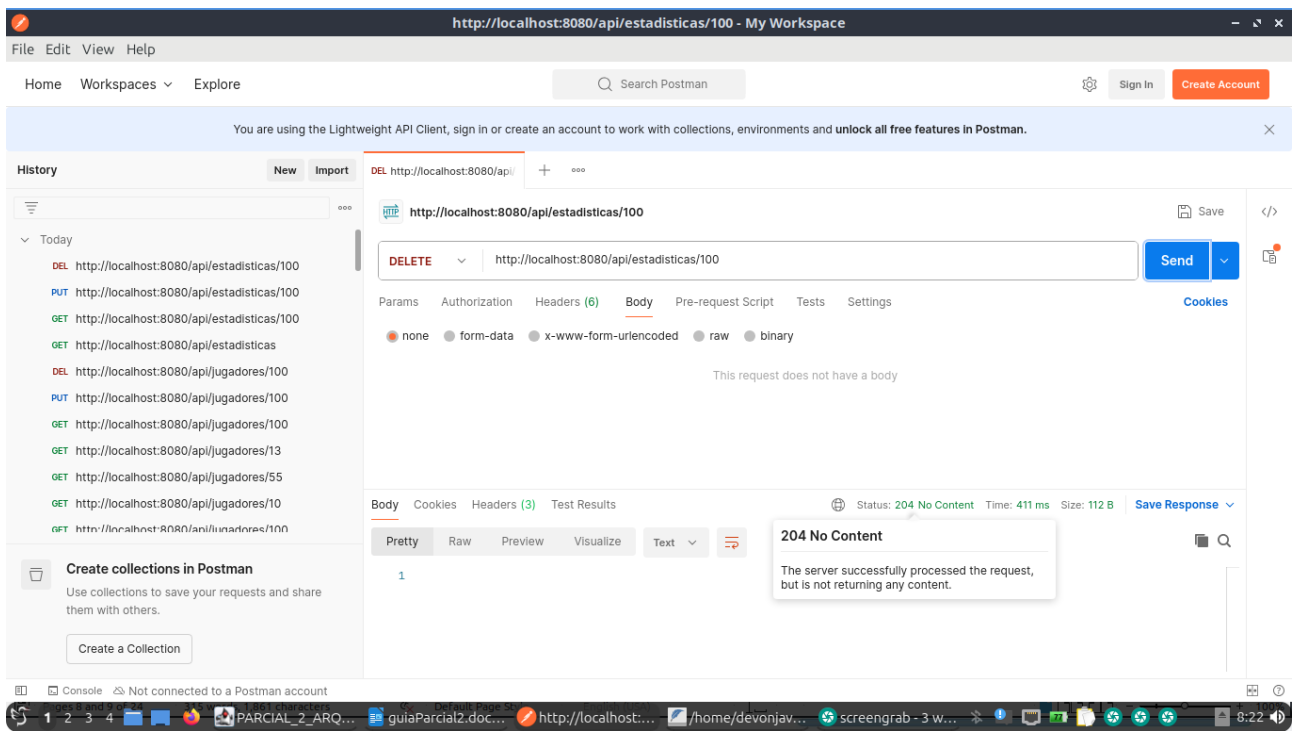


## - Crear estadística jugador

## - Actualizar estadística jugador (id = 100)

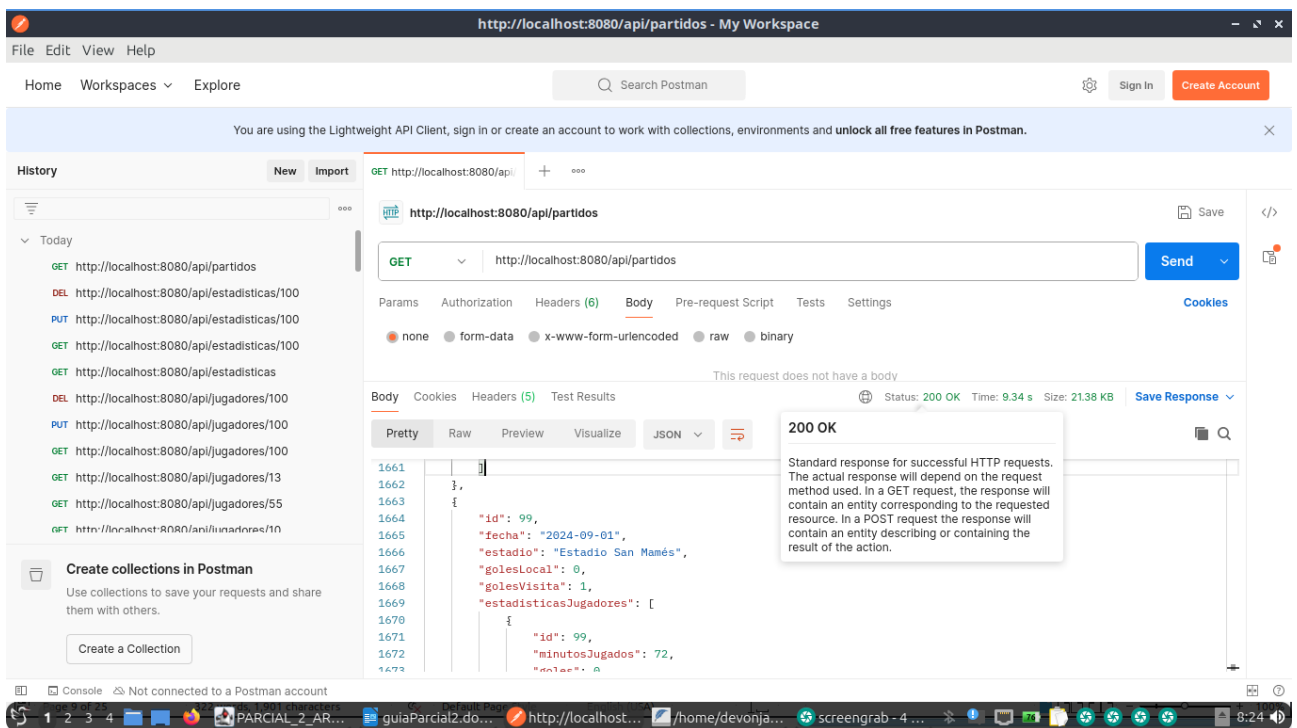


## - Borrar estadística jugador (id = 100)



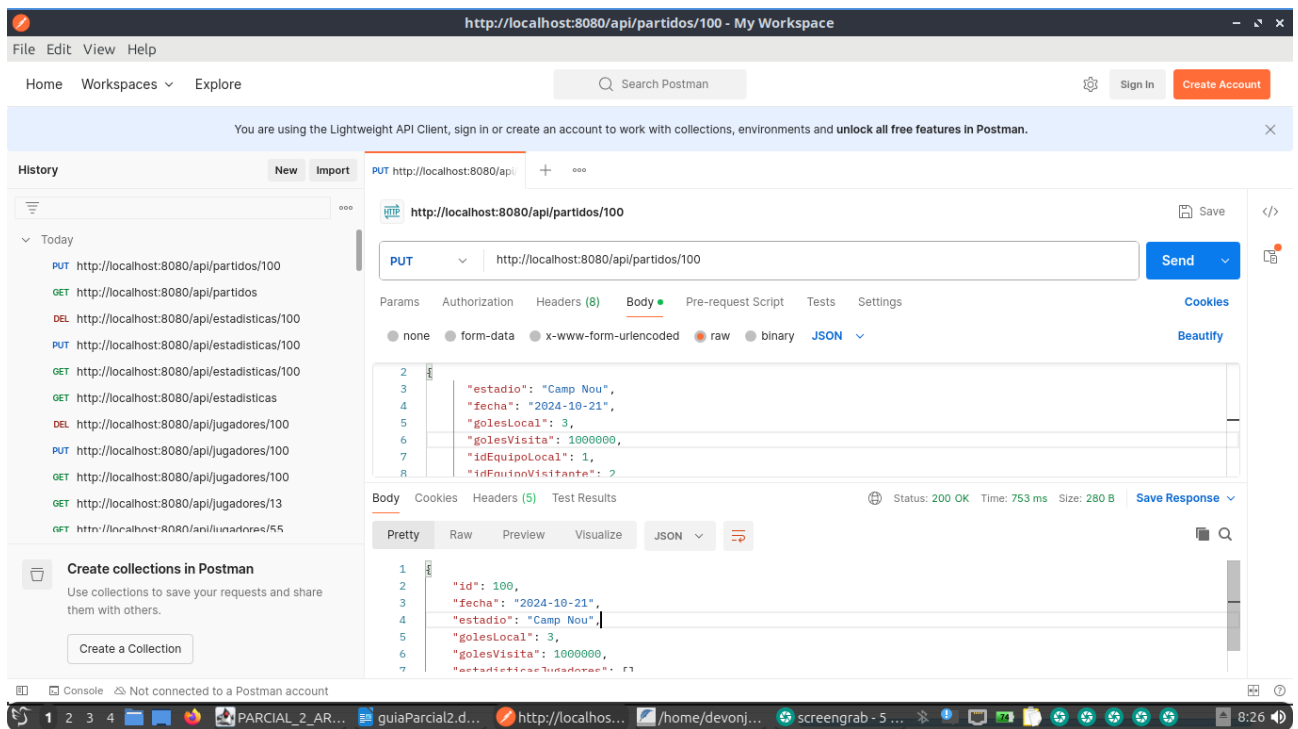
## Tabla Equipo

- Obtener todos los partidos

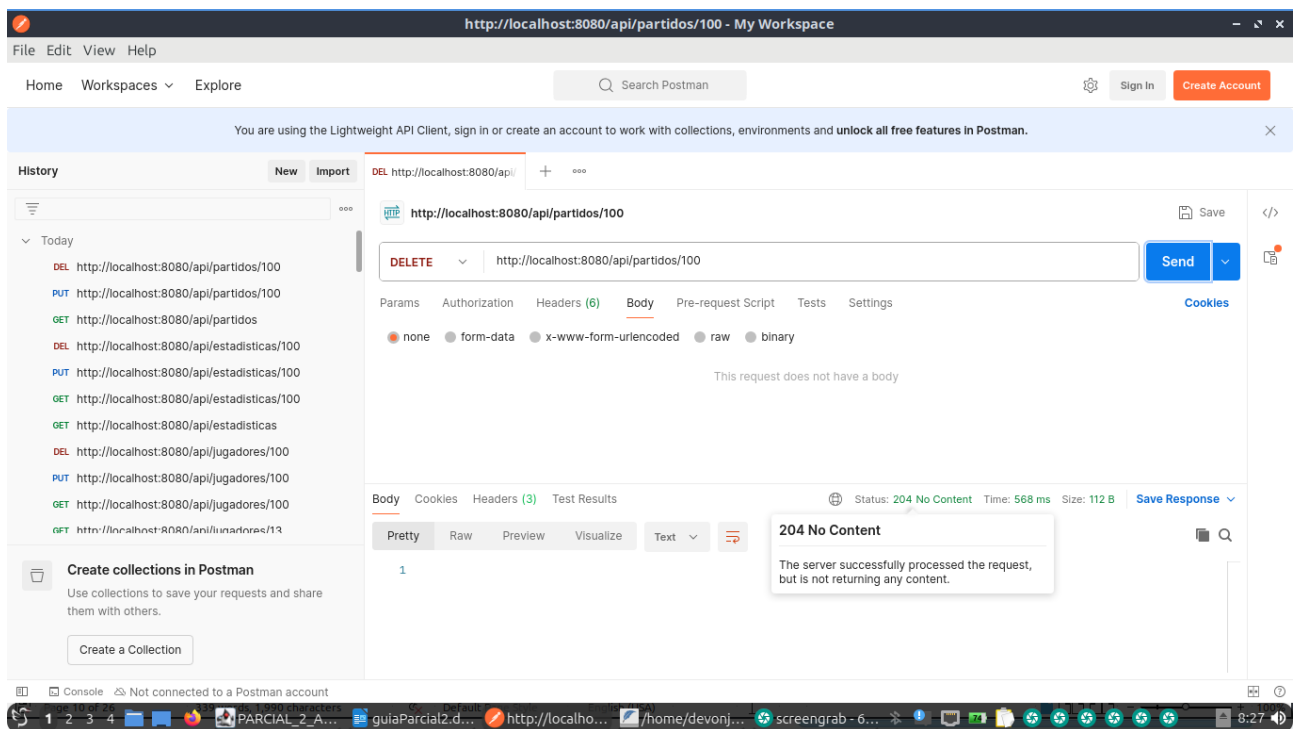


- Crear nuevo partido

- Actualizar partido existente (id = 100)

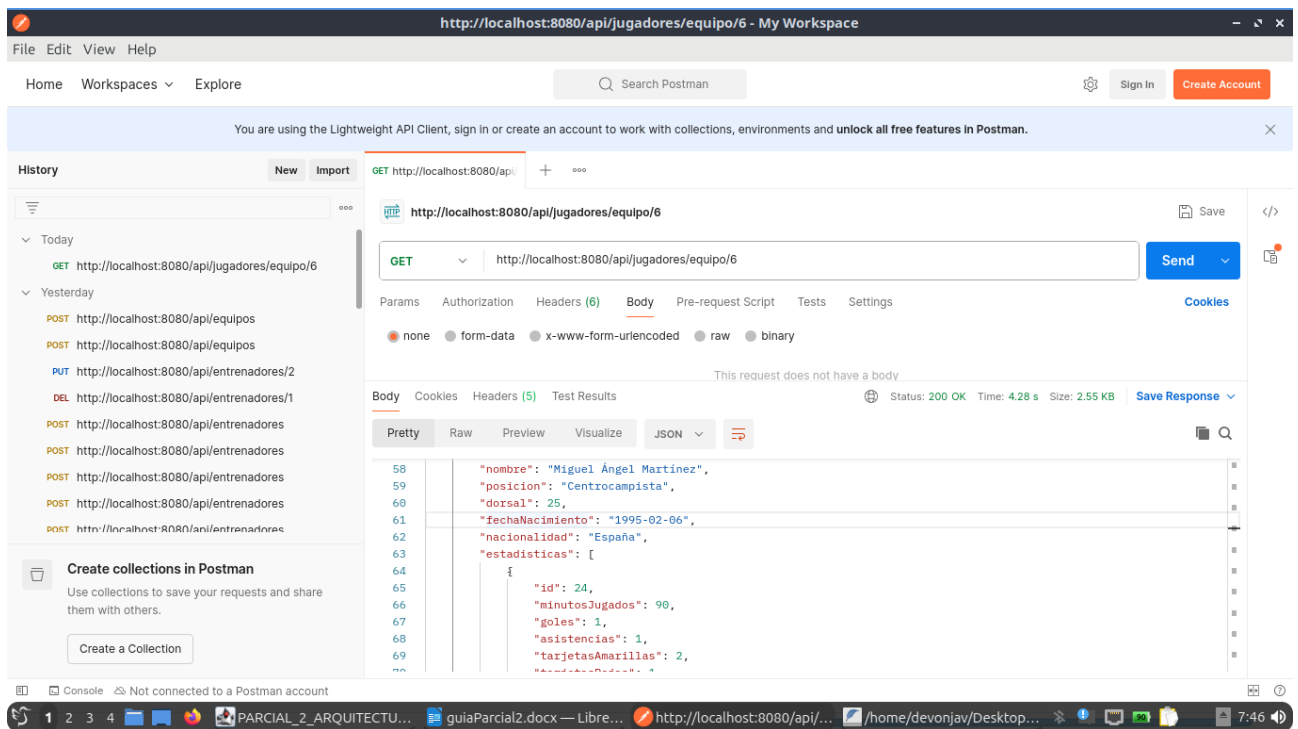


- Borrar partido (id = 100)

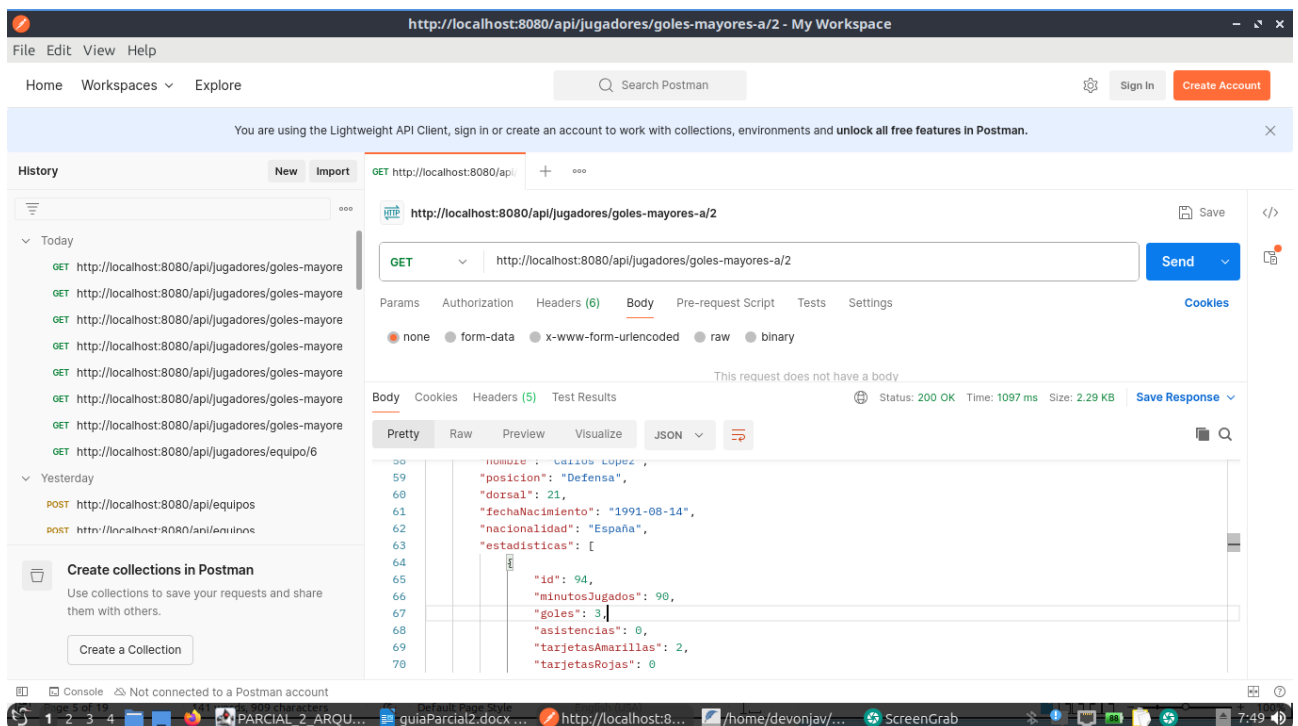


## Consultas Nativas

1. Todos los jugadores que pertenecen al equipo con ese ID (6)

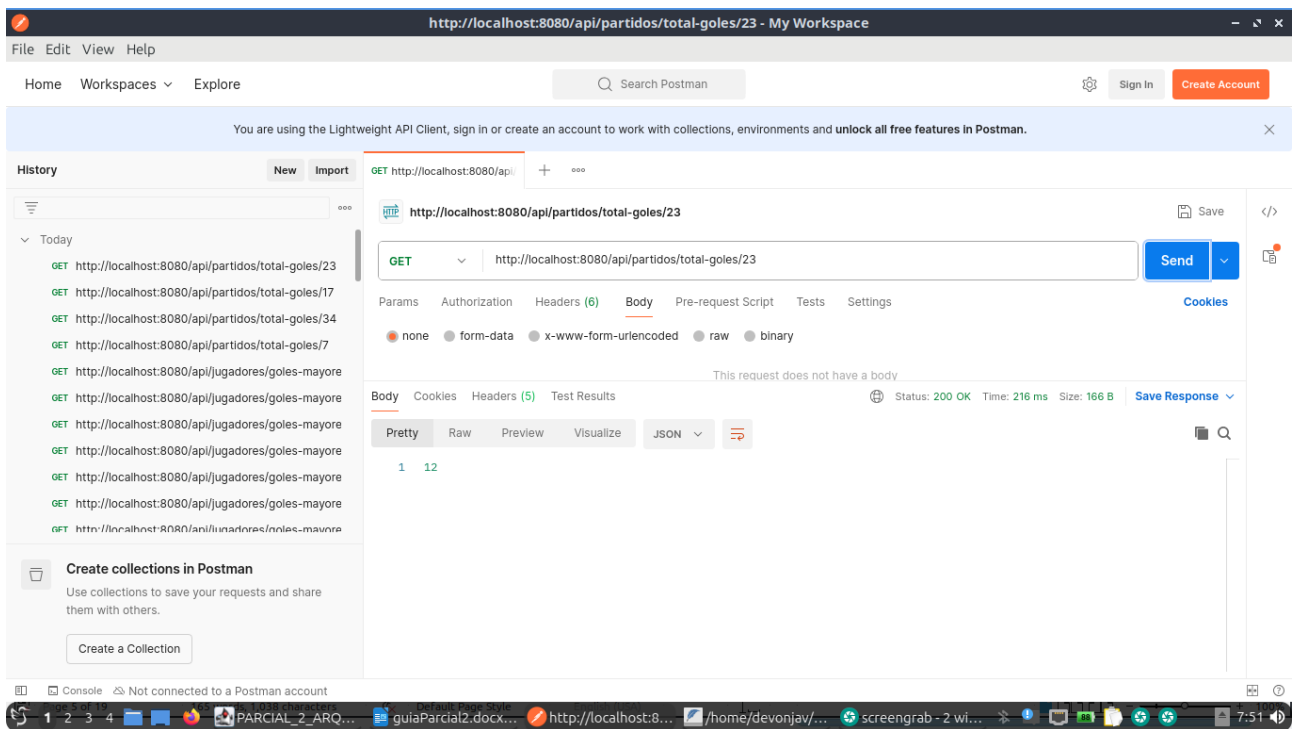


## 2. Jugadores que tienen más de 2 goles acumulados

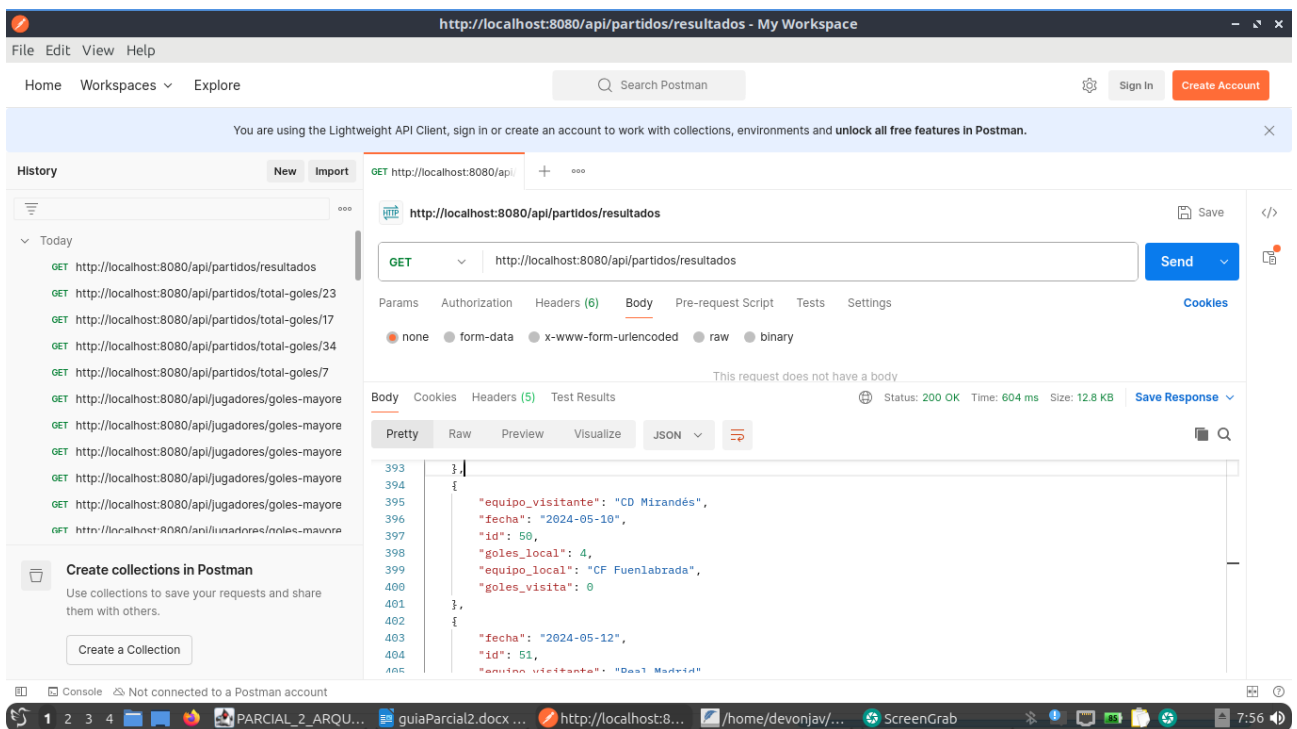


No habian jugadores que tuvieran mucho mas de 10 goles por ejemplo ni tampoco 5.

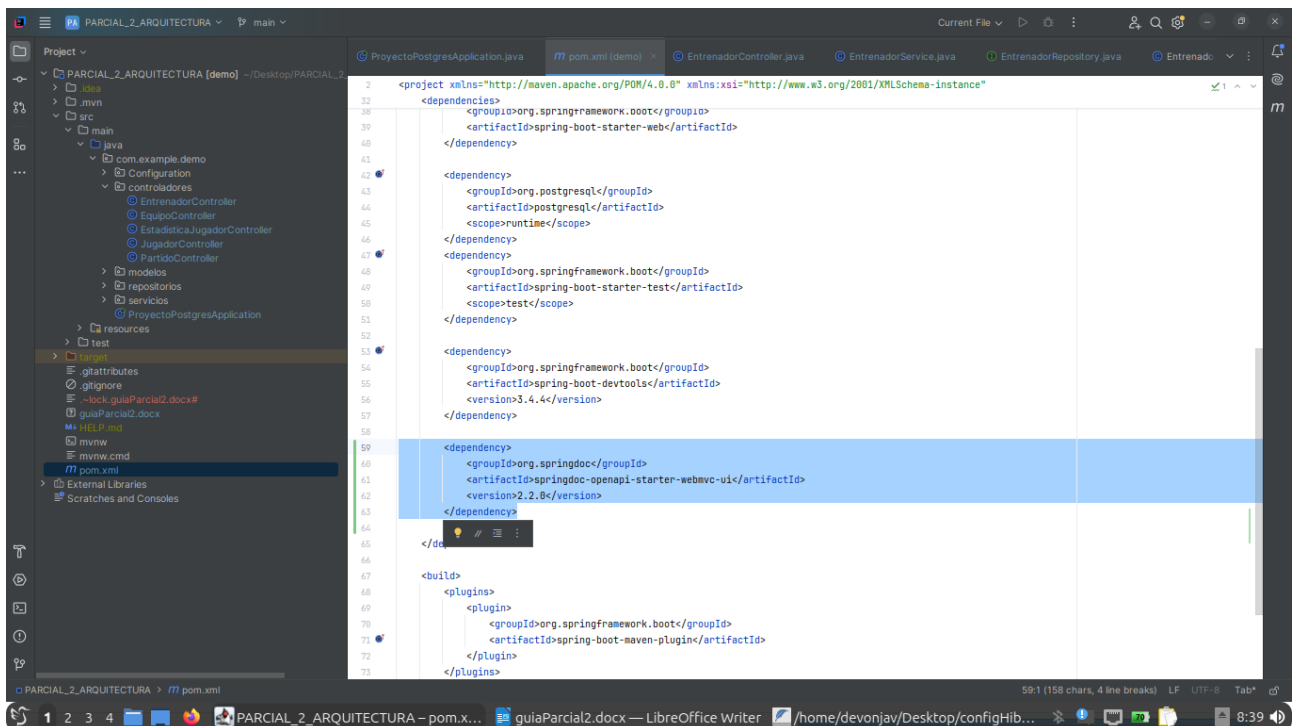
## 3. El total de goles de un equipo marcados en todos sus partidos (id = 23)



#### 4. Los resultados con nombres de equipo local y visitante, goles, y fecha



#### 7. Swagger



Tenemos que usar la dependencia de springdoc openapi starter webmvc-ui para poder hacer la documentación.

```
<!-- https://mvnrepository.com/artifact/org.springdoc/springdoc-openapi-starter-webmvc-ui -->
```

```
<dependency>
```

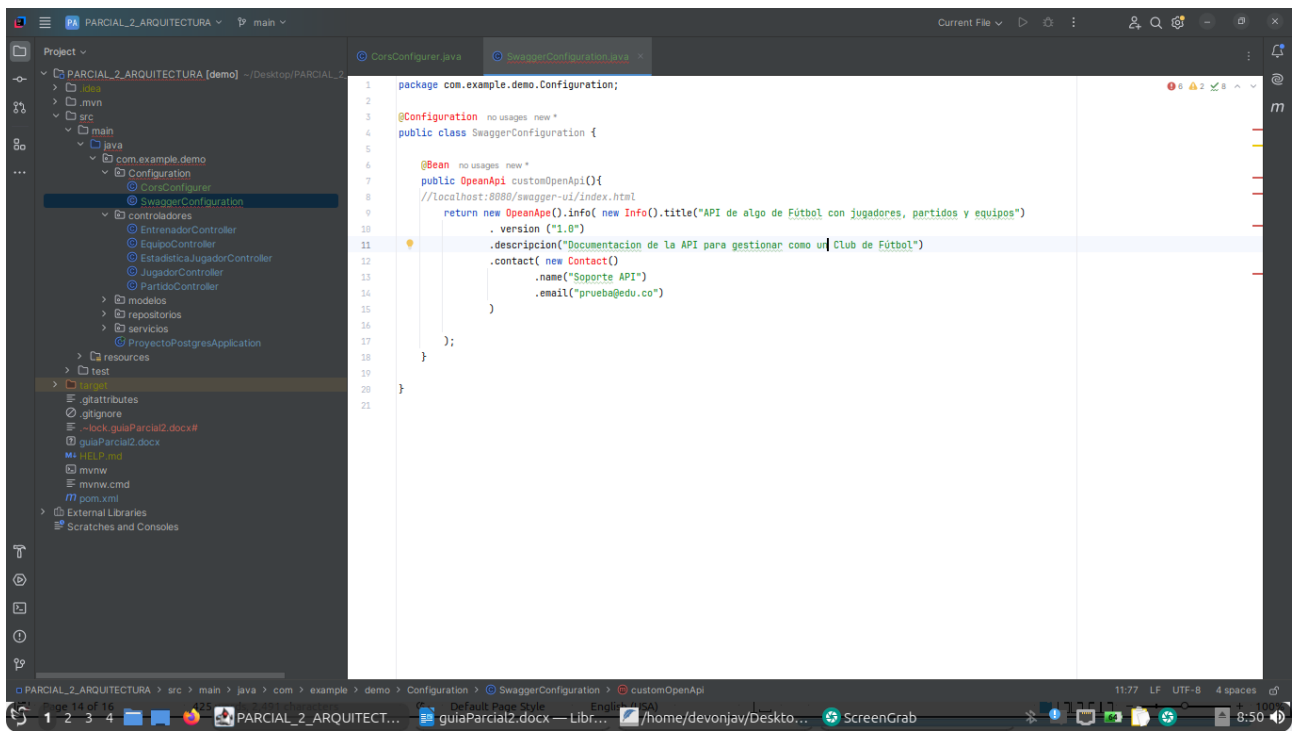
```
    <groupId>org.springdoc</groupId>
```

```
    <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
```

```
    <version>2.2.0</version>
```

```
</dependency>
```

Luego de eso creamos como se ve en la imagen una paquete llamado Configuration en donde creamos 2 clases.



En la primera se escribe eso y en la otra se escribe algo así





## **Bibliografía**

(JPA query methods, s/f)

JPA query methods. (s/f). Spring.io. Recuperado el 22 de abril de 2025, de

<https://docs.spring.io/spring-data/jpa/reference/jpa/query-methods.html>

*Postman documentation overview* | *Postman Docs*. (2025, 7 febrero). Postman Docs.

<https://learning.postman.com/docs/introduction/overview/>

*Spring boot* :: *Spring boot*. (s. f.). <https://docs.spring.io/spring-boot/index.html>