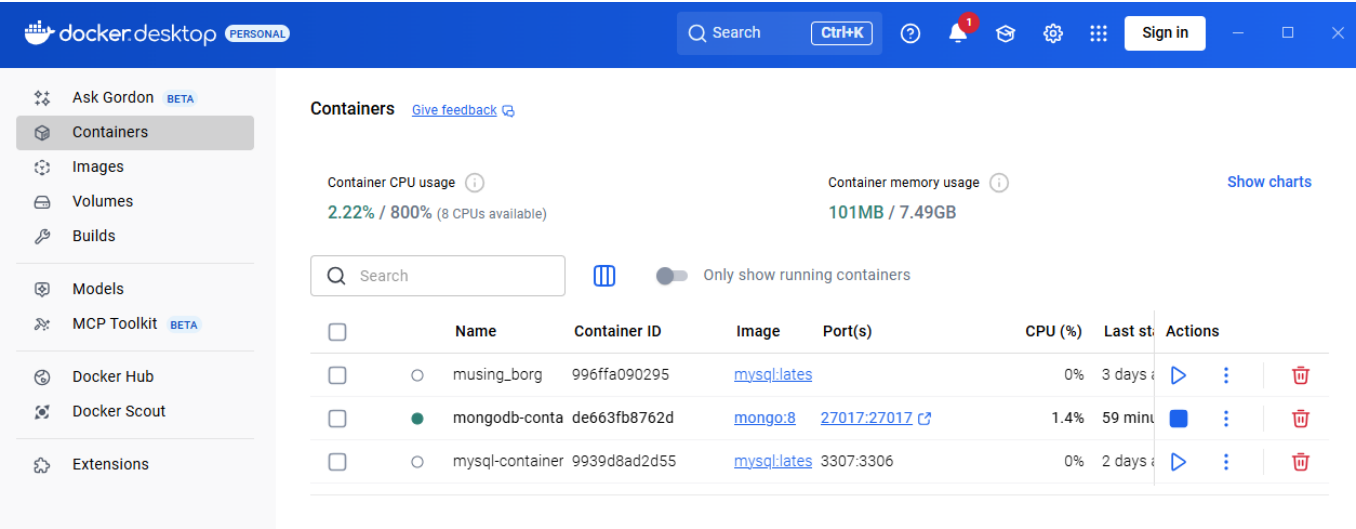
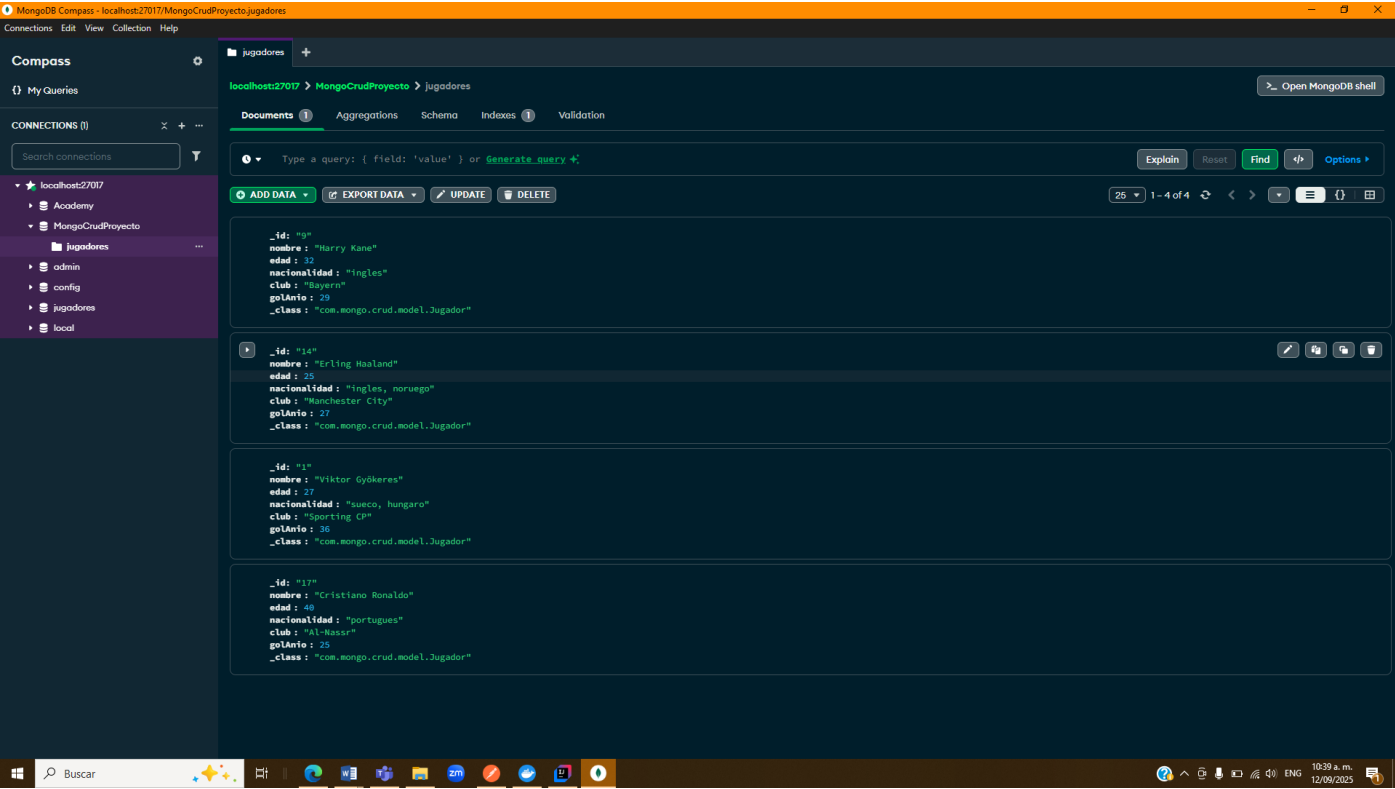


# MONGO CRUD

La base de datos usada para la conexión se llama MongoCrudProyecto, con una conexión a un Contenedor en Docker llamado mongodb-container por el puerto 27017.

La colección que se creó al unir la aplicación Spring con MongoDB Compass se llama jugadores, haciendo referencia a la clase Jugador.

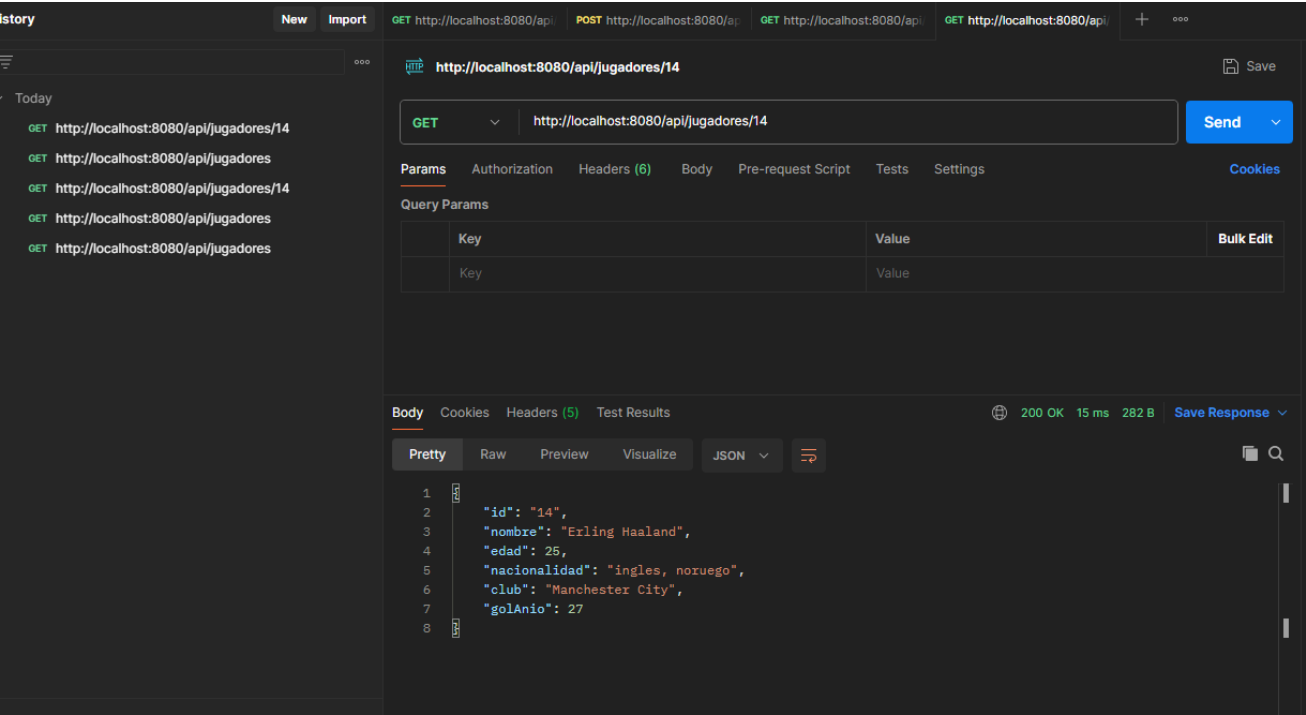
Tomcat funciona utilizando el puerto 8080 (http) en localhost.



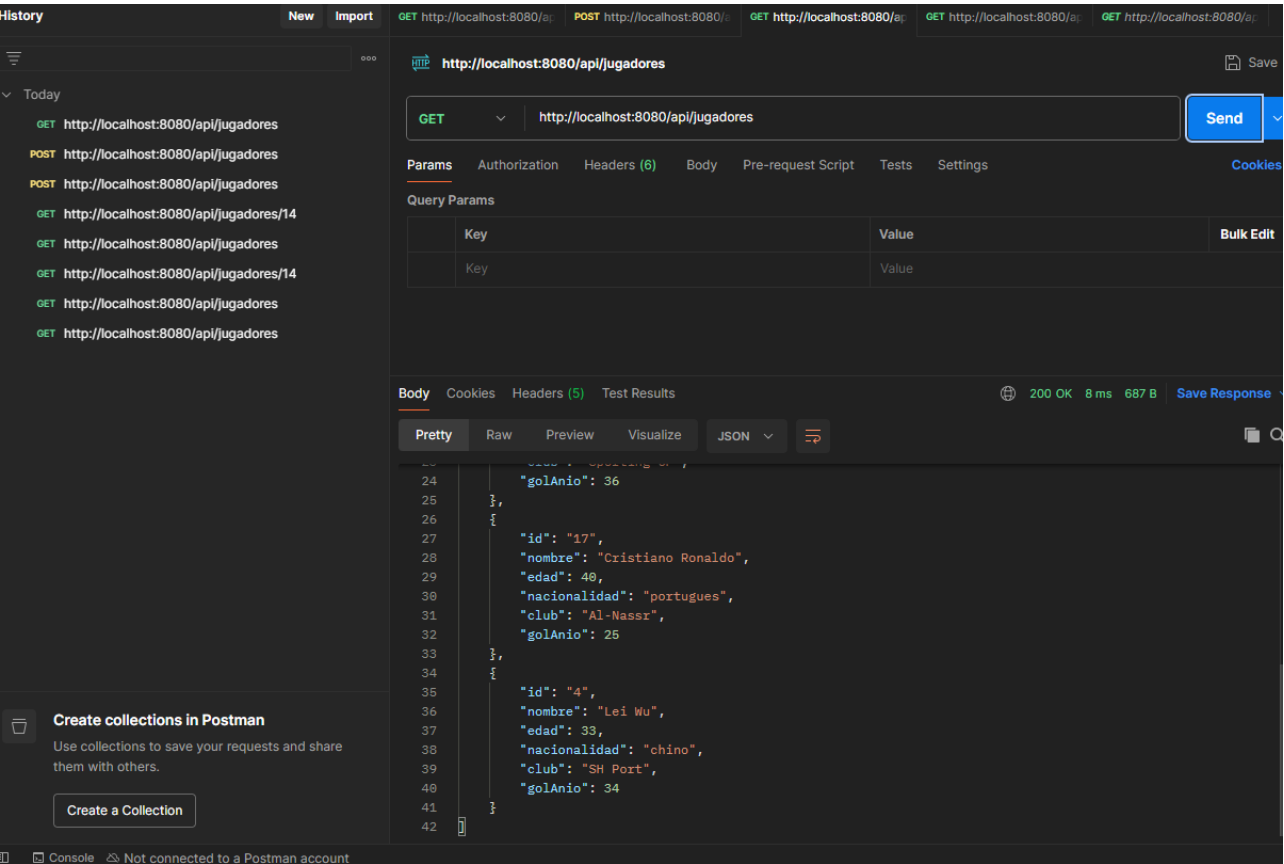
# POSTMAN

Todas las funciones CRUD trabajan con codigo de respuesta 200.

- GET Jugador por Id



- GET all jugadores



Yalbani Aranda

- POST

History

NewImport

GET http://localhost:8080/api/POST http://localhost:8080/api/GET http://localhost:8080/api/GET http://localhost:8080/api/+...

Today

POST http://localhost:8080/api/jugadores

GET http://localhost:8080/api/jugadores/14

GET http://localhost:8080/api/jugadores

GET http://localhost:8080/api/jugadores/14

GET http://localhost:8080/api/jugadores

GET http://localhost:8080/api/jugadores

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

http://localhost:8080/api/jugadores

Save

POSThttp://localhost:8080/api/jugadores

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookiesBeautify

noneform-datax-www-form-urlencoderauwbinaryJSON

```
1{"id": "17",
2  "nombre": "Cristiano Ronaldo",
3  "edad": 40,
4  "nacionalidad": "portugues",
5  "club": "Al-Nassr",
6  "golAnio": 25
7}
8
```

BodyCookiesHeaders (5)Test Results200 OK20 ms272 BSave Response

PrettyRawPreviewVisualizeJSON

```
1{"id": "17",
2  "nombre": "Cristiano Ronaldo",
3  "edad": 40,
4  "nacionalidad": "portugues",
5  "club": "Al-Nassr",
6  "golAnio": 25
7}
8
```

- DELETE

History

NewImport

GET http://localhost:8080/api/POST http://localhost:8080/api/GET http://localhost:8080/api/GET http://localhost:8080/api/DEL http://localhost:8080/api/+...

Today

DEL http://localhost:8080/api/jugadores/4

GET http://localhost:8080/api/jugadores

POST http://localhost:8080/api/jugadores

POST http://localhost:8080/api/jugadores

GET http://localhost:8080/api/jugadores/14

GET http://localhost:8080/api/jugadores

GET http://localhost:8080/api/jugadores/14

GET http://localhost:8080/api/jugadores

GET http://localhost:8080/api/jugadores

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

http://localhost:8080/api/jugadores/14

Save

DELETEhttp://localhost:8080/api/jugadores/4

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

Key	Value	Bulk Edit
Key	Value	

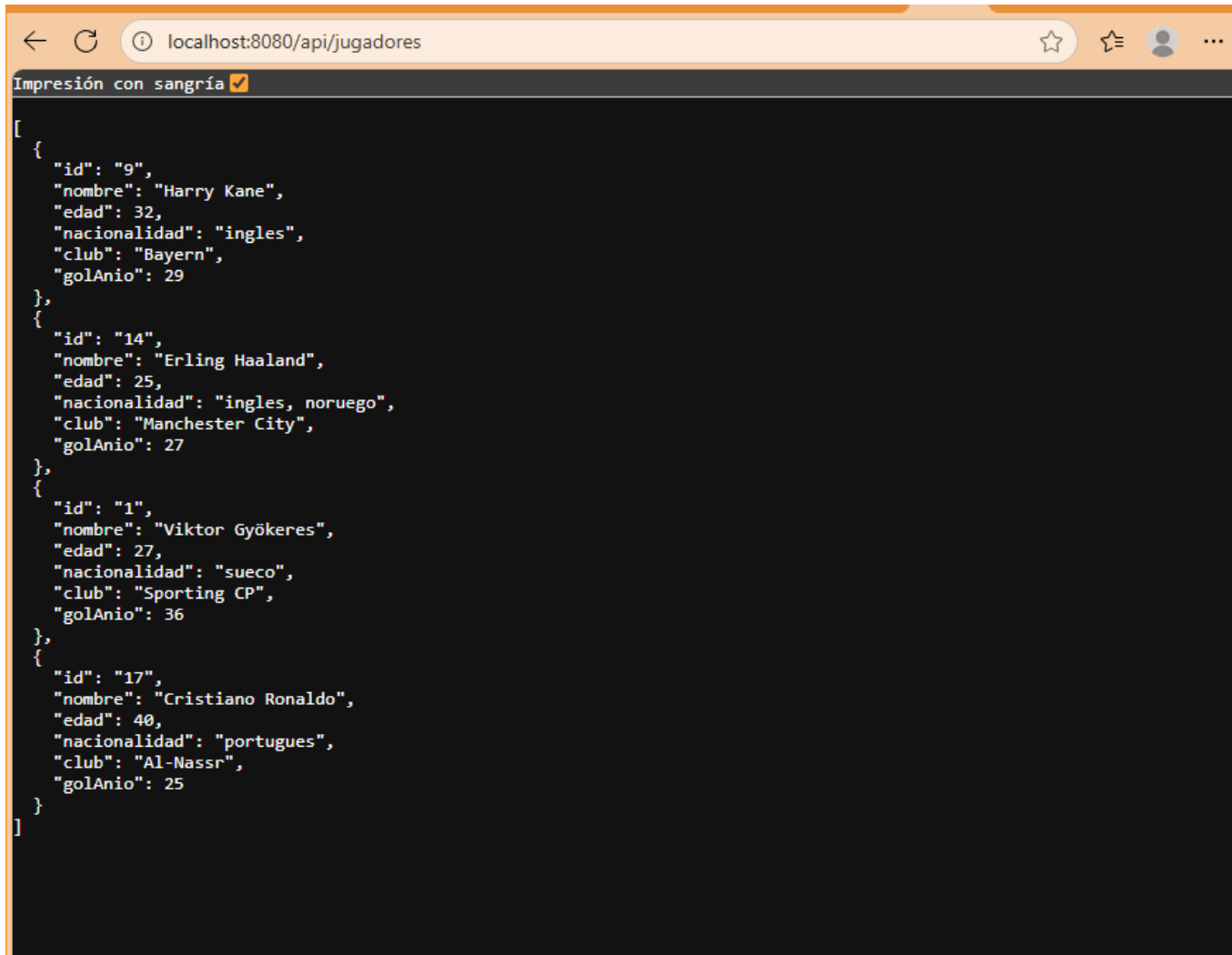
BodyCookiesHeaders (5)Test Results200 OK31 ms255 BSave Response

PrettyRawPreviewVisualizeJSON

```
1{"id": "4",
2  "nombre": "Lei Wu",
3  "edad": 33,
4  "nacionalidad": "chino",
5  "club": "SH Port",
6  "golAnio": 34
7}
8
```

Yalbani Aranda

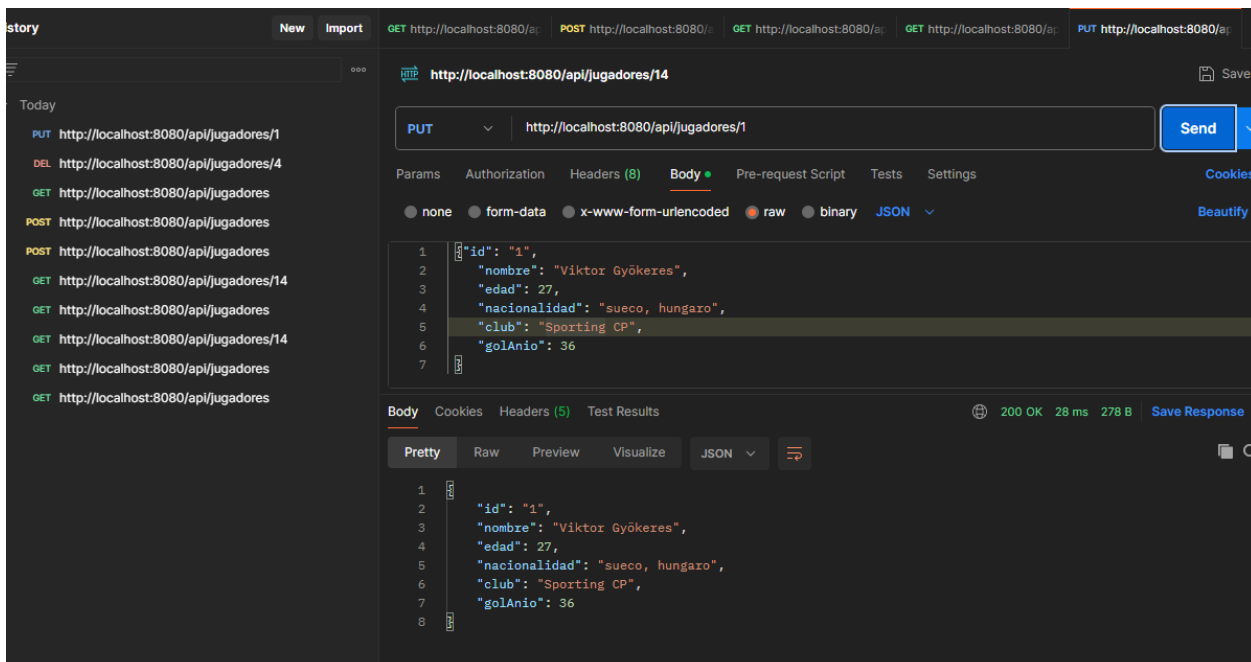
- La vista en el **localhost** 8080 <http://localhost:8080/api/jugadores> en formato JSON.



A screenshot of a web browser window. The address bar shows 'localhost:8080/api/jugadores'. The page content displays a JSON array of four player objects. The first player is Harry Kane (id: 9, age: 32, club: Bayern, goals: 29). The second is Erling Haaland (id: 14, age: 25, club: Manchester City, goals: 27). The third is Viktor Gyökeres (id: 1, age: 27, club: Sporting CP, goals: 36). The fourth is Cristiano Ronaldo (id: 17, age: 40, club: Al-Nassr, goals: 25). The browser interface includes a back button, a refresh button, and a search icon.

```
[
  {
    "id": "9",
    "nombre": "Harry Kane",
    "edad": 32,
    "nacionalidad": "ingles",
    "club": "Bayern",
    "golAnio": 29
  },
  {
    "id": "14",
    "nombre": "Erling Haaland",
    "edad": 25,
    "nacionalidad": "ingles, noruego",
    "club": "Manchester City",
    "golAnio": 27
  },
  {
    "id": "1",
    "nombre": "Viktor Gyökeres",
    "edad": 27,
    "nacionalidad": "sueco",
    "club": "Sporting CP",
    "golAnio": 36
  },
  {
    "id": "17",
    "nombre": "Cristiano Ronaldo",
    "edad": 40,
    "nacionalidad": "portugues",
    "club": "Al-Nassr",
    "golAnio": 25
  }
]
```

- PUT



A screenshot of a REST client interface. The top bar shows a list of requests, with the current one being a PUT request to 'http://localhost:8080/api/jugadores/1'. The main panel shows the request details: Method (PUT), URL (http://localhost:8080/api/jugadores/1), and Body (JSON). The body contains a JSON object for Viktor Gyökeres. The bottom panel shows the response details: Status (200 OK), Time (28 ms), and Size (278 B). The response body is displayed in a Pretty format, showing the same JSON object as the request body.

```
PUT http://localhost:8080/api/jugadores/1

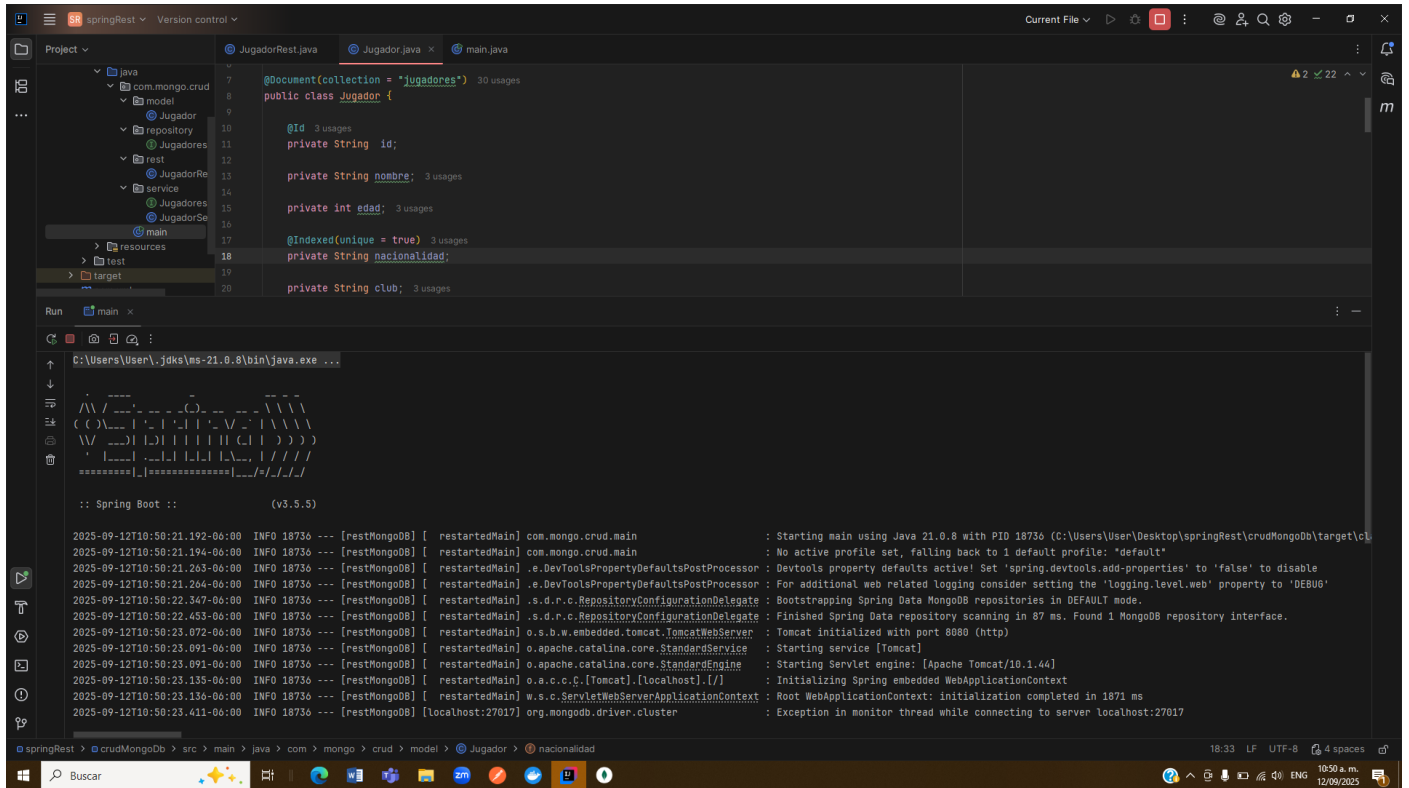
{
  "id": "1",
  "nombre": "Viktor Gyökeres",
  "edad": 27,
  "nacionalidad": "sueco, hungaro",
  "club": "Sporting CP",
  "golAnio": 36
}
```

```
200 OK 28 ms 278 B

{
  "id": "1",
  "nombre": "Viktor Gyökeres",
  "edad": 27,
  "nacionalidad": "sueco, hungaro",
  "club": "Sporting CP",
  "golAnio": 36
}
```

Yalbani Aranda

- IntelliJ Clase Jugador y Spring



The screenshot displays the IntelliJ IDEA IDE interface. The top toolbar shows the 'Run' button (a green play icon). The left sidebar contains the Project tool window, showing a hierarchical view of the project structure: `com.mongo.crud` > `model` > `Jugador`. The main editor window shows the `Jugador.java` file. The class is annotated with `@Document(collection = "jugadores")` and contains the following fields: `@Id private String id;`, `private String nombre;`, `private int edad;`, `@Indexed(unique = true) private String nacionalidad;`, and `private String club;`. The bottom panel shows the Run tool window with the following output:

```
C:\Users\User\jdk\ms-21.0.8\bin\java.exe ...

:: Spring Boot :: (v3.5.5)

2025-09-12T10:50:21.192-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] com.mongo.crud.main : Starting main using Java 21.0.8 with PID 18736 (C:\Users\User\Desktop\springRest\crudMongoDB\target\cl
2025-09-12T10:50:21.194-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] com.mongo.crud.main : No active profile set, falling back to 1 default profile: "default"
2025-09-12T10:50:21.263-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
2025-09-12T10:50:21.264-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
2025-09-12T10:50:22.453-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
2025-09-12T10:50:23.072-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 87 ms. Found 1 MongoDB repository interface.
2025-09-12T10:50:23.091-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-09-12T10:50:23.091-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-09-12T10:50:23.135-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.44]
2025-09-12T10:50:23.136-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-09-12T10:50:23.411-06:00 INFO 18736 --- [restMongoDB] [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1871 ms
2025-09-12T10:50:23.411-06:00 INFO 18736 --- [restMongoDB] [localhost:27017] org.mongodb.driver.cluster : Exception in monitor thread while connecting to server localhost:27017
```

The status bar at the bottom indicates the current file is `nacionalidad` within the `Jugador` class, and the system clock shows 10:50 a.m. on 12/09/2025.