



Laboratorio 3

Nombre: diego Alfredo Molina Medina

Sábado 28 de octubre del 2023

Modelo de posición:

```

package com.mitocode.model;

import lombok.*;
import javax.persistence.*;

@ToString
@Getter
@Setter
@Entity
@Table
@JsonIgnoreConstructor
@JsonIgnoreConstructor
@JsonIgnore
public class Posiciones {

    @ToString
    @JsonIgnore
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer idPosicion;

    @Column(nullable = false, length = 50)
    private String name;
}

```

Controlador de tabla posición:

The image shows a screenshot of an IDE with multiple tabs open at the top: Posiciones.java, MediciServiceImpl.java, PosicionesImpl.java, JugadorImpl.java, JugadorDTO.java, PatientController.java, MediciController.java, and PosicionController.java. The PosicionController.java tab is active, displaying the following Java code:

```
import java.util.stream.Collectors;

no usages
@RestController
@RequestMapping("/posiciones")
public class PosicionController {

    6 usages
    @Autowired
    private IPosicionesService service;

    3 usages
    @Autowired
    private ModelMapper mapper;

    @GetMapping
    public ResponseEntity<List<PosicionDTO>> findAll() {
        List<PosicionDTO> list = service.findAll().stream().map(p -> mapper.map(p, PosicionDTO.class)).collect(Collectors.toList());
        return new ResponseEntity<>(list, HttpStatus.OK);
    }

    @PostMapping
    public ResponseEntity<Void> save(@Valid @RequestBody PosicionDTO dto) {
        Posiciones p = service.save(mapper.map(dto, Posiciones.class));
        //localhost:8080/medics/3
        URI location = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}").buildAndExpand(p.getIdPosicion()).toUri();
        return ResponseEntity.created(location).build();
    }

    @PutMapping
    public ResponseEntity<Posiciones> update(@Valid @RequestBody PosicionDTO dto) {
        Posiciones obj = service.findById(dto.getIdPosicion());
        if (obj == null) {
            throw new ModelNotFoundException("ID NOT FOUND: " + dto.getIdPosicion());
        }

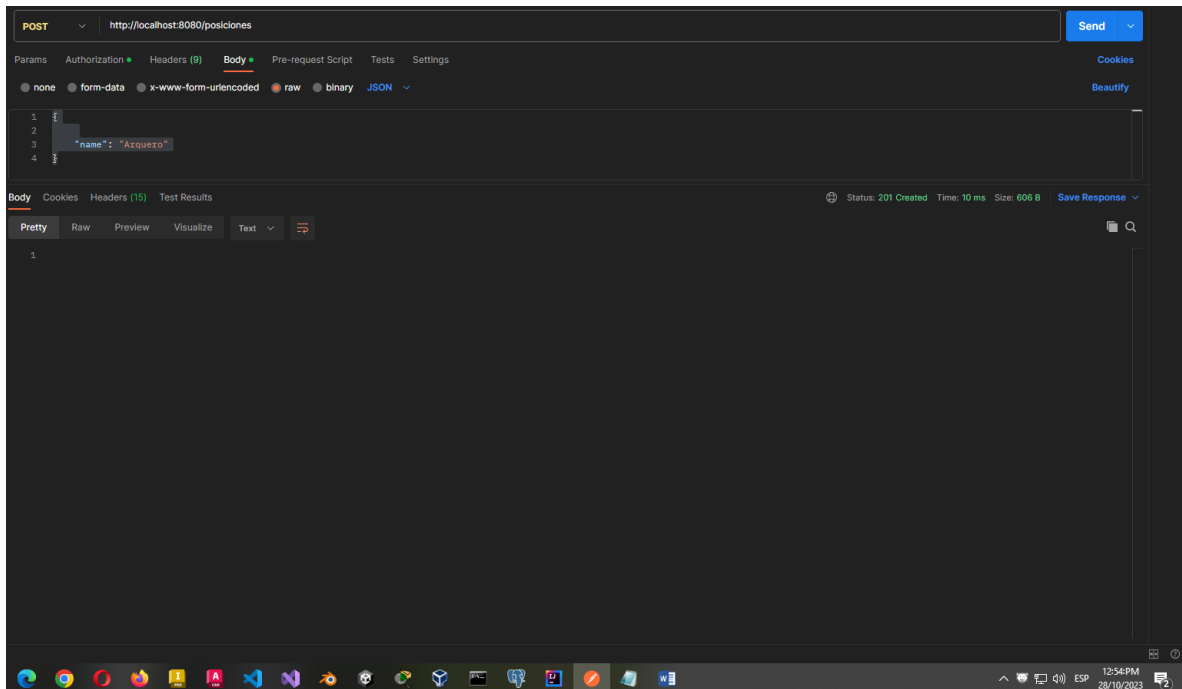
        return new ResponseEntity<>(service.update(mapper.map(dto, Posiciones.class)), HttpStatus.OK);
    }
}
```

At the bottom of the screen, there is a taskbar with various application icons and a system tray on the right showing the date and time: 3/11 CRLF UTF-8 4 spaces 12:51 PM 28/10/2023.

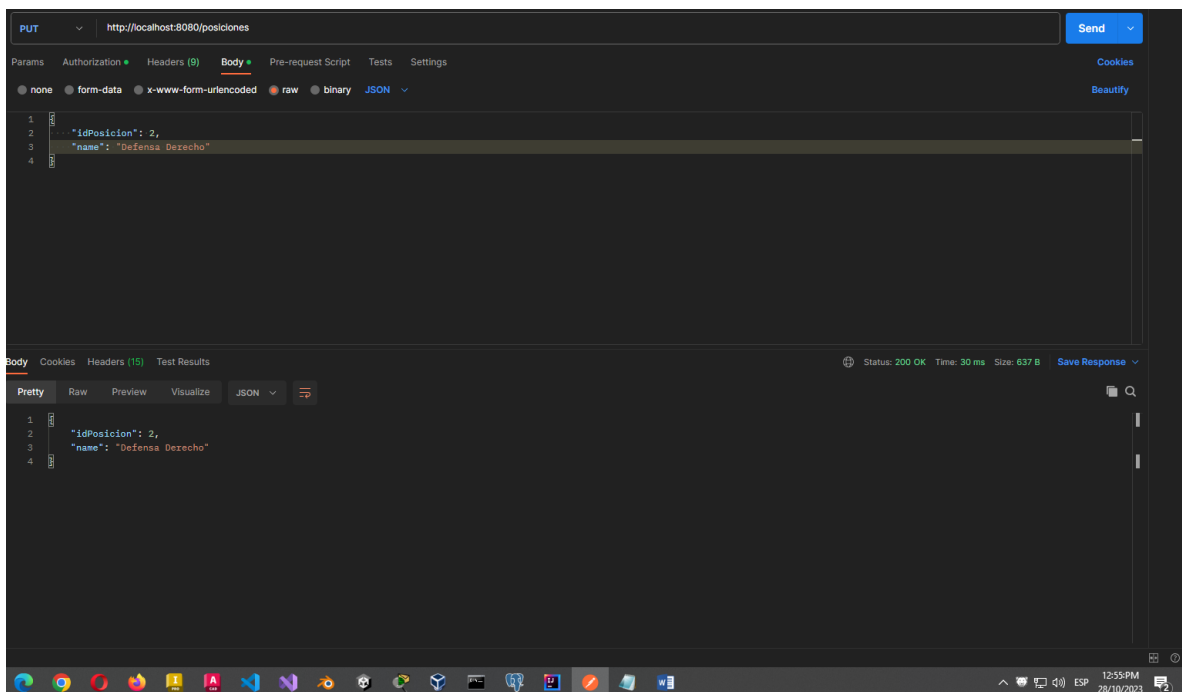
Obteniendo el token:



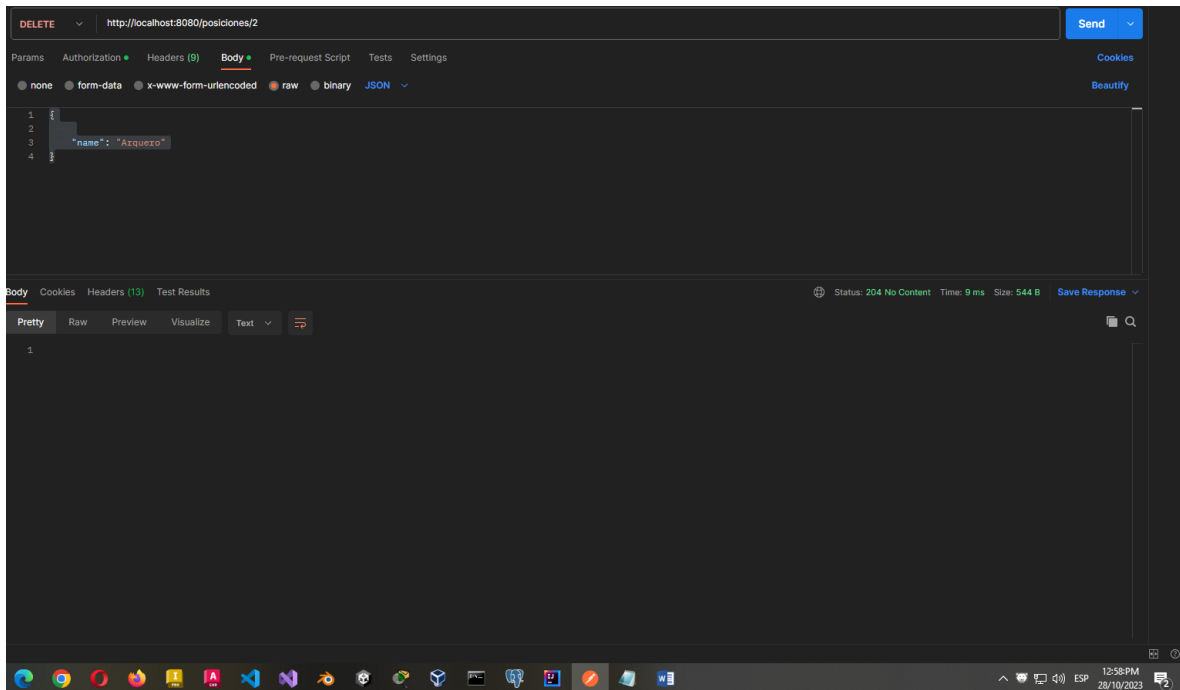
Insert:



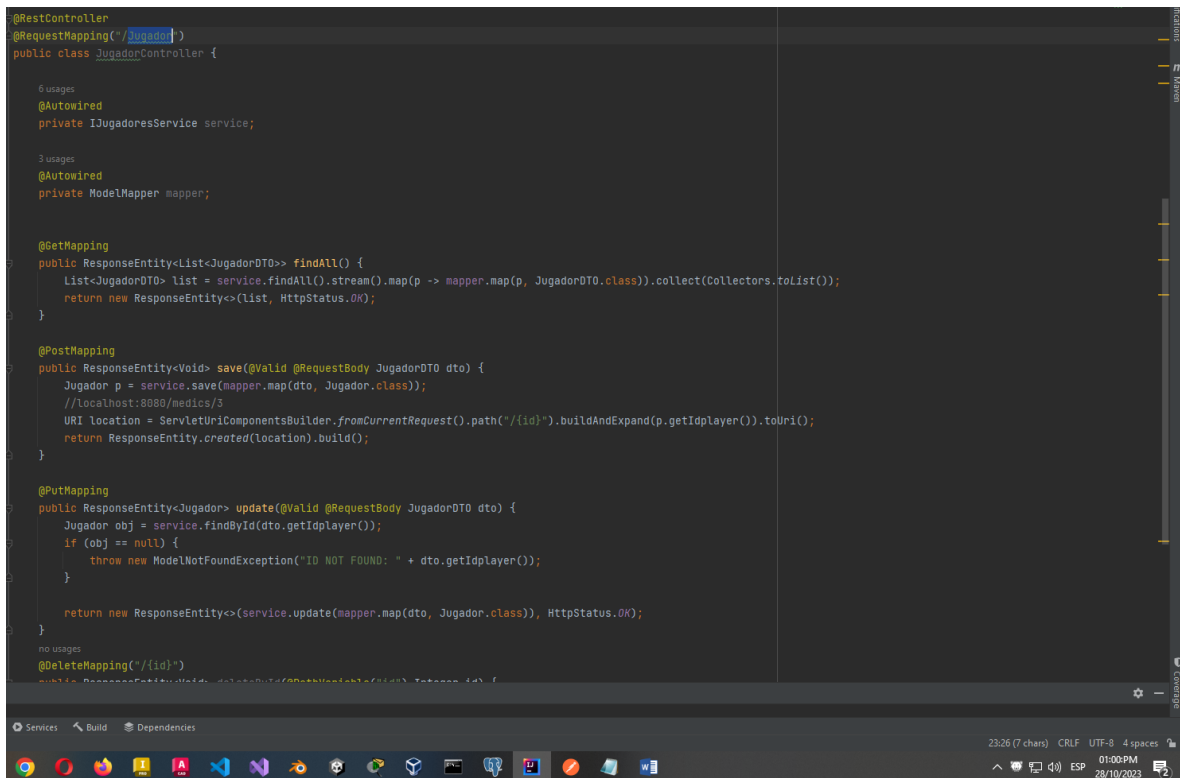
Update:



Delete:



Controlador De jugadores.



Modelo de Jugadores:

```
import lombok.*;

import javax.persistence.*;

15 usages
@Getter
@Setter
@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity

public class Jugador {

    no usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer idplayer;

    no usages
    @Column(nullable = false, length = 50)
    private String name;

    no usages
    @Column(nullable = false)
    private Integer num;

    no usages
    @ManyToOne
    @JoinColumn(name = "idPosicion")
    private Posiciones posiciones;

}

Services Build Dependencies
```

Captura postman:

Consulta:

```
GET http://localhost:8080/Jugador Send

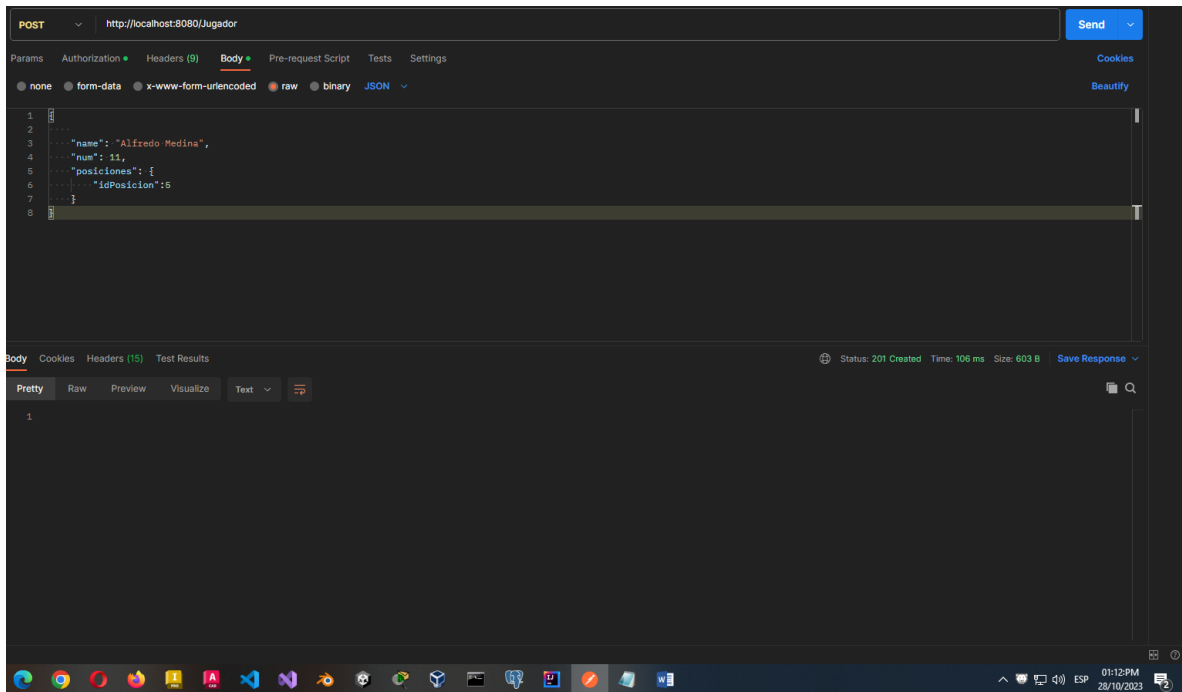
Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary JSON Beautify

1
2
3
4
5
6
7
8
{
  "name": "Alfredo Medina",
  "num": 11,
  "posiciones": {
    "idPosicion": 5
  }
}

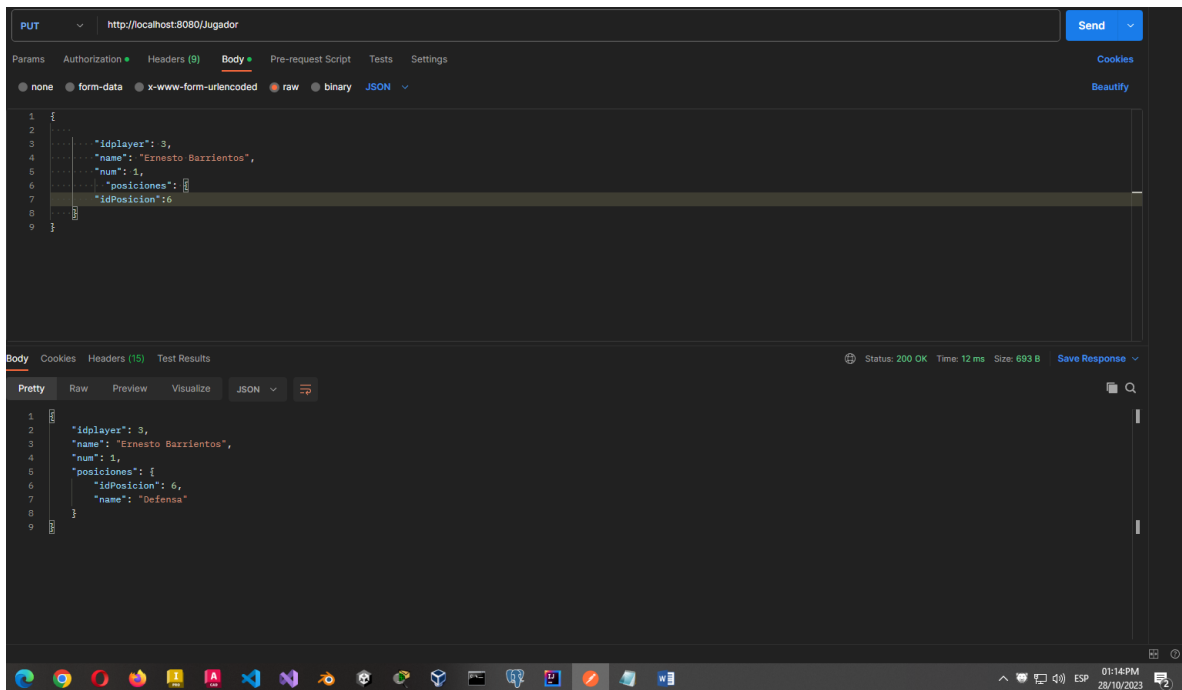
Body Cookies Headers (15) Test Results Status: 200 OK Time: 8 ms Size: 888 B Save Response
Pretty Raw Preview Visualize JSON

12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
{
  "posiciones": null
},
{
  "idplayer": 3,
  "name": "Ernesto Barrientos",
  "num": 4,
  "posiciones": null
},
{
  "idplayer": 4,
  "name": "Alfredo Medina",
  "num": 11,
  "posiciones": {
    "idPosicion": 5,
    "name": "Arquero"
  }
}
```

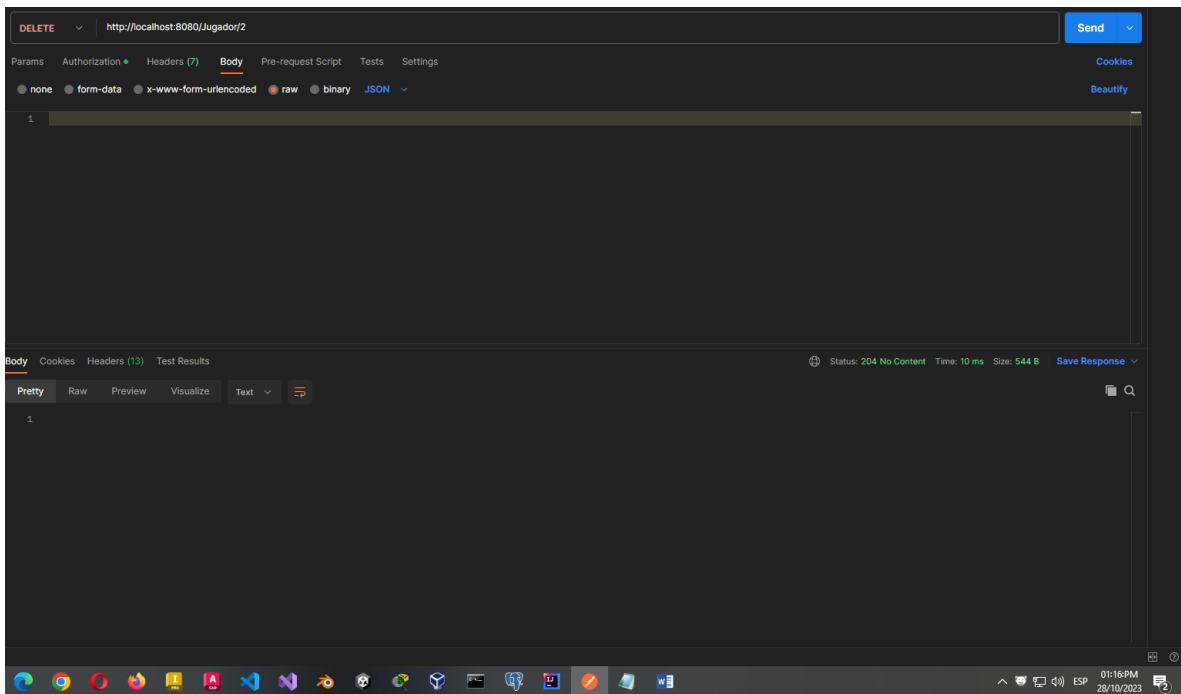
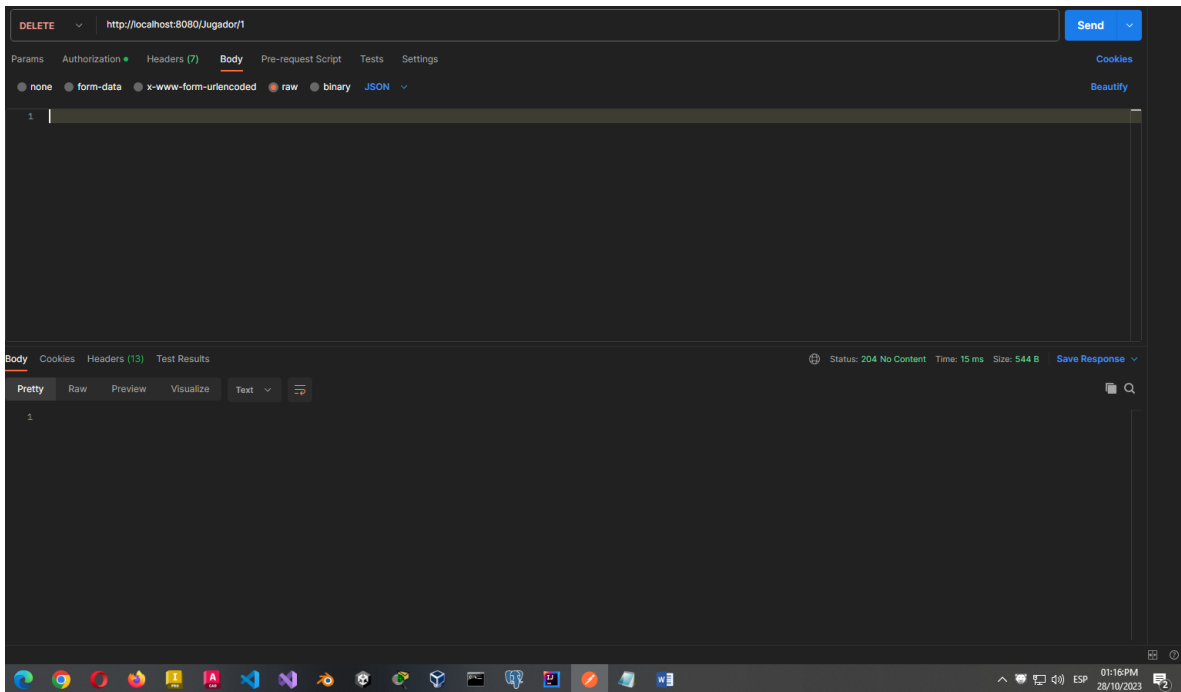
Insert:



Update:



Delete (se eliminaron 2 registros):



Captura de resultado final de la tabla:

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/Jugador`. The response is a JSON array of two player objects, displayed in the 'Body' tab. The status is 200 OK, and the response size is 790 B.

```
1 [
2   {
3     "idplayer": 4,
4     "name": "Alfredo Medina",
5     "num": 11,
6     "posiciones": {
7       "idPosicion": 5,
8       "name": "Arquero"
9     }
10  },
11  {
12    "idplayer": 3,
13    "name": "Ernesto Barrientos",
14    "num": 1,
15    "posiciones": {
16      "idPosicion": 6,
17      "name": "Defensa"
18    }
19  }
20 ]
```