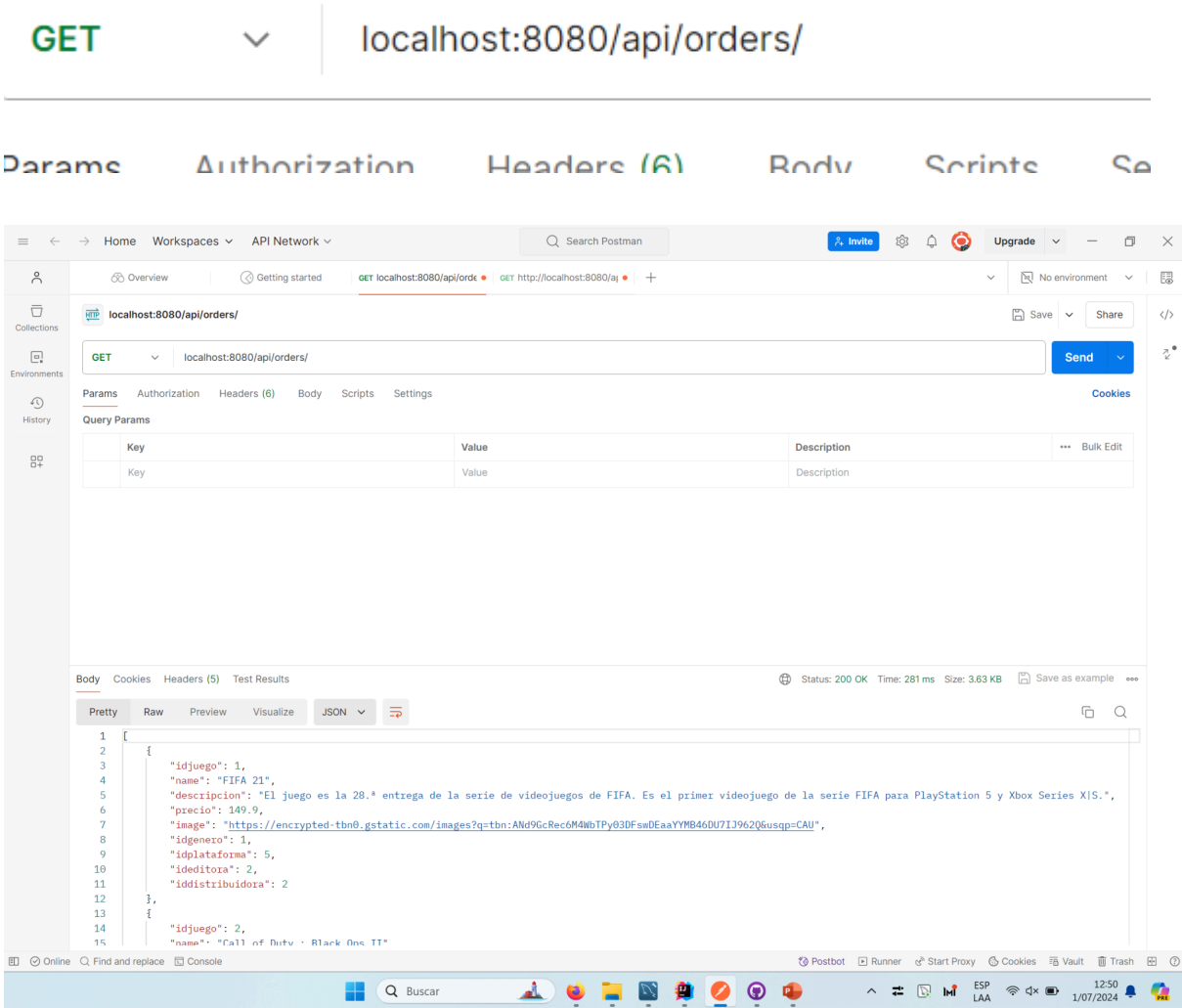


Examen final- GTICS

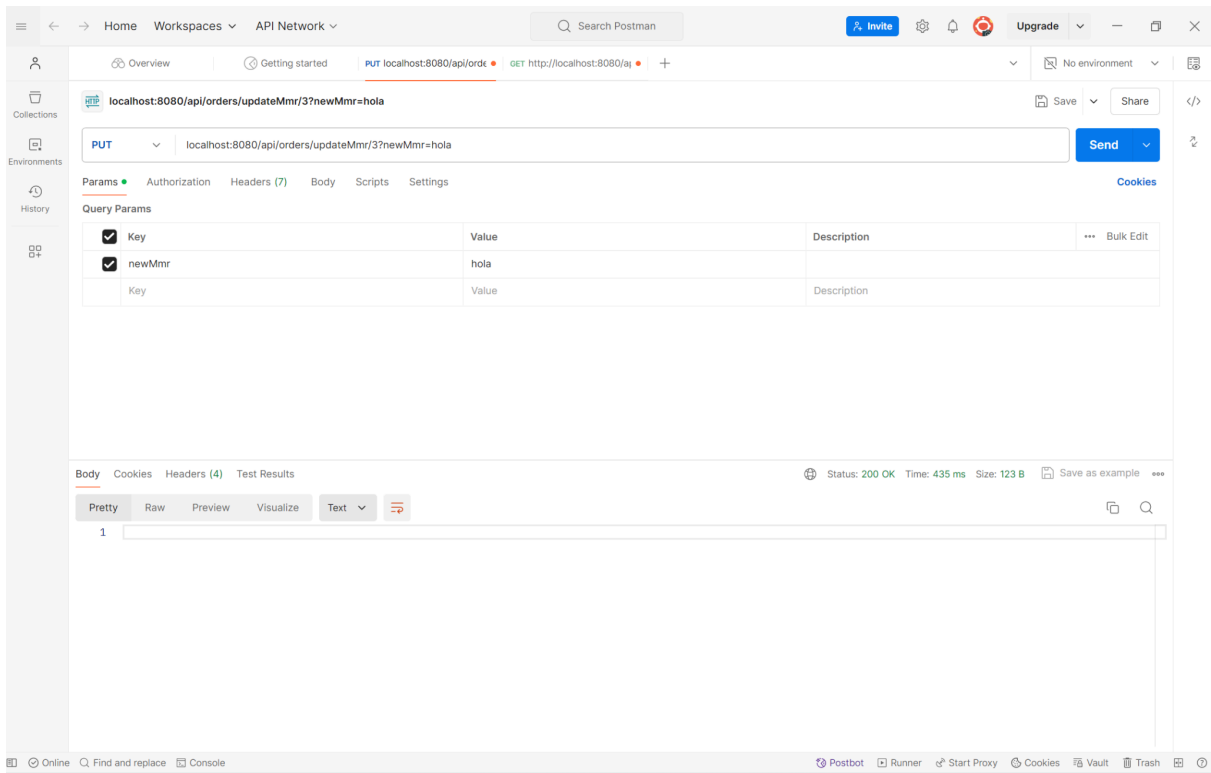
Nombre: Gerardo Manuel Gutierrez Aguilar

1. Usamos el metodo

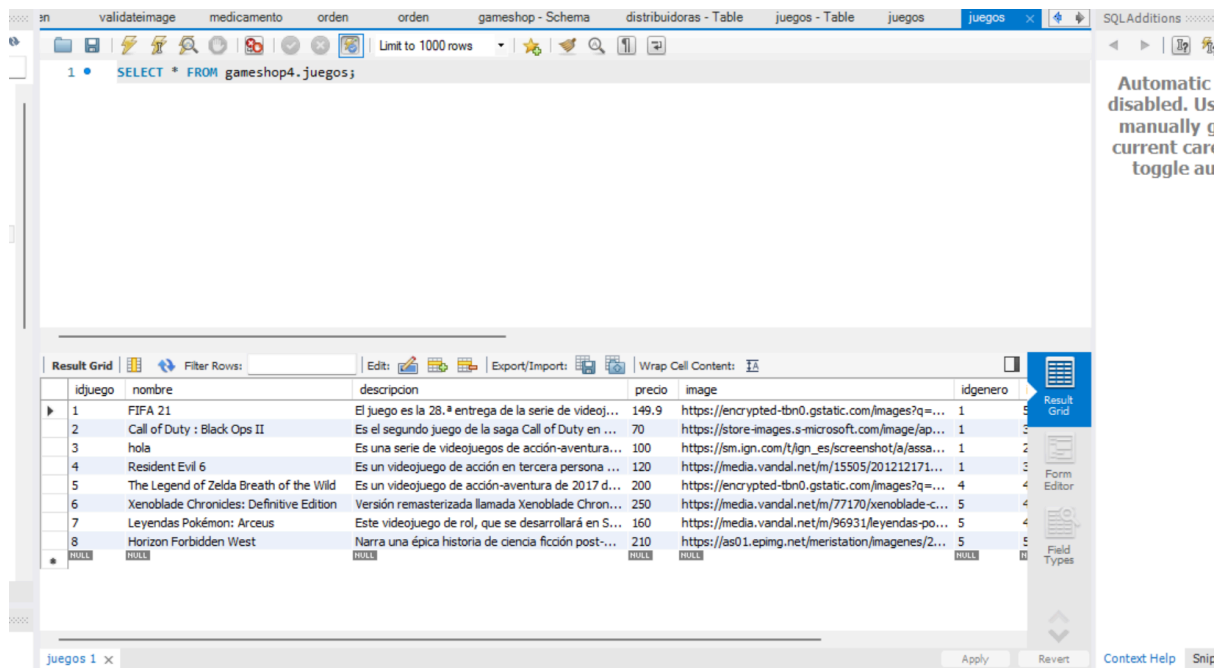


Actualizar

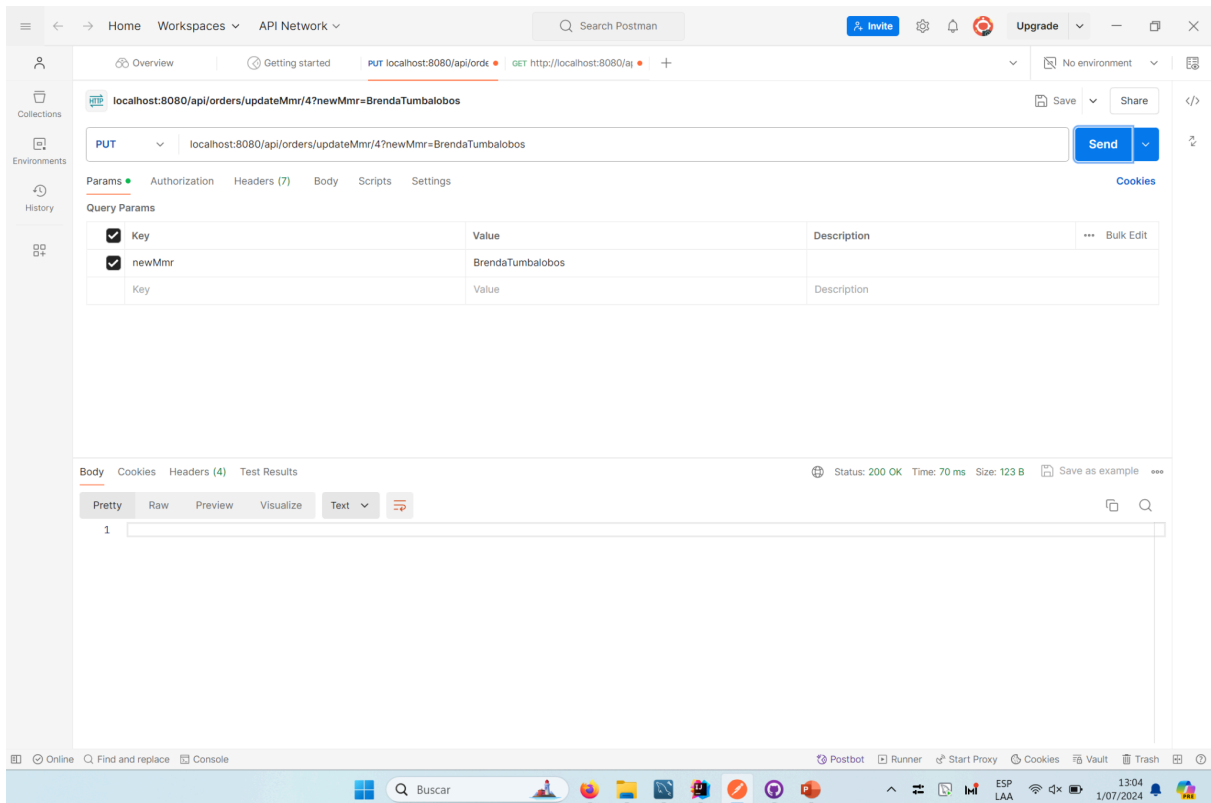
Estamos cambiando el de id =3



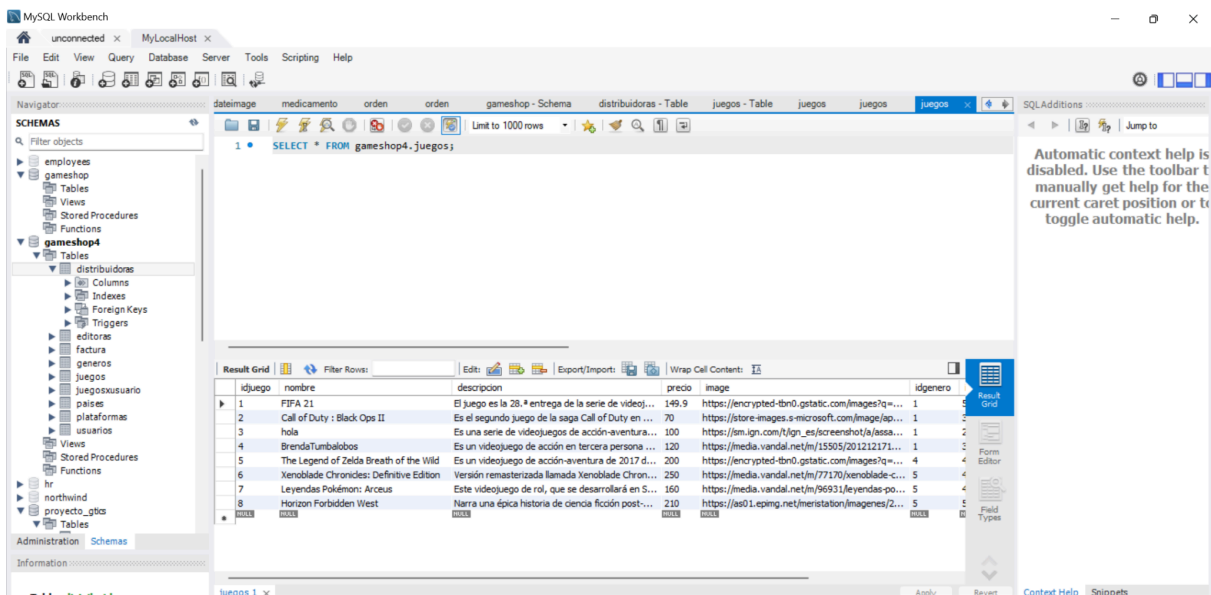
Finalmente, se puede visualizar el cambio



Otro ejemplo,



Se puede apreciar en el id=4 que cambia el nombre



Finalmente el codigo que se ha usado,

Logica de controller

```
@GetMapping("/")
public List<Juegos> getLeaderboard() {
    String var= "success";
    return playerRepository.findAll();
}

gerardo2810 *
@PostMapping("/add")
public void addPlayer(@RequestBody Juegos player) {
    playerRepository.save(player);
    recalculateLeaderboardPositions(player.getIdjuego());
}

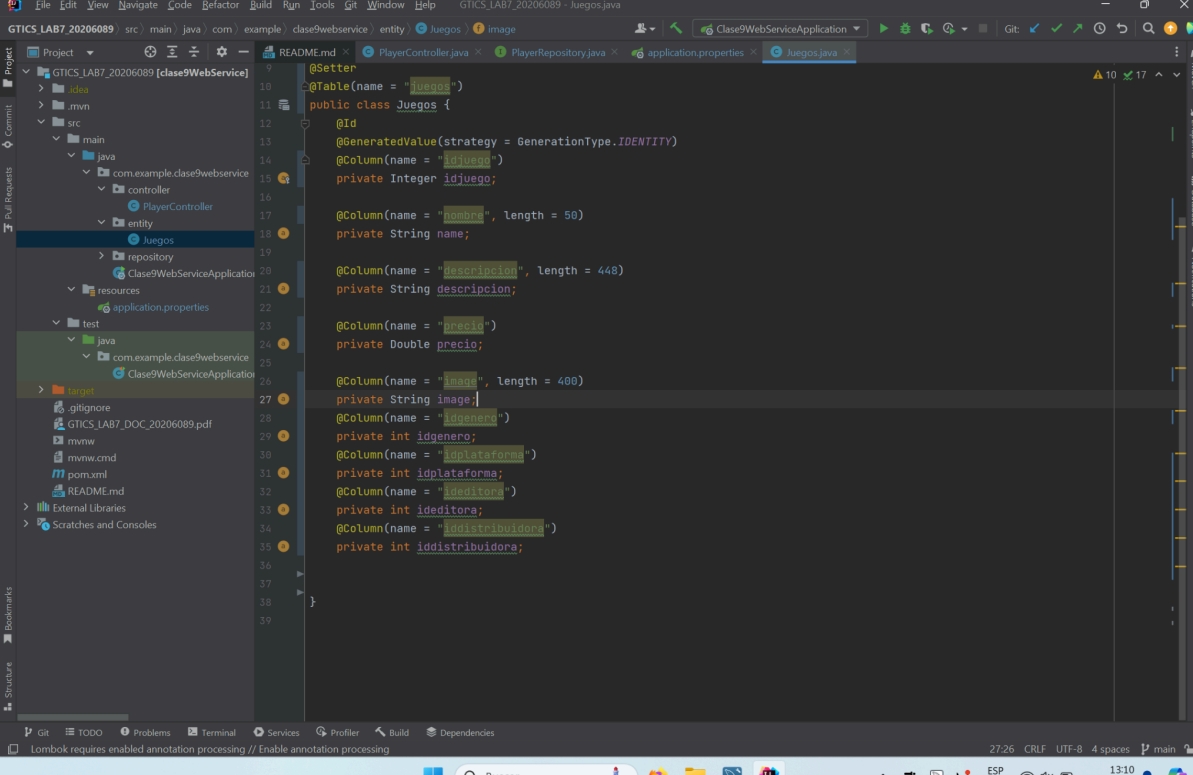
gerardo2810 *
@PutMapping("/updateMmr/{idjuego}")
public void updatePlayerMmr(@PathVariable int idjuego, @RequestParam String newMmr) {
    Juegos player = playerRepository.findById(idjuego).orElse( other: null);
    if (player != null) {
        player.setName(newMmr);
        playerRepository.save(player);
        recalculateLeaderboardPositions(player.getIdjuego());
    }
}

gerardo2810 *
@DeleteMapping("/delete/{idjuego}")
public void deletePlayer(@PathVariable int playerId) {
    Juegos player = playerRepository.findById(playerId).orElse( other: null);
    if (player != null) {

        playerRepository.delete(player);
    }
}

2 usages gerardo2810 *
private void recalculateLeaderboardPositions(Integer juego) {
    List<Juegos> players = playerRepository.findAll();
    for (int i = 0; i < players.size(); i++) {
        Juegos currentPlayer = players.get(i);
        currentPlayer.setIdjuego(i+1);
        playerRepository.save(currentPlayer);
    }
}
}
```

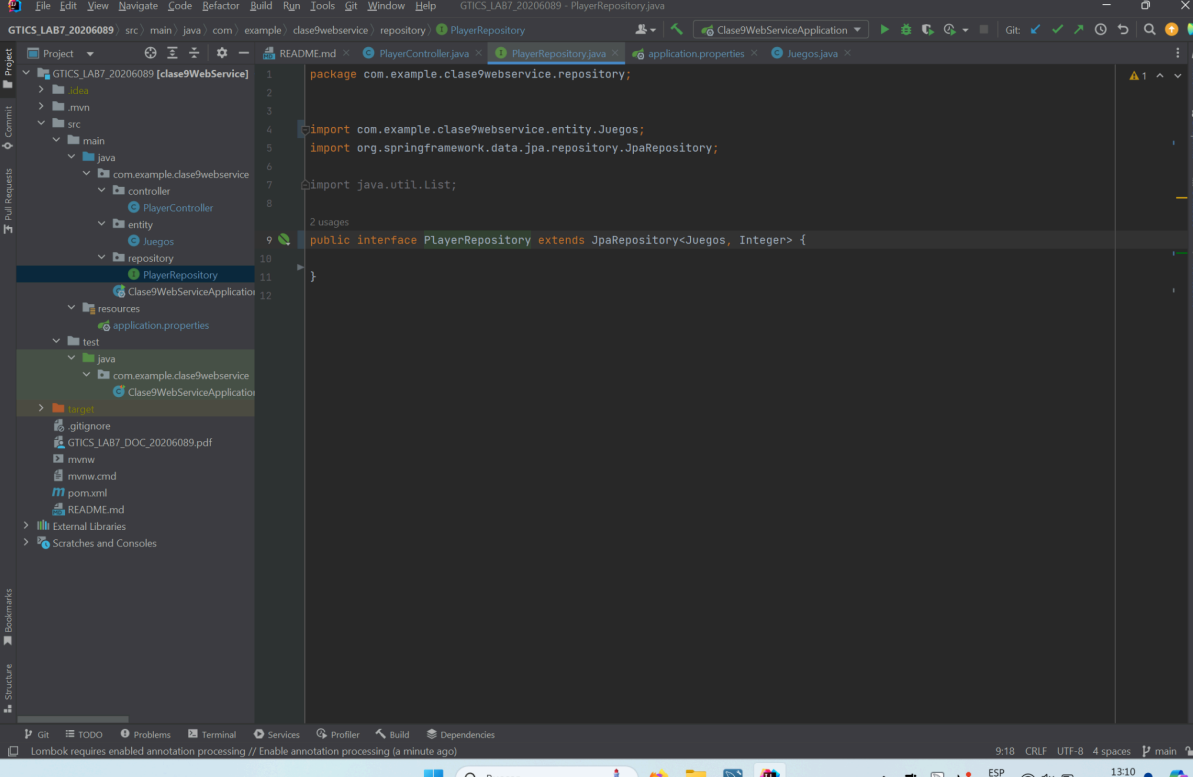
Entity:



The screenshot shows an IDE with the project structure on the left. The main editor displays the `Juegos.java` file. The class is annotated with `@Table(name = "juegos")` and `@Setter`. It contains several fields with corresponding database column annotations: `idjuego` (primary key, generated), `nombre` (length 50), `descripcion` (length 448), `precio` (Double), `image` (length 400), `idgenero`, `idplataforma`, `ideditora`, and `iddistribuidora`.

```
9 @Setter
10 @Table(name = "juegos")
11 public class Juegos {
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     @Column(name = "idjuego")
15     private Integer idjuego;
16
17     @Column(name = "nombre", length = 50)
18     private String nombre;
19
20     @Column(name = "descripcion", length = 448)
21     private String descripcion;
22
23     @Column(name = "precio")
24     private Double precio;
25
26     @Column(name = "image", length = 400)
27     private String image;
28     @Column(name = "idgenero")
29     private int idgenero;
30     @Column(name = "idplataforma")
31     private int idplataforma;
32     @Column(name = "ideditora")
33     private int ideditora;
34     @Column(name = "iddistribuidora")
35     private int iddistribuidora;
36
37 }
38
39
```

Repository:



The screenshot shows the same IDE with the `PlayerRepository.java` file open. It defines a package `com.example.clase9webservice.repository` and imports `Juegos` and `JpaRepository`. The interface `PlayerRepository` extends `JpaRepository<Juegos, Integer>`.

```
1 package com.example.clase9webservice.repository;
2
3
4 import com.example.clase9webservice.entity.Juegos;
5 import org.springframework.data.jpa.repository.JpaRepository;
6
7 import java.util.List;
8
9 2 usages
10 public interface PlayerRepository extends JpaRepository<Juegos, Integer> {
11 }
12
```

De todas formas se colocara el github donde estara el pdf y el codigo completo que he usado.