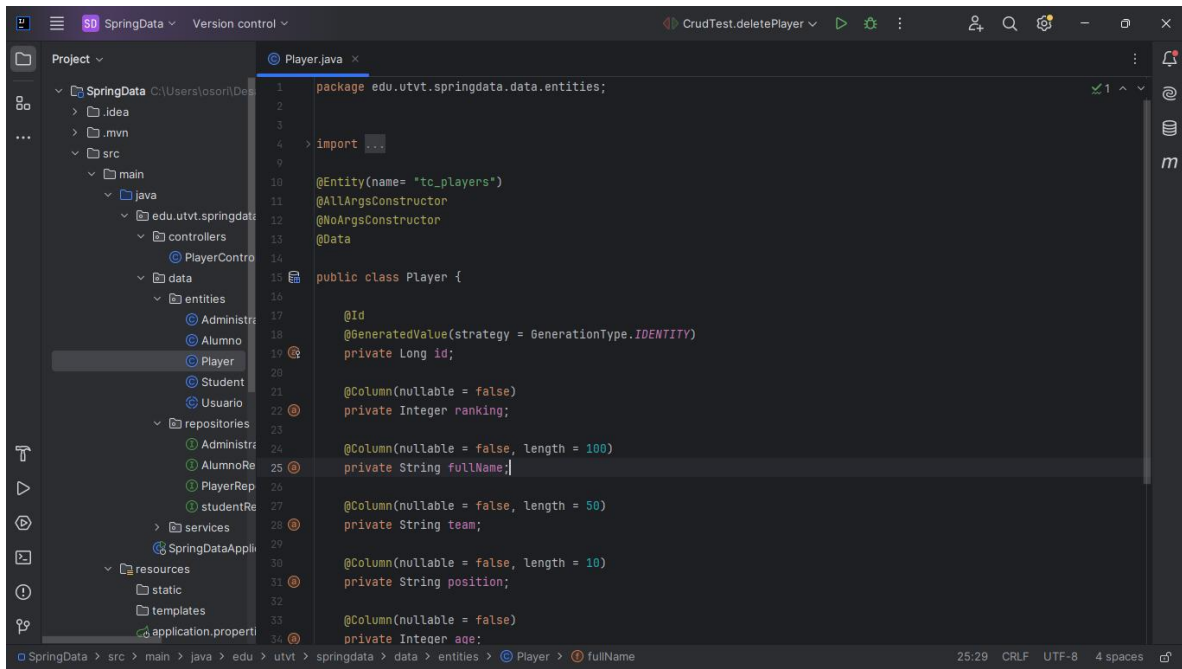
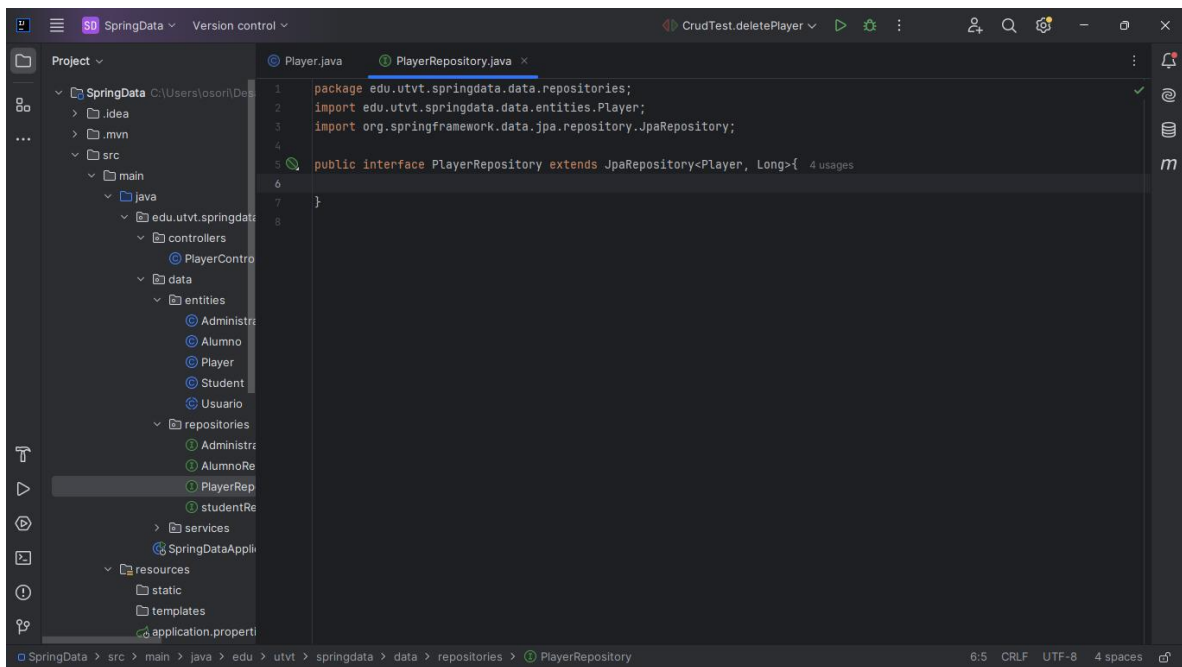


Player Entities



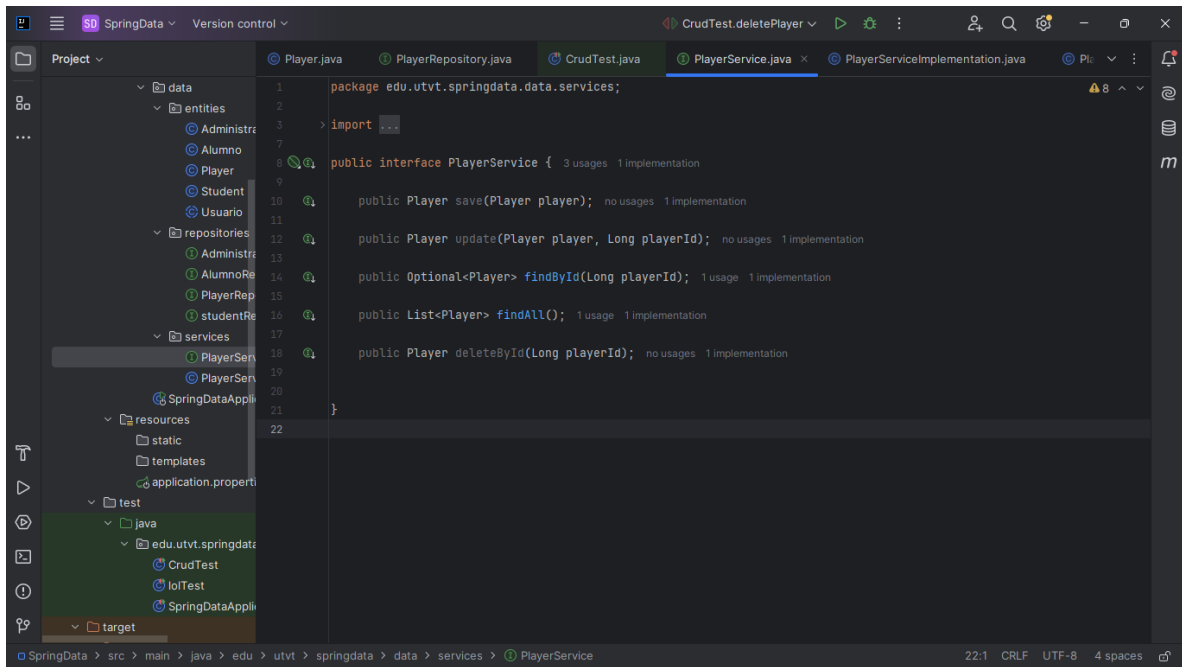
```
1 package edu.utvt.springdata.data.entities;
2
3
4 > import ...
5
6
7
8
9
10 @Entity(name= "tc_players")
11 @AllArgsConstructor
12 @NoArgsConstructor
13 @Data
14
15 public class Player {
16
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private Long id;
20
21     @Column(nullable = false)
22     private Integer ranking;
23
24     @Column(nullable = false, length = 100)
25     private String fullName;
26
27     @Column(nullable = false, length = 50)
28     private String team;
29
30     @Column(nullable = false, length = 10)
31     private String position;
32
33     @Column(nullable = false)
34     private Integer age;
```

Player Repository



```
1 package edu.utvt.springdata.data.repositories;
2 import edu.utvt.springdata.data.entities.Player;
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 public interface PlayerRepository extends JpaRepository<Player, Long>{
6
7 }
8
```

Player Service



```
package edu.utvt.springdata.data.services;

import java.util.List;
import java.util.Optional;

public interface PlayerService {

    public Player save(Player player);

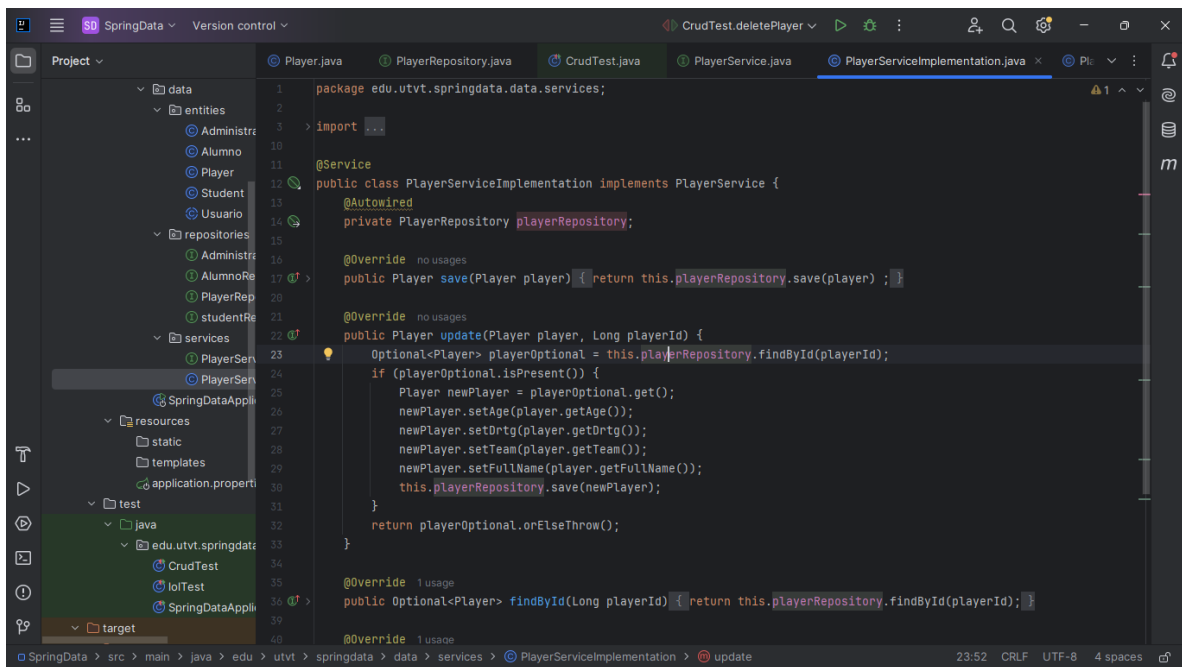
    public Player update(Player player, Long playerId);

    public Optional<Player> findById(Long playerId);

    public List<Player> findAll();

    public Player deleteById(Long playerId);
}
```

PlayerServicesImplementation



```
package edu.utvt.springdata.data.services;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
public class PlayerServiceImplementation implements PlayerService {

    @Autowired
    private PlayerRepository playerRepository;

    @Override
    public Player save(Player player) {
        return this.playerRepository.save(player);
    }

    @Override
    public Player update(Player player, Long playerId) {
        Optional<Player> playerOptional = this.playerRepository.findById(playerId);
        if (playerOptional.isPresent()) {
            Player newPlayer = playerOptional.get();
            newPlayer.setAge(player.getAge());
            newPlayer.setDrtg(player.getDrtg());
            newPlayer.setTeam(player.getTeam());
            newPlayer.setFullName(player.getFullName());
            this.playerRepository.save(newPlayer);
        }
        return playerOptional.orElseThrow();
    }

    @Override
    public Optional<Player> findById(Long playerId) {
        return this.playerRepository.findById(playerId);
    }

    @Override
    public List<Player> findAll() {
        return this.playerRepository.findAll();
    }

    @Override
    public Player deleteById(Long playerId) {
        this.playerRepository.deleteById(playerId);
        return null;
    }
}
```

Test de Player

Player Controller

```

package edu.utvt.springdata.controllers;

import ...

@RestController
@RequestMapping("/v1/players")
public class PlayerController {

    @Autowired
    private PlayerService playerService;

    @GetMapping("/v1/all")
    public List<Player> getAllPlayers() { return playerService.findAll(); }

    @GetMapping("/v1/{id}")
    public ResponseEntity<Player> getPlayerById(@PathVariable Long id) {
        return ResponseEntity.of(this.playerService.findById(id));
    }
}

```