

COMMAND QUERY RESPONSABILITY SEGREGATION

INTRODUCCIÓN A CQRS

Daniel Blanco Calviño

COMANDOS Y CONSULTAS

- **Comandos.**

- Acciones que **realizan una modificación en el estado del sistema** y que no devuelven información.
- Ejemplo: Iniciar un partido de tenis.

- **Consultas.**

- Acciones que **no alteran el estado del sistema**, tan sólo devuelven datos.
- Ejemplo: Obtener los últimos diez partidos de tenis.

```

5  public class TennisMatch {
6      Long id;
7      String player1;
8      String player2;
9      String winner;
10     String status;
11     List<TennisSet> sets;
12
13     // Getters and Setters
14 }

```

```

6  public class TennisMatch {
7      private static final int NUMBER_OF_SETS = 5;
8      private static final int MAX_WARNINGS = 3;
9
10     private Map<String, Player> players;
11     private String player1;
12     private String player2;
13     private Status status;
14     private List<TennisSet> sets;
15     private TennisSet currentSet;
16     private Player winner;
17
18     public TennisMatch(String player1, String player2) {...}
19
20
21
22
23
24
25
26
27
28     public void start() {...}
29
30
31
32
33
34     public void finish(String winnerPlayer) {...}
35
36
37
38
39
40
41     public void point(String player) {...}
42
43
44
45
46
47
48
49
50
51     private void advanceSet() {...}
52
53
54
55
56     public void warn(String player) {...}
57
58
59
60
61
62
63
64
65     public Player getWinner() {...}
66
67
68
69
70
71
72 }

```

```

5  public class TennisMatch {
6      Long id;
7      String player1;
8      String player2;
9      String winner;
10     String status;
11     List<TennisSet> sets;
12
13     // Getters and Setters
14 }

```

- Representa fielmente nuestro modelo en Base de Datos.
- No tiene lógica.
- Riesgo de estados incongruentes.
- **Apropiado para las consultas.**

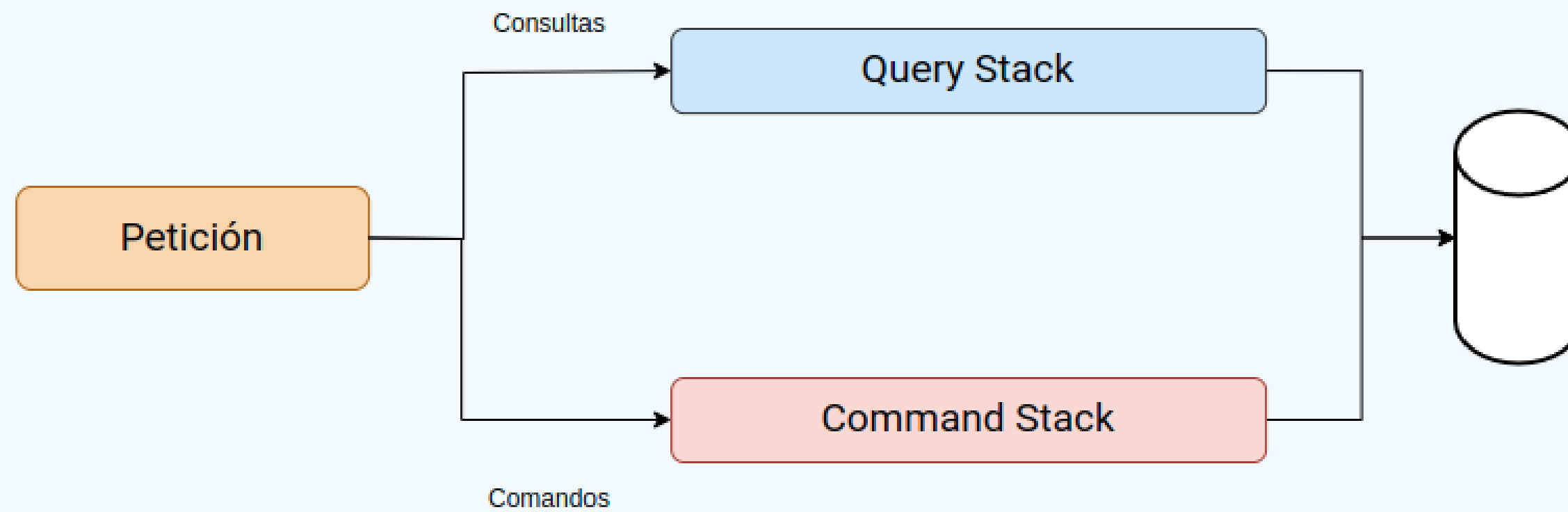
```

6  public class TennisMatch {
7      private static final int NUMBER_OF_SETS = 5;
8      private static final int MAX_WARNINGS = 3;
9
10     private Map<String, Player> players;
11     private String player1;
12     private String player2;
13     private Status status;
14     private List<TennisSet> sets;
15     private TennisSet currentSet;
16     private Player winner;
17
18     public TennisMatch(String player1, String player2) {...}
19
20     public void start() {...}
21
22     public void finish(String winnerPlayer) {...}
23
24     public void point(String player) {...}
25
26     private void advanceSet() {...}
27
28     public void warn(String player) {...}
29
30     public Player getWinner() {...}
31 }

```

- Representa fielmente nuestro modelo de dominio.
- Necesita modificaciones para la persistencia.
- Exponemos lógica de negocio a capas superiores que no la necesitan.
- **Apropiado para los comandos.**

CQRS



COMMAND QUERY RESPONSABILITY SEGREGATION

INTRODUCCIÓN A CQRS

Daniel Blanco Calviño