

**Consultas realizadas:**

**Alta de jugadores:**

```
INSERT INTO jugadores (nombre, email, puntosTotales) VALUES (?, ?, ?);
```

Inserta un nuevo jugador en la base de datos con puntuación inicial a 0 y guarda el ID generado.

**Contar partidas:**

```
SELECT COUNT(*) FROM partidas;
```

Se utiliza para comprobar que no se superen las 5 partidas máximas permitidas.

**Alta de partidas:**

```
INSERT INTO partidas (narrador_id, fecha, resultado) VALUES (?, ?, ?);
```

Guarda una nueva partida.

**Sumar puntos a un jugador:**

```
UPDATE jugadores SET puntosTotales = puntosTotales + ? WHERE id = ?;
```

Actualiza la puntuación de un jugador según el resultado de la partida.

**Jugador con más puntos:**

```
SELECT nombre, puntosTotales FROM jugadores ORDER BY puntosTotales DESC LIMIT 1;
```

Muestra el jugador que tiene más puntos.

**Ranking de jugadores:**

```
SELECT nombre, puntosTotales FROM jugadores ORDER BY puntosTotales DESC;
```

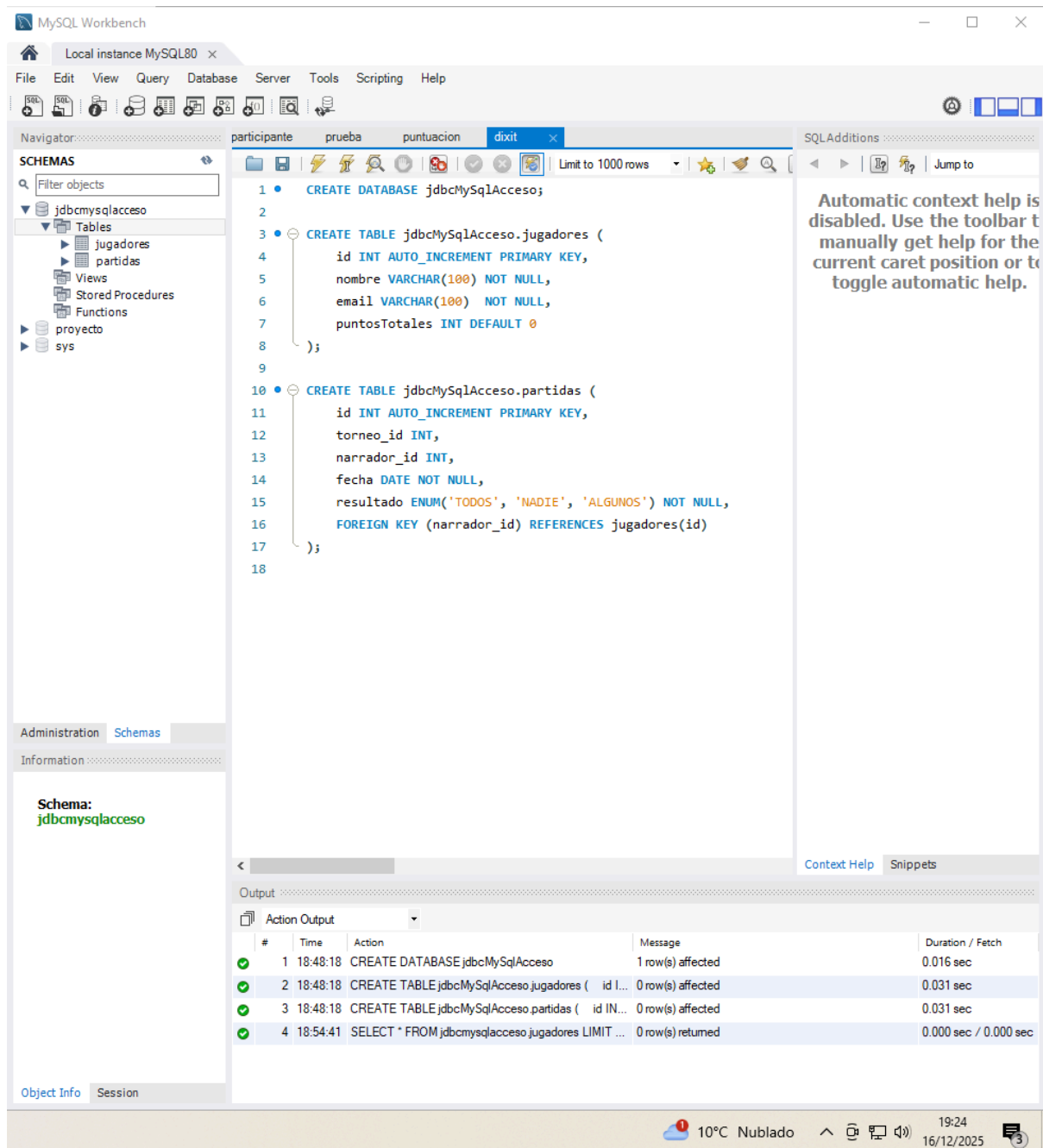
Lista todos los jugadores ordenados por puntos.

**Listar partidas por fecha:**

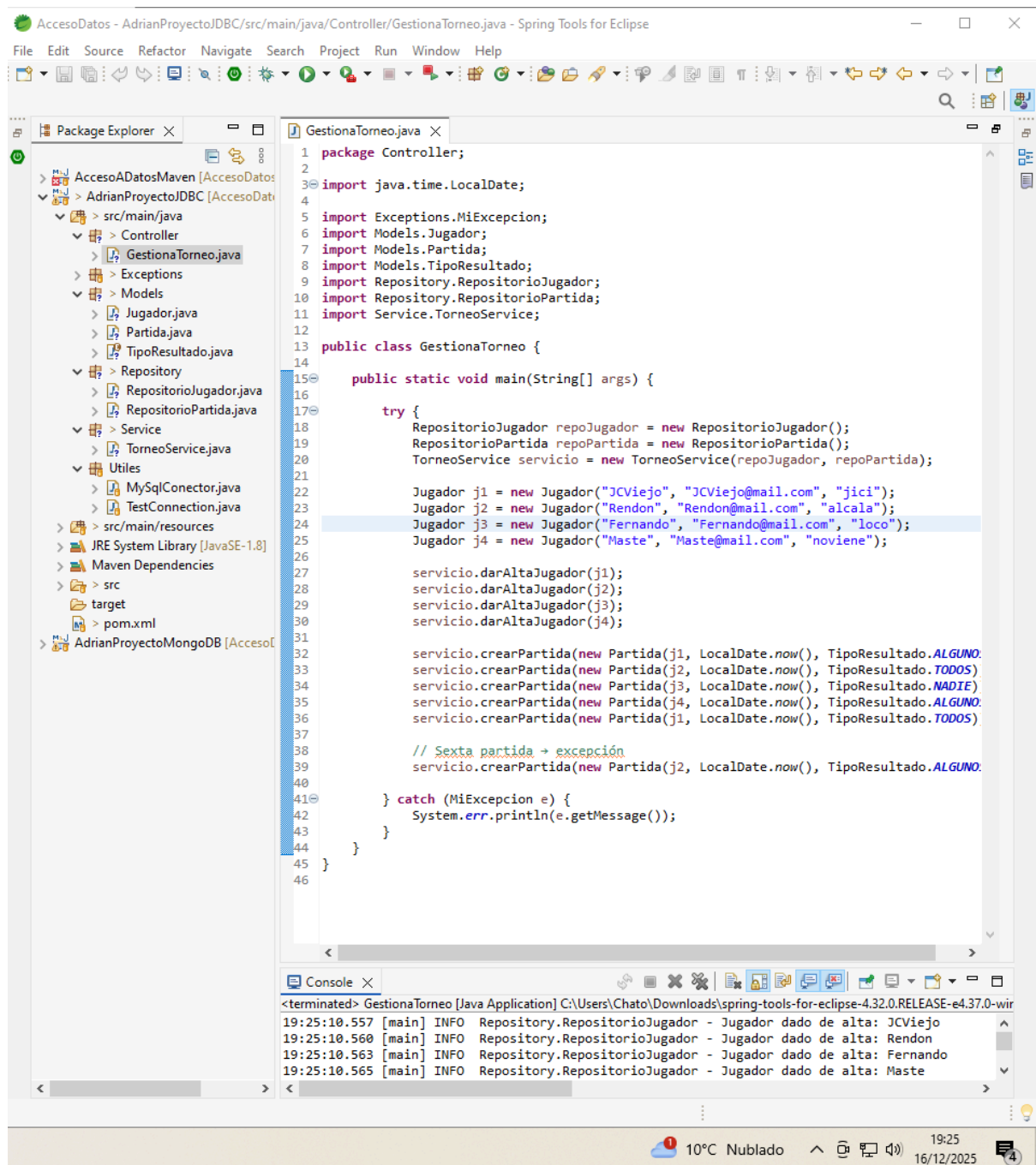
```
SELECT * FROM partidas ORDER BY fecha;
```

Muestra las partidas ordenadas.

## Capturas:



Esquema y base de datos creada.



```
1 package Controller;
2
3 import java.time.LocalDate;
4
5 import Exceptions.MiExcepcion;
6 import Models.Jugador;
7 import Models.Partida;
8 import Models.TipoResultado;
9 import Repository.RepositorioJugador;
10 import Repository.RepositorioPartida;
11 import Service.TorneoService;
12
13 public class GestionaTorneo {
14
15     public static void main(String[] args) {
16
17         try {
18             RepositorioJugador repoJugador = new RepositorioJugador();
19             RepositorioPartida repoPartida = new RepositorioPartida();
20             TorneoService servicio = new TorneoService(repoJugador, repoPartida);
21
22             Jugador j1 = new Jugador("JC Viejo", "JC Viejo@mail.com", "jici");
23             Jugador j2 = new Jugador("Rendon", "Rendon@mail.com", "alcala");
24             Jugador j3 = new Jugador("Fernando", "Fernando@mail.com", "loco");
25             Jugador j4 = new Jugador("Maste", "Maste@mail.com", "noviene");
26
27             servicio.darAltaJugador(j1);
28             servicio.darAltaJugador(j2);
29             servicio.darAltaJugador(j3);
30             servicio.darAltaJugador(j4);
31
32             servicio.crearPartida(new Partida(j1, LocalDate.now(), TipoResultado.ALGUNO);
33             servicio.crearPartida(new Partida(j2, LocalDate.now(), TipoResultado.TODOS);
34             servicio.crearPartida(new Partida(j3, LocalDate.now(), TipoResultado.NADIE);
35             servicio.crearPartida(new Partida(j4, LocalDate.now(), TipoResultado.ALGUNO);
36             servicio.crearPartida(new Partida(j1, LocalDate.now(), TipoResultado.TODOS);
37
38             // Sexta partida + excepción
39             servicio.crearPartida(new Partida(j2, LocalDate.now(), TipoResultado.ALGUNO);
40
41         } catch (MiExcepcion e) {
42             System.err.println(e.getMessage());
43         }
44     }
45 }
46
```

Console

```
<terminated> GestionaTorneo [Java Application] C:\Users\Chato\Downloads\spring-tools-for-eclipse-4.32.0.RELEASE-e4.37.0-wir
19:25:10.557 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: JC Viejo
19:25:10.560 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Rendon
19:25:10.563 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Fernando
19:25:10.565 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Maste
```

Damos de alta a 4 jugadores.

The screenshot displays the MySQL Workbench interface for a local instance named 'MySQL80'. The left sidebar shows the 'SCHEMAS' tree with 'jdbcmysqlacceso' selected, containing tables 'jugadores' and 'partidas'. The central editor shows a SQL script with the following queries:

```
1 use jdbcMySQLAcceso;
2
3 select * from jugadores;
4
```

The 'Result Grid' below the script shows the results of the query 'select \* from jugadores', displaying 4 rows of player data:

id	nombre	email	puntosTotales
1	JCViejo	JCViejo@mail.com	0
2	Rendon	Rendon@mail.com	0
3	Fernando	Fernando@mail.com	0
4	Maste	Maste@mail.com	0

The bottom 'Output' pane shows the 'Action Output' log, confirming the successful execution of the database creation and query:

#	Time	Action	Message	Duration / Fetch
1	18:48:18	CREATE DATABASE jdbcMySQLAcceso	1 row(s) affected	0.016 sec
2	18:48:18	CREATE TABLE jdbcMySQLAcceso jugadores ( id I...	0 row(s) affected	0.031 sec
3	18:48:18	CREATE TABLE jdbcMySQLAcceso partidas ( id IN...	0 row(s) affected	0.031 sec
4	18:54:41	SELECT * FROM jdbcmysqlacceso jugadores LIMIT ...	0 row(s) returned	0.000 sec / 0.000 sec
5	19:26:52	use jdbcMySQLAcceso	0 row(s) affected	0.000 sec
6	19:26:52	select * from jugadores LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

A notification on the right states: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

Comprobamos que se han añadido correctamente.

```
7 import Models.Partida;
8 import Models.TipoResultado;
9 import Repository.RepositorioJugador;
10 import Repository.RepositorioPartida;
11 import Service.TorneoService;
12
13 public class GestionaTorneo {
14
15     public static void main(String[] args) {
16
17         try {
18             RepositorioJugador repoJugador = new RepositorioJugador();
19             RepositorioPartida repoPartida = new RepositorioPartida();
20             TorneoService servicio = new TorneoService(repoJugador, repoPartida);
21
22             Jugador j1 = new Jugador("JCviejo", "JCviejo@mail.com", "jici");
23             Jugador j2 = new Jugador("Rendon", "Rendon@mail.com", "alcala");
24             Jugador j3 = new Jugador("Fernando", "Fernando@mail.com", "loco");
25             Jugador j4 = new Jugador("Maste", "Maste@mail.com", "noviene");
26
27             servicio.darAltaJugador(j1);
28             servicio.darAltaJugador(j2);
29             servicio.darAltaJugador(j3);
30             servicio.darAltaJugador(j4);
31
32             servicio.crearPartida(new Partida(j1, LocalDate.now(), TipoResultado.ALGUNO);
33             servicio.crearPartida(new Partida(j2, LocalDate.now(), TipoResultado.TODOS);
34             servicio.crearPartida(new Partida(j3, LocalDate.now(), TipoResultado.NADIE);
35             servicio.crearPartida(new Partida(j4, LocalDate.now(), TipoResultado.ALGUNO);
36             servicio.crearPartida(new Partida(j1, LocalDate.now(), TipoResultado.TODOS);
37
38             // Sexta partida + excepción
39             servicio.crearPartida(new Partida(j2, LocalDate.now(), TipoResultado.ALGUNO);
40
41         } catch (MiExcepcion e) {
42             System.err.println(e.getMessage());
43         }
44     }
45 }
46
```

Console Output:

```
<terminated> GestionaTorneo [Java Application] C:\Users\Chato\Downloads\spring-tools-for-eclipse-4.32.0.RELEASE-e4.37.0-wir
19:32:14.660 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: JCviejo
19:32:14.664 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Rendon
19:32:14.666 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Fernando
19:32:14.669 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Maste
19:32:14.683 [main] INFO Repository.RepositorioPartida - Partida dada de alta
19:32:14.687 [main] INFO Repository.RepositorioPartida - Partida dada de alta
19:32:14.711 [main] INFO Repository.RepositorioPartida - Partida dada de alta
19:32:14.733 [main] INFO Repository.RepositorioPartida - Partida dada de alta
19:32:14.753 [main] INFO Repository.RepositorioPartida - Partida dada de alta
No se pueden crear más de 5 partidas
```

Creamos las 5 partidas.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

jdbcmysqlacceso

Tables

jugadores

partidas

Views

Stored Procedures

Functions

proyecto

sys

prueba puntuacion dixit\* jugadores jugadores partidas

Limit to 1000 rows

1 • SELECT \* FROM jdbcmysqlacceso.partidas;

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

	id	torneo_id	narrador_id	fecha	resultado
▶	7	NULL	5	2025-12-16	ALGUNOS
	8	NULL	6	2025-12-16	TODOS
	9	NULL	7	2025-12-16	NADIE
	10	NULL	8	2025-12-16	ALGUNOS
	11	NULL	5	2025-12-16	TODOS
*	NULL	NULL	NULL	NULL	NULL

Administration Schemas

Information

Table: jugadores

Columns:

- id int AI PK
- nombre varchar(100)
- email varchar(100)
- puntosTotales int

partidas 1 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 5	19:26:52	use jdbcmysqlacceso	0 row(s) affected	0.000 sec
✓ 6	19:26:52	select * from jugadores LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
✓ 7	19:31:32	SELECT * FROM jdbcmysqlacceso.jugadores LIMIT...	4 row(s) returned	0.000 sec / 0.000 sec
✗ 8	19:31:47	Refresh Recordset	There are pending changes. Please commit or roll...	
✓ 9	19:31:52	SELECT * FROM jdbcmysqlacceso.jugadores LIMIT...	4 row(s) returned	0.000 sec / 0.000 sec
✓ 10	19:33:01	SELECT * FROM jdbcmysqlacceso.partidas LIMIT ...	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

10°C Nublado 19:33 16/12/2025

Comprobamos en la base de datos

```
7 import Models.Partida;
8 import Models.TipoResultado;
9 import Repository.RepositorioJugador;
10 import Repository.RepositorioPartida;
11 import Service.TorneoService;
12
13 public class GestionaTorneo {
14
15     public static void main(String[] args) {
16
17         try {
18             RepositorioJugador repoJugador = new RepositorioJugador();
19             RepositorioPartida repoPartida = new RepositorioPartida();
20             TorneoService servicio = new TorneoService(repoJugador, repoPartida);
21
22             Jugador j1 = new Jugador("JCViejo", "JCViejo@mail.com", "jici");
23             Jugador j2 = new Jugador("Rendon", "Rendon@mail.com", "alcala");
24             Jugador j3 = new Jugador("Fernando", "Fernando@mail.com", "loco");
25             Jugador j4 = new Jugador("Maste", "Maste@mail.com", "noviene");
26
27             servicio.darAltaJugador(j1);
28             servicio.darAltaJugador(j2);
29             servicio.darAltaJugador(j3);
30             servicio.darAltaJugador(j4);
31
32             servicio.crearPartida(new Partida(j1, LocalDate.now(), TipoResultado.ALGUNO);
33             servicio.crearPartida(new Partida(j2, LocalDate.now(), TipoResultado.TODOS);
34             servicio.crearPartida(new Partida(j3, LocalDate.now(), TipoResultado.NADIE);
35             servicio.crearPartida(new Partida(j4, LocalDate.now(), TipoResultado.ALGUNO);
36             servicio.crearPartida(new Partida(j1, LocalDate.now(), TipoResultado.TODOS);
37
38             // Sexta partida -> excepción
39             servicio.crearPartida(new Partida(j2, LocalDate.now(), TipoResultado.ALGUNO);
40
41         } catch (MiExcepcion e) {
42             System.err.println(e.getMessage());
43         }
44     }
45 }
46
```

Console Output:

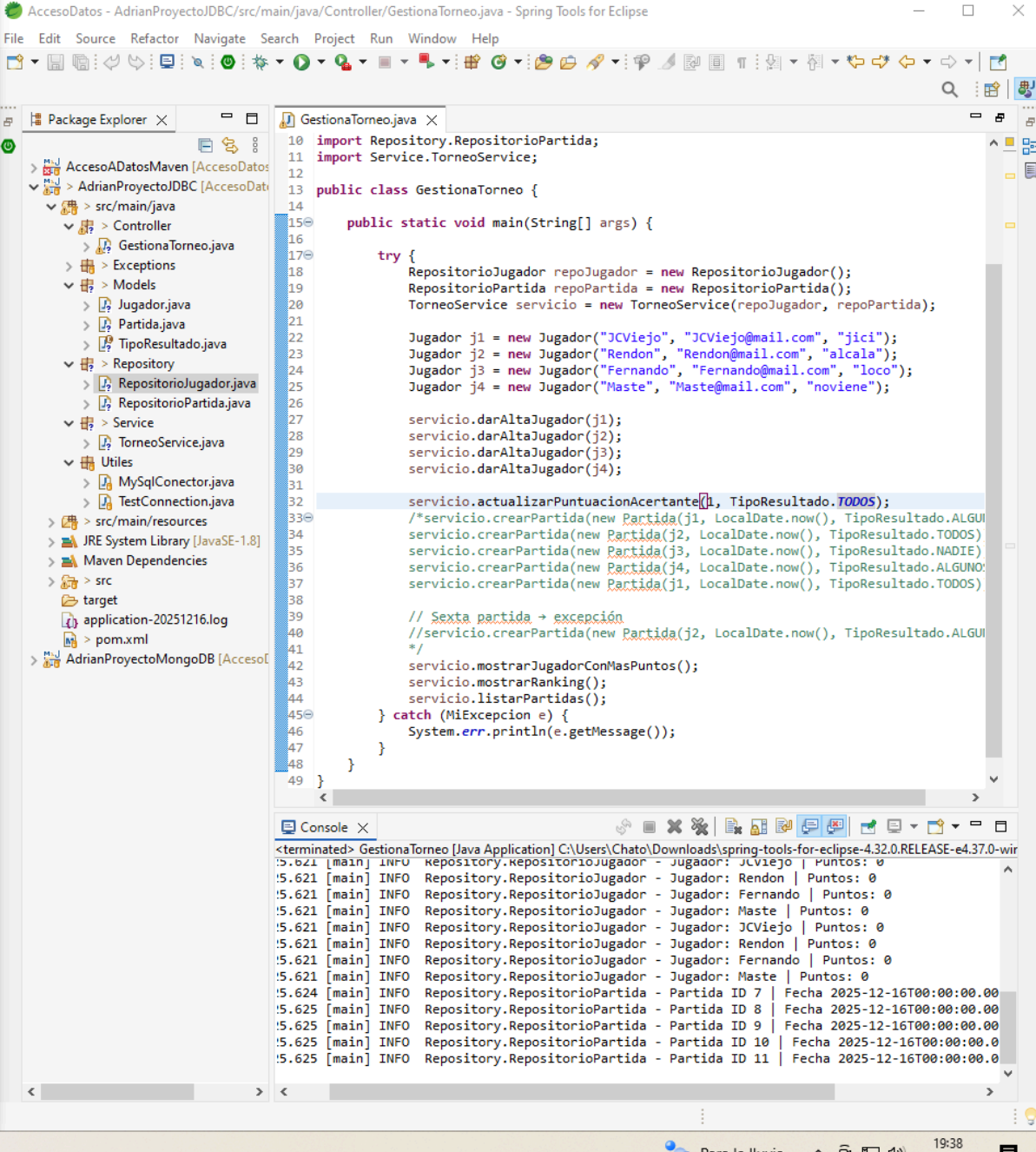
```
<terminated> GestionaTorneo [Java Application] C:\Users\Chato\Downloads\spring-tools-for-eclipse-4.32.0.RELEASE-e4.37.0-wir
19:32:14.660 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: JCViejo
19:32:14.664 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Rendon
19:32:14.666 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Fernando
19:32:14.669 [main] INFO Repository.RepositorioJugador - Jugador dado de alta: Maste
19:32:14.683 [main] INFO Repository.RepositorioPartida - Partida dada de alta
19:32:14.687 [main] INFO Repository.RepositorioPartida - Partida dada de alta
19:32:14.711 [main] INFO Repository.RepositorioPartida - Partida dada de alta
19:32:14.733 [main] INFO Repository.RepositorioPartida - Partida dada de alta
19:32:14.753 [main] INFO Repository.RepositorioPartida - Partida dada de alta
No se pueden crear más de 5 partidas
```

Nos salta la excepción al crear la 6 partida.

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Displays the project structure for 'AccesoDatos'. It includes packages like 'src/main/java', 'Controller', 'Models', 'Repository', 'Service', and 'Utiles'. The 'GestionaTorneo.java' file is selected in the 'Controller' package.
- Editor:** Shows the code for 'GestionaTorneo.java'. The code imports 'Repository.RepositorioPartida' and 'Service.TorneoService'. It defines a 'main' method that:
  - Creates instances of 'RepositorioJugador' and 'RepositorioPartida'.
  - Creates a 'TorneoService' instance with the repositories.
  - Creates four 'Jugador' objects: 'JCViejo', 'Rendon', 'Fernando', and 'Maste'.
  - Calls 'servicio.darAltaJugador()' for each player.
  - Calls 'servicio.actualizarPuntuacionAcertante()' with '1' and 'TiposResultado.TODOS'.
  - Creates four 'Partida' objects and calls 'servicio.crearPartida()' for each.
  - Calls 'servicio.mostrarJugadorConMasPuntos()', 'servicio.mostrarRanking()', and 'servicio.listarPartidas()'.
  - Catches any 'MiExcepcion' and prints the error message.
- Console:** Shows the output of the application. It includes messages from 'Repository.RepositorioJugador' and 'Repository.RepositorioPartida' indicating player creation and ranking. The output shows that 'JCViejo' has 2 points, while the others have 0.

## Info por puntos y por fecha de partidas:



The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows the project structure for 'AdrianProyectoJDBC'. The 'src/main/java' directory contains:
  - Controller: GestionaTorneo.java
  - Exceptions
  - Models: Jugador.java, Partida.java, TipoResultado.java
  - Repository: RepositorioJugador.java, RepositorioPartida.java
  - Service: TorneoService.java
  - Utils: MySqlConnection.java, TestConnection.java
- Main Editor:** Displays 'GestionaTorneo.java'. The code includes imports for 'Repository.RepositorioPartida' and 'Service.TorneoService'. The main method initializes a 'TorneoService' and creates four 'Jugador' objects. It then calls 'servicio.actualizarPuntuacionAcertante(1, TipoResultado.TODOS)' and 'servicio.crearPartida' for several matches. A catch block handles 'MiExcepcion'.
- Console:** Shows the output of the program. It lists player names and scores, followed by a list of matches with IDs and dates.
 

```
<terminated> GestionaTorneo [Java Application] C:\Users\Chato\Downloads\spring-tools-for-eclipse-4.32.0.RELEASE-e4.37.0-wir
:5.621 [main] INFO Repository.RepositorioJugador - Jugador: JC Viejo | Puntos: 0
:5.621 [main] INFO Repository.RepositorioJugador - Jugador: Rendon | Puntos: 0
:5.621 [main] INFO Repository.RepositorioJugador - Jugador: Fernando | Puntos: 0
:5.621 [main] INFO Repository.RepositorioJugador - Jugador: Maste | Puntos: 0
:5.621 [main] INFO Repository.RepositorioJugador - Jugador: JC Viejo | Puntos: 0
:5.621 [main] INFO Repository.RepositorioJugador - Jugador: Rendon | Puntos: 0
:5.621 [main] INFO Repository.RepositorioJugador - Jugador: Fernando | Puntos: 0
:5.621 [main] INFO Repository.RepositorioJugador - Jugador: Maste | Puntos: 0
:5.624 [main] INFO Repository.RepositorioPartida - Partida ID 7 | Fecha 2025-12-16T00:00:00.00
:5.625 [main] INFO Repository.RepositorioPartida - Partida ID 8 | Fecha 2025-12-16T00:00:00.00
:5.625 [main] INFO Repository.RepositorioPartida - Partida ID 9 | Fecha 2025-12-16T00:00:00.00
:5.625 [main] INFO Repository.RepositorioPartida - Partida ID 10 | Fecha 2025-12-16T00:00:00.00
:5.625 [main] INFO Repository.RepositorioPartida - Partida ID 11 | Fecha 2025-12-16T00:00:00.00
```