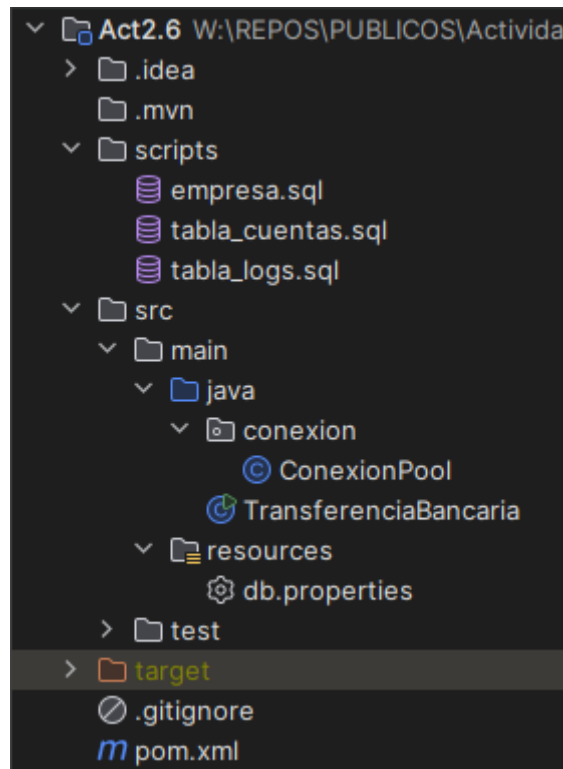


# Actividad 2.6

Iris Pérez Aparicio  
2º DAM  
Campus FP Emprende Humanes

[GITHUB](#)

## ESTRUCTURA DEL PROYECTO



Enlace al código del proyecto:

[https://github.com/IrisCampusFP/ActividadesAccesoADatos/tree/main/UD2-JDBC\\_Avanzado/Act2.6](https://github.com/IrisCampusFP/ActividadesAccesoADatos/tree/main/UD2-JDBC_Avanzado/Act2.6)

1. En la base de datos `empresa` , creo la tabla `cuentas` :

```
CREATE TABLE cuentas (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  titular VARCHAR(100),  
  saldo DECIMAL(10,2)  
);
```

```
INSERT INTO cuentas (titular, saldo) VALUES  
( 'Ana', 2000.00),  
( 'Luis', 1500.00);
```

[https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC\\_Avanzado/Act2.6/scripts/tabla\\_cuentas.sql](https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC_Avanzado/Act2.6/scripts/tabla_cuentas.sql)

2. Implemento en Java una clase `TransferenciaBancaria` que:

Realiza una transferencia de 500€ de la cuenta 1 a la cuenta 2.

Usa transacciones manuales ( `setAutoCommit(false)` ).

Aplica `commit()` si la operación se completa correctamente.

Aplica `rollback()` si ocurre un error.

[https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC\\_Avanzado/Act2.6/src/main/java/TransferenciaBancaria.java](https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC_Avanzado/Act2.6/src/main/java/TransferenciaBancaria.java)

3. Extiendo el ejercicio para que:

Se registre en una tabla `logs` cada paso de la transacción.

Si falla el registro del segundo paso, se use un **savepoint** para mantener el primero.

Creación de la tabla `logs`:

[https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC\\_Avanzado/Act2.6/scripts/tabla\\_logs.sql](https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC_Avanzado/Act2.6/scripts/tabla_logs.sql)

Código de la transacción con los logs:

[https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC\\_Avanzado/Act2.6/src/main/java/TransferenciaBancaria.java](https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC_Avanzado/Act2.6/src/main/java/TransferenciaBancaria.java)

Código sql de la base de datos completo:

[https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC\\_Avanza do/Act2.6/scripts/empresa\\_completo.sql](https://github.com/IrisCampusFP/ActividadesAccesoADatos/blob/main/UD2-JDBC_Avanza do/Act2.6/scripts/empresa_completo.sql)

### ★ Ejecución sin errores:

Estado de las cuentas antes de la operación:

51 • `SELECT * FROM cuentas;`

<

Result Grid | Filter Rows:

	id	titular	saldo
▶	1	Ana	2000.00
	2	Luis	1500.00
*	NULL	NULL	NULL

### EJECUCIÓN:

```
[main] INFO com.zaxxer.hikari.HikariDataSource - ConexionPool - Starting...
[main] INFO com.zaxxer.hikari.pool.HikariPool - ConexionPool - Added connection com.mysql.cj.jdbc.ConnectionImpl@682b2fa
[main] INFO com.zaxxer.hikari.HikariDataSource - ConexionPool - Start completed.
Pool de conexiones HikariCP inicializado correctamente.

Transacción realizada con éxito.

Process finished with exit code 0
```

Estado de las cuentas después de la operación:

51 • `SELECT * FROM cuentas;`

<

Result Grid | Filter Rows:

	id	titular	saldo
▶	1	Ana	1500.00
	2	Luis	2000.00
*	NULL	NULL	NULL

Tabla logs:

52 • `SELECT * FROM logs;`

<

Result Grid | Filter Rows:  Edit:

	id	mensaje	fecha
▶	1	Retirados 500.0 € de la cuenta con id 1	2025-11-16 17:56:03
	2	Ingresados 500.0 € en la cuenta 2	2025-11-16 17:56:03
*	NULL	NULL	NULL

## ★ Ejecuciones con rollback:

### 1. Fuerzo un error para provocar un rollback parcial:

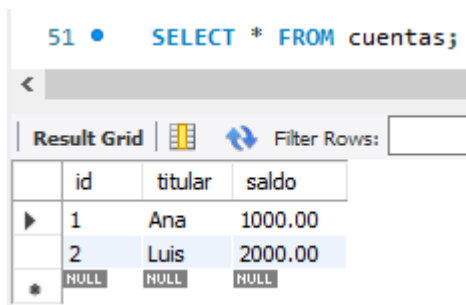
```
[main] INFO com.zaxxer.hikari.HikariDataSource - ConexionPool - Starting...
[main] INFO com.zaxxer.hikari.pool.HikariPool - ConexionPool - Added connection com.mysql.cj.jdbc.ConnectionImpl@ba54932
[main] INFO com.zaxxer.hikari.HikariDataSource - ConexionPool - Start completed.
Pool de conexiones HikariCP inicializado correctamente.

java.sql.SQLException Create breakpoint : Error forzado para probar savepoint
    at TransferenciaBancaria.main(TransferenciaBancaria.java:55)
Fallo en el segundo paso de la transacción (ingreso en cuenta destino)
Se mantiene el primer paso (retirada cuenta origen).

Process finished with exit code 0
```

Estado de las tablas:

- cuentas

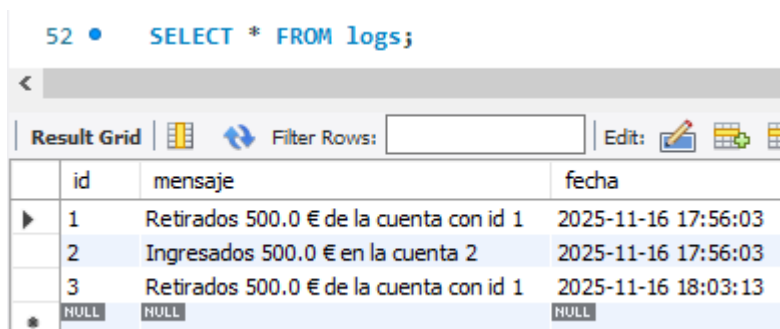


51 • SELECT \* FROM cuentas;

Result Grid

	id	titular	saldo
▶	1	Ana	1000.00
	2	Luis	2000.00
*	NULL	NULL	NULL

- logs



52 • SELECT \* FROM logs;

Result Grid

	id	mensaje	fecha
▶	1	Retirados 500.0 € de la cuenta con id 1	2025-11-16 17:56:03
	2	Ingresados 500.0 € en la cuenta 2	2025-11-16 17:56:03
	3	Retirados 500.0 € de la cuenta con id 1	2025-11-16 18:03:13
*	NULL	NULL	NULL

Como indica el mensaje del error, se revierte la transacción hasta el savepoint manteniendo el primer paso (la retirada y su log).

## 2. Fuerzo un error después del log del segundo paso:

```
[main] INFO com.zaxxer.hikari.HikariDataSource - ConexiónPool - Starting...
[main] INFO com.zaxxer.hikari.pool.HikariPool - ConexiónPool - Added connection com.mysql.cj.jdbc.ConnectionImpl@ba54932
[main] INFO com.zaxxer.hikari.HikariDataSource - ConexiónPool - Start completed.
Pool de conexiones HikariCP inicializado correctamente.

java.sql.SQLException Create breakpoint : Segundo error forzado para probar vuelta al savepoint tras el log del segundo paso
    at TransferenciaBancaria.main(TransferenciaBancaria.java:62)
Fallo en el segundo paso de la transacción (ingreso en cuenta destino)
Se mantiene el primer paso (retirada cuenta origen).

Process finished with exit code 0
```

Estado de las tablas:

- cuentas

51 • `SELECT * FROM cuentas;`

Result Grid

	id	titular	saldo
▶	1	Ana	500.00
	2	Luis	2000.00
*	NULL	NULL	NULL

- logs

52 • `SELECT * FROM logs;`

Result Grid

	id	mensaje	fecha
▶	1	Retirados 500.0 € de la cuenta con id 1	2025-11-16 17:56:03
	2	Ingresados 500.0 € en la cuenta 2	2025-11-16 17:56:03
	3	Retirados 500.0 € de la cuenta con id 1	2025-11-16 18:03:13
	4	Retirados 500.0 € de la cuenta con id 1	2025-11-16 18:08:15
*	NULL	NULL	NULL

De nuevo se ha revertido la transacción hasta el savepoint manteniendo únicamente el primer paso (la retirada y su log).

Código completo del proyecto en [GitHub](#)