

# Transacciones

**DigitalHouse**>  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# ¿Qué son las transacciones?

Una transacción es un conjunto de operaciones sobre una base de datos que se deben ejecutar como una unidad.

# ¿Qué son las transacciones?

Hay ocasiones en las que es necesario que varias operaciones sobre la base de datos se realicen en bloque, es decir, que se ejecuten o todas o ninguna, pero que no que se realicen solo algunas.

Si se ejecutan parcialmente hasta que una da error, el estado de la base de datos puede quedar inconsistente. En este caso necesitaríamos un mecanismo para devolverla a su estado anterior, pudiendo deshacer todas las operaciones realizadas.

## Ejemplo

Supongamos que vamos a reservar un vuelo desde Alicante hasta Sydney.

Para llegar a Sydney tenemos que realizar varias escalas. Tendremos que reservar un vuelo de Alicante a Madrid, otro de Madrid a Dubai y por último, volar desde Dubai a Sydney. Si representamos cada reserva como un insert en la base de datos tendremos las siguientes instrucciones:



```
try {  
    ....  
        s.executeUpdate("INSERT INTO RESERVAS(pasajero, origen, destino)  
VALUES('Felipe', 'Alicante', 'Madrid')");  
        s.executeUpdate("INSERT INTO RESERVAS(pasajero, origen, destino) VALUES  
( 'Felipe', 'Madrid', 'Dubai')");  
        s.executeUpdate("INSERT INTO RESERVAS(pasajero, origen, destino)  
VALUES('Felipe', 'Dubai', 'Sydney')");  
    ....  
} catch(SQLException e) {  
    // Se ha producido un error pero no sabemos donde  
    // ¿Qué operaciones se han realizado? ¿Cómo deshacerlas?  
}
```

Un objeto **Connection** por defecto realiza automáticamente cada operación sobre la base de datos. Esto significa que cada vez que se ejecuta una instrucción, se refleja en la base de datos y no puede ser deshecha. Por defecto está habilitado el **modo auto-commit** en la conexión.

Los siguientes métodos en la interfaz Connection son utilizados para gestionar las transacciones en la base de datos:

```
void setAutoCommit(boolean valor)
void commit()
void rollback()
```

Para iniciar una transacción deshabilitamos el modo auto-commit mediante el método `setAutoCommit(false)`. Esto nos da el control sobre lo que se realiza y cuándo se realiza.

Una llamada al método `commit()` realizará todas las instrucciones emitidas desde la última vez que se invocó el método `commit()`.

Una llamada a `rollback()` deshacerá todos los cambios realizados desde el último `commit()`. Una vez se ha emitido una instrucción `commit()`, esas transacciones no pueden deshacerse con `rollback()`.

DigitalHouse>  
Coding School