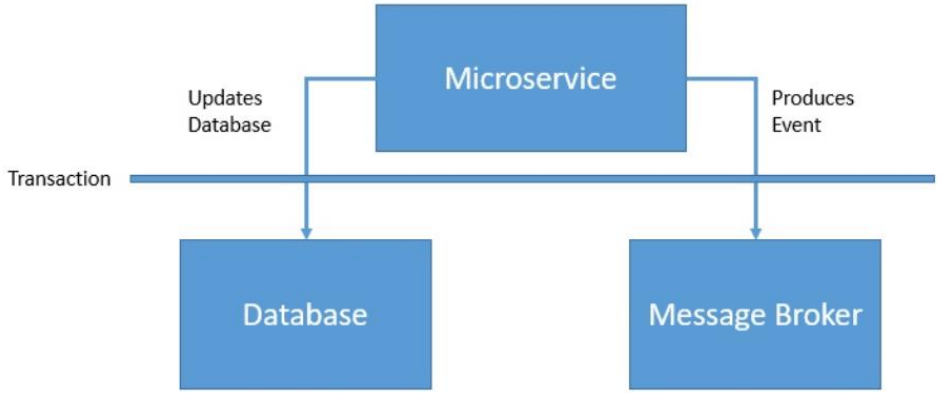
Transaction

Natalia Reyes Altamirano

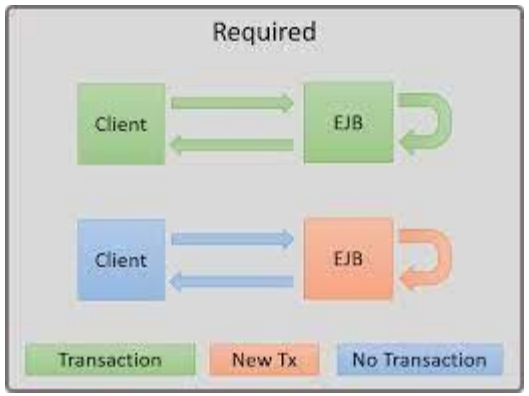
¿Qué es transaction?

- Se refiere a un conjunto de operaciones que se realizan como una unidad atómica e indivisible. En decir, es una secuencia de operaciones que se ejecutan como si fueran una sola operación, y garantiza la integridad y la consistencia de los datos en caso de fallos o problemas durante el proceso.
- En una transacción se ejecutan todas las operaciones o ninguna.



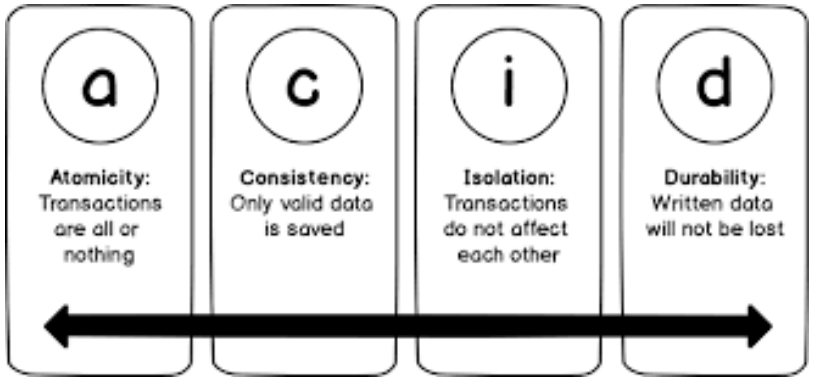
• Ejemplo de cómo funciona una transacción en Java utilizando la API JDBC:

- **Establecer la conexión a la base de datos**: Primero, se establece una conexión a la base de datos utilizando la API JDBC. Esto se hace a través de objetos como Connection que representan la conexión física a la base de datos.
- **Desactivar el modo de autocommit:** Por defecto, la API JDBC está en modo de autocommit, lo que significa que cada instrucción SQL se confirma automáticamente en la base de datos. Para trabajar con transacciones, generalmente se desactiva el autocommit usando connection.setAutoCommit(false).
- **Realizar operaciones dentro de la transacción:** Las operaciones que deben realizarse como parte de la transacción se ejecutan utilizando objetos como Statement o PreparedStatement. Estas operaciones pueden incluir inserciones, actualizaciones y eliminaciones en la base de datos.
- **Confirmar o revertir la transacción:** Una vez que todas las operaciones han sido ejecutadas sin problemas, se llama al método commit() en el objeto Connection para confirmar la transacción y hacer permanentes los cambios en la base de datos. Si ocurre algún error o excepción durante las operaciones, se llama al método rollback() para deshacer todos los cambios realizados en la transacción.
- **Cerrar la conexión**: Después de confirmar o revertir la transacción, es importante cerrar la conexión a la base de datos utilizando el método close() en el objeto Connection.



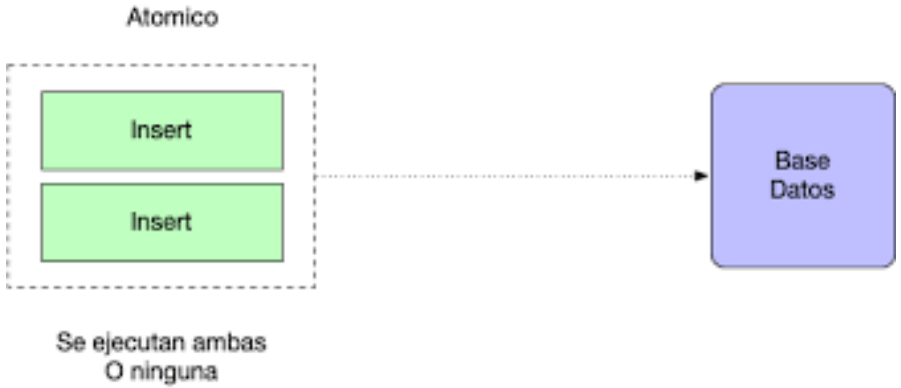
Propiedades ACID de las transacciones

- **Atomicidad (Atomicity):** Todas las operaciones en una transacción se ejecutan como una unidad indivisible. Si una operación falla, todas las operaciones previas se deshacen (rollback) para mantener la consistencia.
- **Consistencia (Consistency):** Una transacción lleva el sistema desde un estado consistente a otro estado consistente. Esto asegura que los datos cumplan con ciertas reglas de validación o integridad durante todo el proceso.
- **Aislamiento (Isolation):** Mientras una transacción está en curso, sus cambios no son visibles para otras transacciones hasta que se complete. Esto previene problemas de concurrencia y asegura que las transacciones no interfieran entre sí.
- **Durabilidad (Durability):** Una vez que una transacción se ha completado con éxito, sus cambios son permanentes y persisten incluso en caso de fallos posteriores del sistema.



Modo de propagación

- El "modo de propagación" en el contexto de transacciones se refiere a cómo una transacción se propaga o se extiende a través de diferentes operaciones o componentes en un sistema. En otras palabras, el modo de propagación define cómo una transacción interactúa con otras transacciones y cómo su alcance se maneja en un entorno multioperacional o multinivel.



- **Algunos de los modos de propagación comunes en el contexto de transacciones incluyen:**

Modo	Si existe una transacción en curso	Si no existe
REQUIRED	Usar	Crear nueva
SUPPORTS	Usar	Nada. El método no será transaccional.
MANDATORY	Usar	Lanzar excepción
REQUIRES_NEW	Suspender y crear una nueva	Crear nueva
NOT_SUPPORTED	Suspender	Nada. El método no será transaccional.
NEVER	Lanzar excepción	Nada. El método no será transaccional.
NESTED	Ejecutar el método en una transacción anidada. Esto significa que una reversión (*rollback)* de la transacción anidada solo afecta a las operaciones ya realizadas. Este modo no funciona en Spring Data JPA.	

La anotación @Transactional

- @Transactional permite definir con una sencillez proverbial el comportamiento de un método de un bean de Spring en lo que respecta a las transacciones.
- Para crear un proceso transaccional no se marca con @Transactional todos los métodos involucrados, tan solo aquel que representa el inicio.