



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 

# ➤ Sentencias para la manipulación de datos y transaccionalidad

---

**Plan formativo:** Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



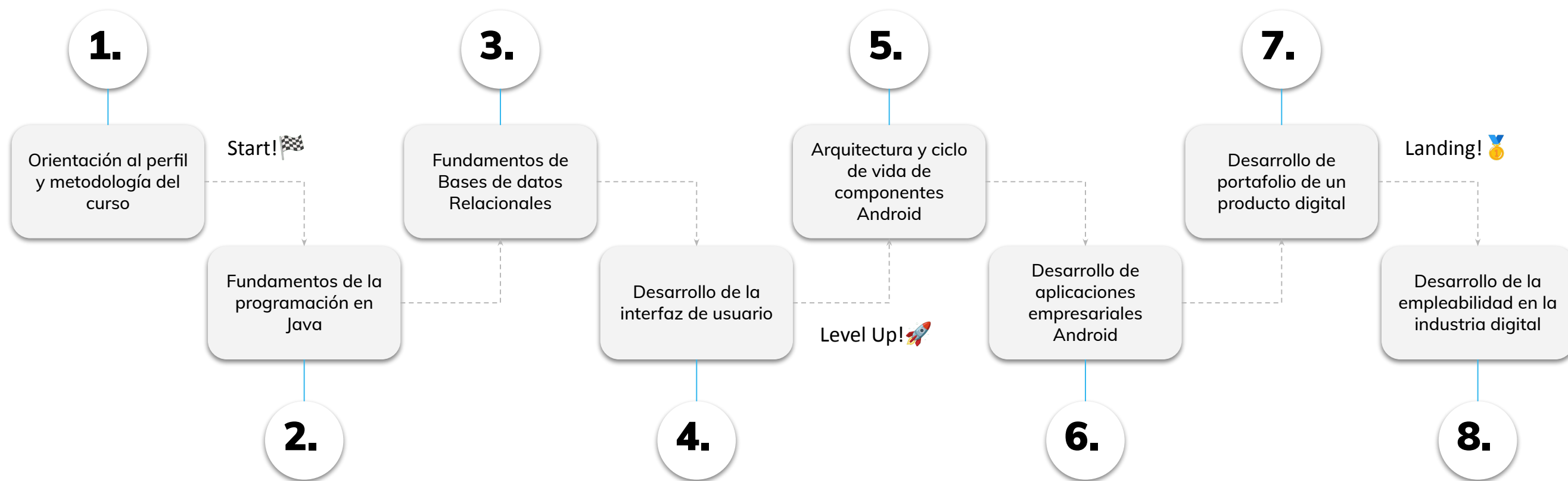
# ➤ Clase 40 - Sentencias para la manipulación de datos y transaccionalidad

---

**Plan formativo:** Desarrollo de Aplicaciones Móviles Android Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



# REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Concepto de *integridad referencial*
- ✓ Cómo eliminar datos con *integridad referencial*
- ✓ Cómo actualizar datos con *integridad referencial*

# LEARNING PATHWAY

3.

Start! 🏁

## Fundamentos de bases de datos relacionales

El objetivo de este tema es brindar una descripción básica de los conceptos asociados a las bases de datos relacionales que te permitirán desde tu ordenador instalar las herramientas necesarias para establecer una conexión a BD y comenzar a manipular dichos datos.

Principios ACID

Transaccionalidad en las operaciones SQL

Simulación de una transferencia de fondos entre dos cuentas bancarias:

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



**Conocer los principios ACID**



**Conocer qué es la transaccionalidad en las operaciones SQL**



**Realizar transacciones para confirmar y revertir cambios en la base de datos**





# › Principios ACID



# Principios ACID



## ¿Qué son las propiedades ACID?:

Los principios ACID son un conjunto de propiedades que se aplican a las transacciones en bases de datos para garantizar la integridad y la coherencia de los datos en un entorno de procesamiento concurrente y en caso de fallos.



ACID es un acrónimo que representa las siguientes propiedades:

- Atomicidad (Atomicity)
- Consistencia (Consistency)
- Aislamiento (Isolation)
- Durabilidad (Durability)





# Principios ACID



## ¿Qué significan las propiedades atomicidad y consistencia?

**Atomicidad (Atomicity):** Una transacción se considera atómica si se trata como una unidad indivisible. Esto significa que todas las operaciones dentro de una transacción se ejecutan en su totalidad o no se ejecutan en absoluto. Si alguna operación dentro de una transacción falla, todas las operaciones anteriores se deshacen (rollback) y los cambios no se aplican.



**Consistencia (Consistency):** Las transacciones deben llevar la base de datos desde un estado válido a otro estado válido. Esto garantiza que la base de datos siempre cumple con ciertas reglas y restricciones de integridad. Si una transacción no puede llevar a cabo una operación que mantiene la consistencia, se cancela.



# Principios ACID



## ¿Qué significan las propiedades aislamiento y durabilidad?

**Aislamiento (Isolation):** Este principio asegura que las transacciones en ejecución sean independientes y no afecten entre sí, incluso cuando se ejecutan concurrentemente. El aislamiento evita que las transacciones en curso interfieran entre sí, lo que puede conducir a problemas como lecturas sucias, lecturas no repetibles, y escrituras fantasma.



**Durabilidad (Durability):** Una vez que una transacción se completa exitosamente, sus cambios se mantienen permanentemente en la base de datos, incluso en caso de fallos posteriores del sistema. Esto garantiza que los datos modificados por una transacción persistan y no se pierdan.

# Principios ACID

## ✕ ¿Para qué se utilizan estos principios?:

Los principios ACID son esenciales en aplicaciones que requieren un alto nivel de confiabilidad, consistencia y precisión en los datos, como sistemas de gestión de bases de datos (DBMS) utilizados en aplicaciones financieras, sistemas de reservas, sistemas de inventario y más

## En resumen:

Los principios ACID son fundamentales para garantizar la integridad y la coherencia de los datos en aplicaciones críticas, especialmente en entornos donde la precisión y la confiabilidad son de suma importancia.



# ➤ Transaccionalidad en las operaciones



# Transaccionalidad en las operaciones



## ¿Qué es una transacción?

La transaccionalidad en el contexto de las operaciones SQL se refiere a la capacidad de agrupar una serie de acciones o consultas en una unidad lógica llamada "transacción", que se ejecuta como una sola entidad indivisible.



## ¿Para que se utilizan?

El propósito principal de las transacciones es garantizar la integridad de los datos, la coherencia y la consistencia en una base de datos, incluso en entornos de concurrencia y en presencia de fallos.





# Transaccionalidad en las operaciones



## Transaccionalidad y principios ACID:

La transaccionalidad en las operaciones SQL se rige por las propiedades ACID (Atomicity, Consistency, Isolation y Durability)

Esto permite ejecutar como una sola unidad operaciones complejas en las que intervienen varias sentencias de la base de datos, lo que proporciona fiabilidad y capacidad de recuperación ante fallos o errores.

Si una transacción falla, puede volver al estado anterior, lo que evita actualizaciones parciales o incoherentes de la base de datos





# Transaccionalidad en las operaciones

✕ Para poner en práctica la transaccionalidad en SQL, generalmente se siguen los siguientes pasos:

## 1. Inicio de la Transacción (BEGIN):

Se inicia una transacción utilizando la sentencia `START TRANSACTION(MySql)`.

A partir de este punto, todas las operaciones que se realicen formarán parte de la transacción actual.

## 2. Ejecución de Operaciones:

Durante la transacción, se ejecutan las operaciones de lectura y escritura en la base de datos como lo harías normalmente con sentencias SQL estándar (`SELECT`, `INSERT`, `UPDATE`, `DELETE`). Todas estas operaciones se consideran parte de la misma transacción hasta que se finalice.



# Transaccionalidad en las operaciones

✕ Para poner en práctica la transaccionalidad en SQL, generalmente se siguen los siguientes pasos:

**3. Confirmación (Commit):** Una vez que todas las operaciones dentro de la transacción se han ejecutado exitosamente y no hay errores, se utiliza la sentencia COMMIT para confirmar la transacción. Esto significa que los cambios realizados en la base de datos durante la transacción se hacen permanentes y se guardan en la base de datos.

**4. Cancelación (Rollback):** Si en algún momento dentro de la transacción ocurre un error o se detecta alguna inconsistencia, se puede usar la sentencia ROLLBACK para cancelar todas las operaciones realizadas en la transacción. Esto devuelve la base de datos a su estado anterior a la transacción.





# Transaccionalidad en las operaciones



## En resumen:

Si una transacción se completa exitosamente, los cambios se confirman; si algo sale mal, se pueden revertir los cambios.

La implementación de transacciones es esencial para aplicaciones que requieren un control estricto sobre los cambios en la base de datos y la gestión de fallos.



# Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

## Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.



# LIVE CODING

Ejemplo en vivo

## Simulación de una transferencia de fondos entre dos cuentas bancarias:

*Deberás crear una transacción que garantice la integridad de los datos en caso de cualquier problema*

- 1. Restar \$200.00 del saldo de la cuenta con ID 1.*
- 2. Añadir \$200.00 al saldo de la cuenta con ID 2.*
- 3. Confirmar la transacción.*
- 4. Cancelar la transacción por saldo insuficiente*

**Tiempo:** 15 minutos

## Slide únicamente para el mentor.

Acá se proveen los datos de la tabla que se necesitan para hacer el live coding

Supongamos que tienes una tabla llamada "Cuentas" con la siguiente estructura:

```
CREATE TABLE Cuentas (  
  ID INT PRIMARY KEY,  
  Saldo DECIMAL(10, 2)  
);
```

```
INSERT INTO Cuentas (ID, Saldo)  
VALUES (1, 1000.00), (2, 1500.00);
```

○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ Definir los principios ACID
- ✓ Reconocer la transaccionalidad en las operaciones SQL
- ✓ Realizar transacciones para confirmar y revertir cambios en la base de datos





# #WorkingTime

Continuemos ejercitando

**¡Antes de cerrar la clase!** Te invitamos a: 📌 📌 📌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. Material 1 (Lectura de la Lección 3: Sentencias para la manipulación de datos y transaccionalidad, páginas 13-15 )
  - a. Material 3 (Ejercicio Ponte a prueba, de la Lección 3: Sentencias para la manipulación de datos y transaccionalidad )
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

# ¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

# Time-out!

🕒 5 min.

