



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 

➤ Consulta a una o varias tablas

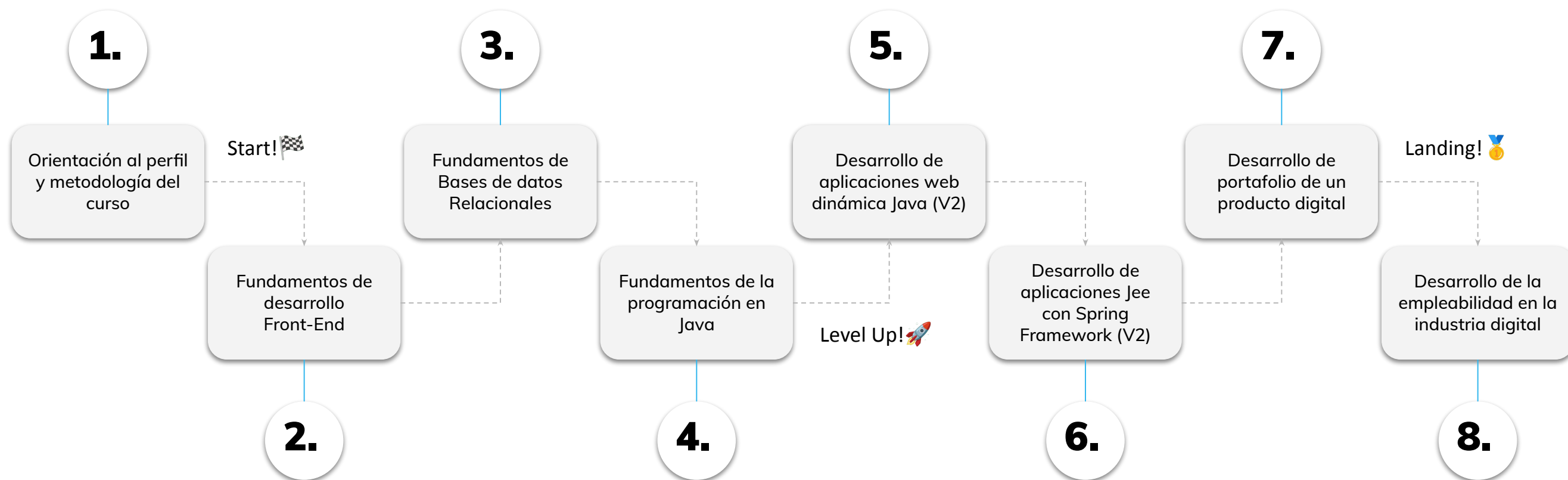
Plan formativo: Desarrollo de Aplicaciones Móviles Android Trainee V2.0

➤ Consulta a una o varias tablas

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

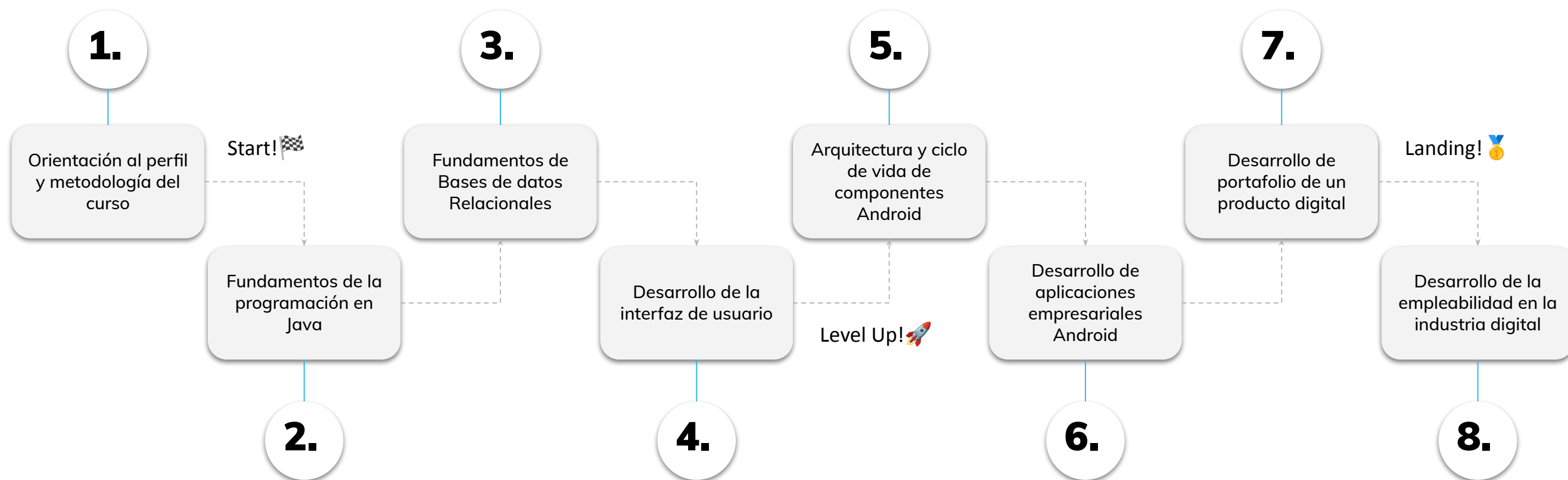
HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



REPASO CLASE ANTERIOR

Clase 35 - Consulta a una o varias
tablas

En la clase anterior trabajamos :

- ✓ Consultas anidadas. ¿Qué son y cómo lucen?
- ✓ Realizar consultas anidadas utilizando SQL

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.

Consultas con funciones
de agrupación

**Escribir consultas
SQL con funciones
de agrupación**

Consultas con diferentes
tipos de JOIN

**Escribir consultas
SQL utilizando JOIN**

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



Realizar consultas con funciones de agrupación utilizando SQL



Realizar consultas con distintos tipos de JOIN





Rompehielo

Contexto:

En este ejercicio, exploraremos cómo usar funciones de agrupación en SQL para calcular el total de ventas por categoría en una tabla de productos.

Consigna:

Escribe una consulta SQL que calcule el total de ventas (suma de precios) por cada categoría de productos.

Imagina que tienes una tabla llamada "productos" con los siguientes campos: id_producto, nombre, categoria y precio.

Paso a paso:

1. Consulta SQL:

```
SELECT categoria, SUM(precio) AS total_ventas  
FROM productos  
GROUP BY categoria;
```

2. Resultado esperado:

Después de ejecutar la consulta, obtendrás una tabla con dos columnas: "categoria" y "total_ventas". Esta tabla mostrará el total de ventas para cada categoría de productos.

3. Preguntas de reflexión: ¿Cómo se agrupan los datos en esta consulta? ¿Qué función de agregación se usa para calcular el total de ventas?

Si quisieras obtener solo las categorías con ventas totales superiores a un cierto valor, ¿cómo podrías modificar la consulta?

Consultas con funciones de agrupación

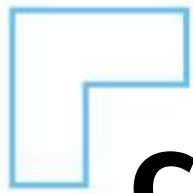
¿Qué son y para qué sirven?

Las consultas con funciones de agregación en SQL son **utilizadas para calcular valores** resumidos o agregados, como sumas, promedios, máximos, mínimos y conteos, sobre un conjunto de datos. Estas funciones permiten realizar cálculos en grupos de registros en lugar de tratar cada registro individualmente.

Si se utilizan junto con la cláusula `GROUP BY` para agrupar los registros según una o varias columnas y luego aplicar las funciones de agregación a cada grupo.

```
-- Contar el número de empleados por departamento
SELECT id_departamento, COUNT(*) AS cantidad_empleados
FROM Empleados
GROUP BY id_departamento;
```

› Consultas con funciones de agrupación



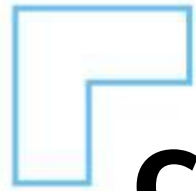
Consultas con funciones de agrupación

Las funciones de agregación más comunes en SQL son:

- **SUM:** Calcula la suma de los valores de una columna numérica en el grupo.
- **AVG:** Calcula el promedio de los valores de una columna numérica en el grupo.
- **COUNT:** Cuenta el número de registros en el grupo. Puede contar todos los registros o solo los registros no nulos en una columna específica.
- **MAX:** Devuelve el valor máximo de una columna en el grupo.
- **MIN:** Devuelve el valor mínimo de una columna en el grupo



Veamos algunos ejemplos en la siguiente diapositiva



Consultas con funciones de agrupación



Algunos ejemplos de consultas con funciones de agregación:

```
-- Calcular la suma de los salarios de todos los empleados
SELECT SUM(salario) AS suma_salarios
FROM Empleados;

-- Calcular el promedio de los salarios por departamento
SELECT id_departamento, AVG(salario) AS promedio_salarios
FROM Empleados
GROUP BY id_departamento;

-- Contar el número de empleados por departamento
SELECT id_departamento, COUNT(*) AS cantidad_empleados
FROM Empleados
GROUP BY id_departamento;

-- Encontrar el salario máximo y mínimo por departamento
SELECT id_departamento, MAX(salario) AS salario_maximo, MIN(salario) AS salario_minimo
FROM Empleados
GROUP BY id_departamento;
```



LIVE CODING

Ejemplo en vivo

Escribir consultas SQL para obtener la siguiente información:

Tienes una base de datos que registra información sobre empleados y departamentos en una empresa. Tienes dos tablas: "Empleados" y "Departamentos".

- 1. Obtener los nombres de los empleados y sus salarios que trabajan en el mismo departamento que el empleado con `id_empleado = 3`.*
- 2. Obtener el nombre del departamento y el nombre del empleado con el salario más alto en cada departamento.*
- 3. Obtener el nombre de los departamentos junto con el número de empleados en cada departamento. Ordenar los resultados por la cantidad de empleados en orden descendente.*
- 4. Obtener el nombre del departamento con el mayor promedio de salarios entre los empleados.*

Slide para el mentor. no mostrar

Estos son los datos de las tablas para el ejercicio anterior. Es importante que el mentor ya tenga las tablas creadas junto con algunos datos previos, para poder recuperar la información en clase

Tabla "Empleados":

id_empleado (Clave primaria)

nombre

apellido

salario

id_departamento (Clave foránea que se relaciona con id_departamento en la tabla Departamentos)

Tabla "Departamentos":

id_departamento (Clave primaria)

nombre_departamento

ubicacion

› Consultas distintos tipos de JOIN



Consultas con distintos tipos de JOIN

¿Qué son las operaciones JOIN?

Las operaciones JOIN en una consulta SQL son utilizadas para combinar datos de dos o más tablas en función de **una relación entre ellas**. Las tablas pueden contener información relacionada, pero dividida en diferentes conjuntos de datos, y las operaciones JOIN permiten unir esos datos para obtener una vista más completa y enriquecida de la información.

Las operaciones JOIN se utilizan principalmente para recuperar información que involucra relaciones entre tablas, como por ejemplo, obtener datos de un cliente junto con sus pedidos o encontrar información sobre estudiantes y los cursos en los que están inscritos.



Consultas con funciones de agrupación

✕ Tipos de operaciones JOIN más utilizados:

INNER JOIN: Combina registros de ambas tablas basados en una condición de coincidencia. Solo los registros que cumplen con la condición se incluyen en el resultado.

LEFT JOIN: Combina todos los registros de la tabla izquierda (primera tabla mencionada) con los registros coincidentes de la tabla derecha (segunda tabla mencionada) según la condición de coincidencia.

RIGHT JOIN : Similar al LEFT JOIN, pero combina todos los registros de la tabla derecha con los registros coincidentes de la tabla izquierda. Si no hay coincidencia, los campos de la tabla izquierda se llenarán con valores NULL.

FULL JOIN: Combina todos los registros de ambas tablas, independientemente de si hay coincidencia o no. Si no hay coincidencia, los campos no coincidentes se llenarán con valores NULL.



Consultas con distintos tipos de JOIN

¡Veamos un ejemplo!

```
SELECT Clientes.nombre, Pedidos.numero_pedido  
FROM Clientes  
INNER JOIN Pedidos ON Clientes.id_cliente = Pedidos.id_cliente;
```



En este ejemplo, estamos combinando los datos de las tablas "Clientes" y "Pedidos" utilizando un **INNER JOIN** basado en la coincidencia de la columna "id_cliente". El resultado será una lista de nombres de clientes junto con los números de pedido correspondientes.

LIVE CODING

Ejemplo en vivo

Escribir consultas SQL para obtener la siguiente información utilizando JOIN:

Tienes una base de datos que almacena información sobre películas y actores en una plataforma de streaming. Tienes dos tablas: "Películas" y "Actores".

- 1. Obtener el título de todas las películas junto con el nombre del actor principal en cada una.*
- 2. Obtener el nombre de los actores y el número de películas en las que han trabajado. Ordenar los resultados por el número de películas en orden descendente.*
- 3. Obtener el título de las películas y el nombre del actor principal en películas en las que el actor tenga el mismo género que la película.*

○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ Realizar consultas con funciones de agrupación diferenciando los distintos métodos
- ✓ Analizar los distintos tipos de JOIN que existen y sus funciones
- ✓ Realizar consultas SQL utilizando JOIN



#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 (Foro)
 - b. Material 2 (Lectura de la Lección 2: Consultas a una o varias tablas en una Base de Datos, páginas 7-9)
 - c. Material 3 (Working time de la Lección 2: Consultas a una o varias tablas en una Base de Datos)
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.

