



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 

# ➤ Sentencias para la definición de tablas

---

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

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



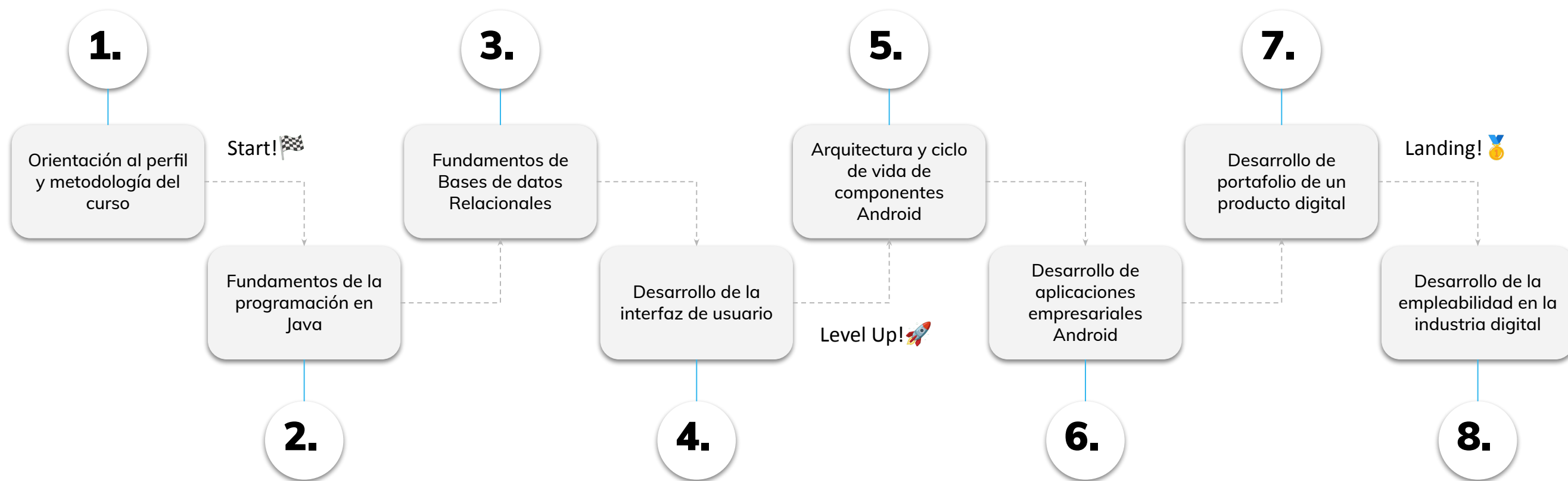
# ➤ Sentencias para la definición de tablas

---

**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 :

- ✓ Comprender el el Lenguaje de Definición de Datos (DDL)
- ✓ Diferenciar los distintos tipos de datos que existen
- ✓ Crear tablas y columnas con diferentes tipos de datos según el requerimiento

# 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.

Crear una tabla en una base de datos utilizando DDL

Diseñar y crear tablas para una tienda online

Restricción de nulidad

Aplicar la restricción de nulidad en el diseño de una tabla

Implementar la restricción de nulidad en una base de datos

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



**Crear y definir campos en una tabla en una base de datos utilizando DDL**



**Conocer el concepto restricción de nulidad**



**Implementar la restricción de nulidad**







# Rompehielo 🧊

## Contexto: 🙌

Imagina que estás construyendo una base de datos para una biblioteca digital que almacena información sobre libros, autores y préstamos y hay que crear una única tabla para todos estos datos en lugar de diseñar tablas separadas



## Consigna: 📝

¿Qué desafíos podrías enfrentar en términos de eficiencia, mantenimiento y consulta de la información? Considera cómo el diseño cuidadoso de las tablas puede facilitar la gestión de datos, mejorar la integridad y simplificar las operaciones de búsqueda en una aplicación tan compleja como una biblioteca digital



Los invitamos a abrir el micrófono 🎤 o escribir en el chat para debatir en conjunto.

➤ Crear una tabla en una base de datos utilizando DDL



# Crear una tabla en una base de datos utilizando DDL



## ¿Cómo crear una tabla?

Crear una tabla en una base de datos utilizando DDL (Lenguaje de Definición de Datos) en SQL implica especificar la estructura de la tabla, incluyendo los nombres de las columnas, los tipos de datos y las restricciones.

### Aquí tienes un ejemplo básico de cómo crear una tabla utilizando DDL:

Supongamos que estamos creando una base de datos para almacenar información sobre libros en una biblioteca.

```
CREATE TABLE Libros (  
    id INT PRIMARY KEY,  
    titulo VARCHAR(100),  
    autor VARCHAR(50),  
    ano_publicacion INT,  
    disponible BOOLEAN  
);
```





# Crear una tabla en una base de datos utilizando DDL



## Analicemos el ejemplo anterior:

Hemos creado una tabla llamada "Libros" con las siguientes columnas:

**id:** Un número entero que servirá como clave primaria para identificar cada libro de manera única.

**título:** Una cadena de caracteres de hasta 100 caracteres para almacenar el título del libro.

**autor:** Una cadena de caracteres de hasta 50 caracteres para almacenar el nombre del autor del libro.

**ano\_publicacion:** Un número entero para almacenar el año de publicación del libro.

**disponible:** Una columna booleana que indica si el libro está disponible en la biblioteca.



# Crear una tabla en una base de datos utilizando DDL

## ✕ Declaracion CREATE:

La declaración CREATE TABLE se utiliza para crear una nueva tabla en la base de datos.

Puedes agregar restricciones adicionales, como claves foráneas, restricciones de integridad, índices, entre otros, dependiendo de tus necesidades.

## En resumen:

La sintaxis pueden variar según el sistema de gestión de bases de datos que estés utilizando.

Recuerda que al crear una tabla, estás definiendo la estructura básica en la que se almacenarán los datos. Es importante elegir los tipos de datos adecuados y diseñar la estructura de manera que sea eficiente para tus necesidades de almacenamiento y consulta.



# LIVE CODING

Ejemplo en vivo

## Diseñar y crear tablas para una tienda online:

*Debes crear tablas para almacenar información sobre productos y categorías de productos, estableciendo relaciones entre ellas.*

- 1. Crear una tabla llamada "Categorias" con 2 columnas: id\_categoria (PK) y nombre\_categoria.*
- 2. Crear una tabla llamada "Productos" con 5 columnas: id\_producto (PK), nombre\_producto, precio, id\_categoria (FK).*
- 3. Insertar algunos datos de ejemplo para probar las relaciones.*

**Tiempo: 15 minutos**

# › Restricción de nulidad



# Restricción de nulidad



## ¿Qué es la restricción de nulidad?:

La restricción de nulidad en bases de datos es una característica que se utiliza para especificar si un valor en una columna de una tabla puede ser nulo (ausente) o si debe contener siempre un valor válido.



Esta restricción ayuda a mantener la integridad y la coherencia de los datos en la base de datos, asegurando que se cumplan ciertas reglas en relación con los valores permitidos en las columnas.







# Restricción de nulidad

## ¿Cómo se puede aplicar?

La restricción de nulidad se puede aplicar en dos modos:

1. **NOT NULL (No nulo):** Al aplicar esta restricción a una columna, se garantiza que dicha columna no pueda contener valores nulos. En otras palabras, cada registro en la tabla debe tener un valor válido en esa columna.

Por ejemplo, si aplicamos la restricción NOT NULL a la columna "Nombre" en una tabla de empleados, estaríamos asegurando que todos los registros tengan un nombre.





# Restricción de nulidad



## ¿Cómo se puede aplicar?

La restricción de nulidad se puede aplicar en dos modos:

**2. NULL (Nulo):** Si no se aplica la restricción NOT NULL a una columna, entonces por defecto la columna permitirá valores nulos. Esto significa que los registros en la tabla podrán tener un valor en esa columna o simplemente estar vacíos (nulos).

En la mayoría de los casos **se realiza al momento de crear la tabla** o al modificar la estructura de la misma utilizando un comando DDL



# Restricción de nulidad

## Ejemplo de restricción NOT NULL:

- ✕ Aquí tienes ejemplos de cómo aplicar la restricción de nulidad en SQL para la creación de una tabla:

```
CREATE TABLE Empleados (  
  id INT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  salario DECIMAL(10, 2)  
);
```

## Analizamos el ejemplo:

En este ejemplo, la columna "nombre" tiene la restricción NOT NULL, lo que significa que todos los registros deben tener un valor en esa columna.





# Restricción de nulidad

## Ejemplo SIN restricción de nulidad (permitiendo valores nulos):

En este ejemplo, las columnas "nombre" y "telefono" no tienen restricciones NOT NULL, lo que permite que los registros en la tabla puedan tener valores nulos en esas columnas.



```
CREATE TABLE Clientes (  
    id INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    telefono VARCHAR(20)  
);
```





## **Ejercicio N°**

# **Aplicar la restricción de nulidad en el diseño de una tabla**



# Aplicar la restricción de nulidad en el diseño de una tabla

## Contexto: 🙌



Imagina que estás diseñando una base de datos para una aplicación de gestión de proyectos en la que los usuarios pueden registrar información sobre tareas asignadas a diferentes miembros del equipo. Algunas tareas pueden tener una fecha de vencimiento y otras no.



## Consigna: 🖋️

¿Cómo crees que la aplicación y los usuarios se verían afectados si no se aplicara la restricción de nulidad adecuadamente en la columna de 'fecha de vencimiento'? ¿Qué problemas podrían surgir si las fechas de vencimiento fueran valores nulos en algunas tareas? Reflexiona sobre cómo la restricción de nulidad puede influir en la confiabilidad, la búsqueda y el análisis de los datos en situaciones del mundo real.

**Tiempo** 🕒: 10 minutos

# 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

## Implementar la restricción de nulidad en una base de datos:

*Estás diseñando una base de datos para gestionar información sobre empleados en una empresa.*

- 1. En la tabla llamada "Usuarios" creada anteriormente crear un nuevo campo llamado "apellido" que sea NOT NULL.*
- 2. inserta algunos datos de ejemplo para trabajar con ellos.*
- 3. Consultar los apellidos de los usuarios.*

Tiempo:



○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ Crear una tabla en una base de datos utilizando DDL
- ✓ Implementar el método CREATE TABLE
- ✓ Diferenciar los dos modos de restricción de nulidad: NOT NULL y NULL



# #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 4: Sentencias para la definición de tablas, páginas 3,7)
  - b. Material 3 (Ejercicio Ponte a prueba de la Lección 4: Sentencias para la definición de tablas)
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.

