



Fase Final

# DESARROLLADOR FULLSTACK CON ENFOQUE EN BACKEND

M Δ S T E R

¡Felicitaciones!  
Tú pasaste al siguiente nivel



**OBJETIVO:** Evaluar tus habilidades de desarrollo en funciones lambda y capacidades en SQL complejos

### INSTRUCCIONES

1. **Esta prueba consta de tres secciones:** Funciones Lambda, Consultas SQL y Gestión de Conexiones a Bases de Datos en funciones Lambda.
2. Envía tu solución en un repositorio de GitHub o similar, incluyendo todo el código fuente, scripts SQL y cualquier instrucción adicional para ejecutar tu solución.

## Sección 1: Funciones Lambda

### Tarea 1: Filtrar y Transformar Datos

- **Descripción:** Dado objeto con información de productos en una tienda, escribe una función lambda que filtre los productos cuyo precio sea mayor a \$10.000 y que devuelva una lista con los nombres de esos productos en mayúsculas.

### Tarea 2: Ordenar Datos

- **Descripción:** Escribe una función lambda para ordenar una lista de tuplas (nombre, edad) en orden descendente por edad.

## Sección 2: Consultas SQL

### Tarea 1: Consultas de Agregación

- **Descripción:** Dada la siguiente estructura de tablas en una base de datos de ventas, escribe una consulta SQL que devuelva el total de ventas por categoría de producto y por mes, junto con el porcentaje de crecimiento respecto al mes anterior.

Unset

```
CREATE TABLE product(  
  id INT PRIMARY KEY,  
  name VARCHAR(100),  
  category VARCHAR(100)  
);
```

```
CREATE TABLE sale (  
  id INT PRIMARY KEY,  
  product_id INT,  
  date DATE,
```



```
quantity INT,  
price DECIMAL(10, 2),  
FOREIGN KEY (product_id ) REFERENCES product(id)  
);
```

#### Instrucciones:

1. La consulta debe agrupar las ventas por categoría de producto y por mes.
2. Debe calcular el total de ventas (en términos monetarios) por cada categoría y mes.
3. Debe incluir una columna adicional que muestre el porcentaje de crecimiento del total de ventas respecto al mes anterior para cada categoría.

**Output Esperado:** Deberías obtener un resultado con columnas: `category`, `month`, `total_sales`, `growth_percentage`.

### Sección 3: Conexiones a Bases de Datos con Node.js en AWS Lambda

#### Tarea 1: Conexión a una Base de Datos MySQL desde AWS Lambda

- **Descripción:** Crea una función Lambda en Node.js que se conecte a una base de datos MySQL, realice varias consultas y devuelva los resultados combinados como una respuesta JSON. La lógica de acceso a la base de datos debe estar implementada en un archivo separado del archivo principal de la función Lambda.
- **Requisitos:**
  1. La función debe utilizar la biblioteca `mysql2` para manejar la conexión a MySQL.
  2. La configuración de la conexión (host, user, password, database, etc.) debe estar almacenada en AWS Secrets Manager.
  3. Crea un archivo `db.js` que maneje la conexión y las consultas a la base de datos.
  4. La función principal en `index.js` debe realizar las siguientes consultas a través de las funciones definidas en `db.js`:
    - Obtener todos los productos.
    - Obtener la cantidad total de ventas por producto.
    - Obtener el producto con el precio más alto.
  5. Combina los resultados de las consultas en un solo objeto JSON y devuélvelo.