

Instrucciones:

- Puedes usar cualquier editor de código y herramientas que consideres necesarias.
- Una vez finalizada, debe hacerse llegar en un github o gitlab público de manera que se pueda clonar, instalar y arrancar el proyecto para verificar el funcionamiento.
- Si se opta por hacer la parte de Java, realizar en otro repositorio y hacer llegar el enlace.

Parte 1: Desarrollo web

Desarrolla una página web simple que muestre un listado de productos y permita filtrar por nombre. Se tendrá en cuenta el diseño y la funcionalidad.

Requisitos:

- Usa **HTML, CSS (SCSS o Bootstrap también son válidos) y JavaScript Vanilla**
- Agrega un campo de búsqueda que filtre los productos en tiempo real.
- Usa datos ficticios de al menos **10 productos**.
- Crea un listado donde se vayan a mostrar los productos con el SKU, Nombre y Precio
- Crea una sola función que permita obtener los productos, filtrar por nombre o precio y además que permita añadir o eliminar productos del listado.

Parte 2: SQL (entregar archivo .sql con las consultas)

Teniendo en cuenta la(s) siguiente(s) consulta realizar:

- Escribe una consulta SQL que obtenga todos los productos cuyo precio sea mayor a 50€ y que tenga stock disponible (tabla productos24).
- Crea una nueva consulta SQL donde se obtenga todos los productos de la tabla productos24 y productos25 donde el precio sea menor a 100€, tenga stock disponible y además la fecha de actualización sea superior al 29/02/2024
- Crea una consulta SQL que actualice el precio de un producto de la tabla productos25 en base a su **id**

```
CREATE TABLE productos24 (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100) NOT NULL, partnumber VARCHAR(100) NULL,  
  precio DECIMAL(10,2) NOT NULL,  
  stock INT NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
  CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE productos25 (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(100) NOT NULL,  
  precio DECIMAL(10,2) NOT NULL,  
  stock INT NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
  CURRENT_TIMESTAMP  
);
```

Parte 3: Java (Opcional y valorado positivamente):

Crea una aplicación Java que lea un archivo CSV con datos de productos y los almacene en una base de datos MySQL.

Requisitos:

- El archivo CSV contendrá columnas: **SKU, Nombre, Precio**.
- Usa JDBC para la conexión con MySQL.
- El nombre de la base de datos se llamará "aseuropa_java"
- Implementa una función que liste los productos almacenados en la base de datos.

Entrega:

- Github o Gitlab público y archivo CSV de prueba