

EJERCICIO U8_B8_E1:

- Con tanto thenComparing a mi me gusta más definir el comparador a parte, si sólo se ordenara por un campo quizá prefiriera añadir el comparing directamente en los paréntesis del sorted().
- En cualquier caso la solución no funcional del enunciado me tiene un aire anticuado. Hay mucho if y mucho paréntesis. Si hay un error en un signo por ejemplo en lugar de != ponemos por error == puede llevar tiempo encontrar el error.
- En fin, como siempre, todo depende del contexto y de las preferencias personales.

```
import java.util.*;

class Producto {
    private String descripcion;
    private int cantidad;
    private double precio;

    public Producto(String descripcion, int cantidad, double precio) {
        this.descripcion = descripcion;
        this.cantidad = cantidad;
        this.precio = precio;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public int getCantidad() {
        return cantidad;
    }

    public double getPrecio() {
        return precio;
    }

    @Override
    public String toString() {
        return cantidad + " - " + precio + " - " + descripcion;
    }
}

class App{
```

```
public static void main(String[] args) {  
    // Creamos una lista de productos desordenada  
    List<Producto> lista = Arrays.asList(  
        new Producto("Producto A", 10, 50.0),  
        new Producto("Producto B", 5, 100.0),  
        new Producto("Producto C", 20, 10.0),  
        new Producto("Producto A", 5, 20.0),  
        new Producto("Producto B", 5, 10.0),  
        new Producto("Producto D", 5, 10.0)  
    );  
  
    // Creamos un comparador para ordenar primero por cantidad, luego por  
    // precio y finalmente por descripción  
    Comparator<Producto> comparador =  
        Comparator.comparing(Producto::getCantidad)  
            .thenComparing(Producto::getPrecio)  
            .thenComparing(Producto::getDescripcion);  
  
    // Ordenamos e imprimimos la lista  
    lista.stream()  
        .sorted(comparador)  
        .forEach(System.out::println);  
}  
}
```