Guia #1.5

Programacion Funcional Avanzada

Contexto

Se trabajará con una lista de productos en un sistema de e-commerce. Cada producto tiene los siguientes atributos:

```
public class Producto {
    private String nombre;
    private double precio;
    private String categoria;
    private int stock;

    // Constructor, Getters y Setters
    public Producto(String nombre, double precio, String categoria, int
stock) {
        this.nombre = nombre;
        this.precio = precio;
        this.categoria = categoria;
        this.stock = stock;
    }

    //Generar Getters, Setters y metodo toString.
}
```

A partir de una lista de productos, se deberá realizar una serie de operaciones utilizando **Streams, Optionals y Lambdas**.

Carga de Datos de Prueba

Para facilitar las pruebas, se proporciona un método estático que carga una lista de 25 productos:

```
public static List<Producto> cargarProductos() {
    return List.of(
        new Producto("Laptop", 1500, "Electrónica", 5),
        new Producto("Smartphone", 800, "Electrónica", 10),
        new Producto("Televisor", 1200, "Electrónica", 3),
        new Producto("Heladera", 2000, "Hogar", 2),
        new Producto("Microondas", 500, "Hogar", 8),
        new Producto("Silla", 150, "Muebles", 12),
        new Producto("Mesa", 300, "Muebles", 7),
        new Producto("Zapatillas", 100, "Deportes", 15),
        new Producto("Pelota", 50, "Deportes", 20),
        new Producto("Bicicleta", 500, "Deportes", 5),
        new Producto("Libro", 30, "Librería", 50),
        new Producto("Cuaderno", 10, "Librería", 100),
        new Producto("Lámpara", 80, "Hogar", 30),
        new Producto("Cafetera", 250, "Hogar", 6),
        new Producto("Auriculares", 120, "Electrónica", 14),
        new Producto("Teclado", 90, "Electrónica", 9),
        new Producto("Mouse", 60, "Electrónica", 18),
        new Producto("Monitor", 700, "Electrónica", 4),
        new Producto("Cama", 800, "Muebles", 2),
        new Producto("Sofá", 1000, "Muebles", 3),
        new Producto("Espejo", 120, "Hogar", 12),
        new Producto("Ventilador", 150, "Hogar", 7),
        new Producto("Patines", 180, "Deportes", 5),
        new Producto("Raqueta", 220, "Deportes", 6),
        new Producto("Taza", 15, "Hogar", 40)
    );
}
```

Ejercicios

1. **Filtrado y Transformación**
    - Obtener una lista con los nombres y precios de los productos de la categoría "Electrónica" cuyo precio sea mayor a 1000, ordenados de mayor a menor precio.
2. **Reducción de Datos**
    - Calcular el precio promedio de los productos de la categoría "Hogar", pero solo considerando aquellos con stock disponible.
3. **Producto mas caro**

    - Obtener el producto más caro de cada categoría y devolver un mapa con la categoría como clave y el producto más caro como valor.
4. **Uso de Optionals**
    - Encontrar el producto de la categoría "Deportes" con stock mayor a 10 unidades, obtener su nombre en minúsculas y devolverlo dentro de un Optional. Mostrarlo o si no existe, mostrar *“Producto Inexistente”*
5. **Producto Mas Barato**
    - Encontrar el producto mas barato calculando el valor total de todas las unidades en stock (Precio * stock). Devolver un Opcional con el producto. En caso de que no exista, lanzar una excepción.
6. **Productos en stock ordenados alfabéticamente**
    - Obtener una lista con los nombres de los productos que tienen stock, ordenarlos alfabéticamente y excluir los productos con nombres de menos de 5 caracteres.
7. **Calculo de Stock Total**
    - Obtener el total de unidades en stock de todos los productos, pero solo considerando aquellos con precio superior al promedio.
8. **Stock por Categoría**
    - Calcular cuántas unidades en stock hay por categoría, excluyendo categorías con menos de 3 productos.
9. **Aplicar descuento**
    - Crear una nueva lista de productos con un 15% de descuento en su precio, pero solo si el producto tiene más de 20 unidades en stock.
10. **Ganancia total inventario**
    - Calcular la ganancia total si se vendiera todo el inventario, considerando que el costo de compra de cada producto es un 45% del valor de venta o de un 65% si pertenece a la categoria Electronica.

Consideraciones

- Utilizar `map`, `filter`, `collect`, `reduce`, `Optional`, `sorted`, `groupingBy` y `max/min` donde sea adecuado. Investigar metodos .stream, ya que algunos ejercicios se pueden resolver utilizando operaciones que no hemos visto
- No utilizar bucles tradicionales (`for` o `while`), sino abordar el problema con **Streams y Lambdas**.
- Usar la lista de 25 productos provista para realizar pruebas.