

## Lista de Compras Avanzada con Precios

Extender la consigna 1 creando un sistema que lee productos con precios y cantidades desde un CSV, y los clasifica por categoría usando enums.  
Archivo de entrada que debes crear (productos.csv):

```
nombre,categoria,precio,cantidad
Manzanas,FRUTAS,150.50,5
Leche,LACTEOS,280.00,2
Pan,PANADERIA,120.75,3
Huevos,LACTEOS,350.00,1
Queso,LACTEOS,450.25,2
Arroz,GRANOS,180.00,2
Pollo,CARNES,890.50,1
Tomates,VERDURAS,220.30,3
Yogurt,LACTEOS,320.00,4
Fideos,GRANOS,95.75,5
```

Crear un enum Categoría con:

```
FRUTAS, VERDURAS, LACTEOS, CARNES, PANADERIA, GRANOS
```

Leer el CSV línea por línea (saltando la primera línea de headers)

Para cada producto calcular el subtotal (precio × cantidad)

Agrupar productos por categoría y calcular totales por categoría

Generar archivo resumen\_compra.txt con:

Lista completa de productos con subtotales

Total de productos por categoría

Total de dinero por categoría

Total general de la compra Categoría más cara y más barata

### AYUDAS:::

Saltar la primera línea del archivo (readLine()) porque tiene los encabezados y no representa un producto real.

Usar split(",") para dividir cada línea en campos separados por comas. (metodo de string)

Aplicar trim() a cada campo para eliminar espacios innecesarios que pueden causar errores.

Usar Double.parseDouble() y Integer.parseInt() para convertir los valores de texto en números (precio y cantidad).

Utilizar Categoría.valueOf() para transformar el texto leído en una constante del enum Categoría.

Guardar los totales por categoría en estructuras Map<Categoría, Integer> y Map<Categoría, Double> para contar productos y acumular subtotales.

getOrDefault() permite obtener un valor del mapa o usar uno por defecto si la clave aún no existe.

Para mostrar precios con dos decimales, usar String.format("%.2f", valor).