

DEPARTAMENTO:	CIENCIAS DE LA COMPUTACION	CARRERA:	Tecnologías de la Información y comunicación		
ASIGNATURA:	PROGRAMACION ORIENTADA A OBJETOS	NIVEL:	2	FECHA:	14/05/2025
DOCENTE:	Ing. Jhon Cruz	PRÁCTICA N°:		CALIFICACIÓN:	

## Archivos CSV

Corina Alejandra Acosta Oliva

### ¿Qué es CSV?

Un archivo CSV (*Comma Separated Values*) es básicamente un archivo de texto plano donde los datos están separados por comas. Es como una tabla de Excel pero en versión simple: cada línea es un registro, y cada valor está separado por comas. Lo bueno es que lo podemos abrir con cualquier editor o programa y es fácil de leer y escribir desde código. Ideal para guardar listas de cosas como productos, usuarios, notas, etc.

### Leer y escribir objetos Producto en archivos CSV.

Manejar excepciones de E/S; Medir el rendimiento de lectura/escritura; Aplicar principios de código limpio; Revisar nombres, encapsulamiento, extracción de métodos, etc.

Crea dos clases: Producto y Categoría. Aplica encapsulamiento completo en ambas clases. Define constructores parametrizados que obliguen a inicializar todos los atributos. Establece una relación de composición: un producto debe pertenecer a una categoría.

#### Incluye validaciones dentro de los setters:

- El precio debe ser mayor que 0.
- El nombre no puede estar vacío.

#### Atributos sugeridos:

- Producto: código, nombre, precio, categoría (objeto).
- Categoría: id, nombre.

#### Objetivos de aprendizaje:

Modelar relaciones entre clases reales. Aplicar encapsulamiento y constructores eficientemente. Reforzar buenas prácticas en validaciones. Reto adicional: Implementa un método `mostrarResumen()` en Producto que imprima los datos completos del producto incluyendo el nombre de su categoría.

#### Código:

##### Main

```
/*
 * 2 clases de producto y categoria
 */

package miespe.p1supermercado;

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
```

```
/**
 *
 * @author espe
 */
public class P1Supermercado {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Map<String, Categoria> categorias = new HashMap<>();
        int opcion = 0;

        do {
            System.out.println("\n=== MENU SUPERMERCADO ===");
            System.out.println("1. Agregar categoria");
            System.out.println("2. Agregar producto a una categoria");
            System.out.println("3. Ver categorias y productos");
            System.out.println("4. Salir");

            // Validar solo que la opcion del menu sea numero entero
            while (true) {
                System.out.print("Elige una opcion: ");
                String input = scanner.nextLine();
                try {
                    opcion = Integer.parseInt(input);
                    break;
                } catch (NumberFormatException e) {
                    System.out.println("Por favor ingresa un numero valido.");
                }
            }

            switch (opcion) {
                case 1:
                    System.out.print("Ingresa el nombre de la categoria: ");
                    String nombreCat = scanner.nextLine().trim().toLowerCase();

                    if (nombreCat.isEmpty()) {
                        System.out.println("La categoria no puede estar vacia.");
                    } else if (categorias.containsKey(nombreCat)) {
                        System.out.println("Esa categoria ya existe.");
                    } else {
                        categorias.put(nombreCat, new Categoria(nombreCat));
                        System.out.println("Categoria agregada correctamente.");
                    }
                    break;

                case 2:
                    if (categorias.isEmpty()) {
                        System.out.println("Primero debes agregar una categoria.");
                        break;
                    }

                    System.out.println("Categorias disponibles:");
                    for (String cat : categorias.keySet()) {
                        System.out.println("- " + cat);
                    }

                    System.out.print("A que categoria quieres agregar el producto? ");
                    String catElegida = scanner.nextLine().trim().toLowerCase();

                    if (!categorias.containsKey(catElegida)) {
                        System.out.println("Esa categoria no existe.");
                        break;
                    }

                    System.out.print("Nombre del producto: ");
                    String nombreProd = scanner.nextLine().trim();

                    if (nombreProd.isEmpty()) {
                        System.out.println("El nombre no puede estar vacio.");
                        break;
                    }
                }
            }
        }
```

```
// Pedimos el precio (permitimos fallos si mete letras)
System.out.print("Precio del producto (usa coma si tiene decimales): ");
float precioProd = scanner.nextFloat();
scanner.nextLine(); // limpiar buffer

if (precioProd <= 0) {
    System.out.println("El precio debe ser mayor que 0.");
    break;
}

Producto nuevoProd = new Producto(nombreProd, precioProd);
categorias.get(catElegida).agregarProducto(nuevoProd);
System.out.println("Producto agregado correctamente.");
break;

case 3:
    if (categorias.isEmpty()) {
        System.out.println("No hay categorias registradas.");
    } else {
        for (Categoria cat : categorias.values()) {
            System.out.println(cat);
        }
    }
    break;

case 4:
    System.out.println("Saliendo del sistema... 🚪");
    break;

default:
    System.out.println("Opcion invalida.");
}

} while (opcion != 4);
}
```

### Clase Categoría

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package miespe.p1supermercado;

/**
 *
 * @author espe
 */
import java.util.ArrayList;
import java.util.List;

public class Categoria {
    private String nombre;
    private List<Producto> productos;

    public Categoria(String nombre) {
        this.nombre = nombre.toLowerCase().trim();
        this.productos = new ArrayList<>();
    }

    public String getNombre() {
        return nombre;
    }

    public List<Producto> getProductos() {
        return productos;
    }

    public void agregarProducto(Producto producto) {
        productos.add(producto);
    }
}
```

```
}

@Override
public String toString() {
    StringBuilder sb = new StringBuilder("Categoria: " + nombre + "\n");
    if (productos.isEmpty()) {
        sb.append(" (sin productos)");
    } else {
        for (Producto p : productos) {
            sb.append(" - ").append(p).append("\n");
        }
    }
    return sb.toString();
}
}
```

### Clase Producto

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package miespe.p1supermercado;

/**
 *
 * @author espe
 */
public class Producto {
    private String nombre;
    private float precio;

    public Producto(String nombre, float precio) {
        this.nombre = nombre;
        this.precio = precio;
    }

    public String getNombre() {
        return nombre;
    }

    public float getPrecio() {
        return precio;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public void setPrecio(float precio) {
        this.precio = precio;
    }

    @Override
    public String toString() {
        return nombre + " - $" + precio;
    }
}
```

EJECUCION:

```
=== MENU SUPERMERCADO ===
1. Agregar categoria
2. Agregar producto a una categoria
3. Ver categorias y productos
4. Salir
Elige una opcion: 1
Ingresa el nombre de la categoria: Lacteos
Categoria agregada correctamente.
```

```
=== MENU SUPERMERCADO ===
1. Agregar categoria
2. Agregar producto a una categoria
3. Ver categorias y productos
4. Salir
Elige una opcion: 1
Ingresa el nombre de la categoria: Cereales
Categoria agregada correctamente.
```

```
=== MENU SUPERMERCADO ===
1. Agregar categoria
2. Agregar producto a una categoria
3. Ver categorias y productos
4. Salir
Elige una opcion: 11
Opcion invalida.
```

```
=== MENU SUPERMERCADO ===
1. Agregar categoria
2. Agregar producto a una categoria
3. Ver categorias y productos
4. Salir
Elige una opcion: 1
Ingresa el nombre de la categoria: Bebida
Categoria agregada correctamente.
```

```
=== MENU SUPERMERCADO ===
1. Agregar categoria
2. Agregar producto a una categoria
3. Ver categorias y productos
4. Salir
Elige una opcion: 2
Categorias disponibles:
- cereales
- lacteos
- bebida
```

A que categoria quieres agregar el producto? |

```
Arroz - $2.1
Categoria: lacteos
- Queso - $2.4
```

Categoria: bebida  
(sin productos)

```
=== MENU SUPERMERCADO ===
1. Agregar categoria
2. Agregar producto a una categoria
3. Ver categorias y productos
4. Salir
Elige una opcion: bebida
Por favor ingresa un numero valido.
Elige una opcion: 2
Categorias disponibles:
- cereales
- lacteos
- bebida
A que categoria quieres agregar el producto? bebida
Nombre del producto: coca cola
Precio del producto (usa coma si tiene decimales): 1
Producto agregado correctamente.
```

```
=== MENU SUPERMERCADO ===
1. Agregar categoria
2. Agregar producto a una categoria
3. Ver categorias y productos
4. Salir
Elige una opcion: 3
Categoria: cereales
- Arroz - $2.1
```

Categoria: lacteos  
- Queso - \$2.4

Categoria: bebida  
- coca cola - \$1.0

```
=== MENU SUPERMERCADO ===
1. Agregar categoria
2. Agregar producto a una categoria
3. Ver categorias y productos
4. Salir
Elige una opcion: |
```