

# Ejercicio de Programación Orientada a Objetos: Inventario de Productos

## Contexto

Una tienda desea desarrollar una aplicación básica para administrar su inventario. Para ello, se utilizarán clases en C++ que representen productos y un inventario. Se debe trabajar con un número fijo de productos, almacenados en un arreglo dentro de la clase `Inventario`.

Cada producto debe tener:

- Un nombre (sin espacios) — tipo `string`
- Un código numérico — tipo `int`
- Un precio — tipo `double`
- Una cantidad en existencia — tipo `int`

## Objetivo

Este ejercicio tiene como propósito reforzar los siguientes aspectos:

- Definición y uso de clases en C++
- Manejo de arreglos de objetos
- Modularización y encapsulamiento
- Uso correcto de `#ifndef`, `#define` y `#endif` en archivos de cabecera

## Instrucciones

1. Crea una clase `Producto` en el archivo `Producto.h`, con los siguientes atributos privados:

- `string nombre;`
- `int codigo;`
- `double precio;`

- `int cantidad;`

Usa las siguientes directivas de preprocesador para evitar inclusiones múltiples:

```
#ifndef PRODUCTO_H
#define PRODUCTO_H
// declaracion de la clase aqui
#endif
```

2. Implementa:

- Un constructor por defecto.
- Un constructor que reciba todos los parámetros.
- Métodos `set` para modificar los atributos.
- Un método `toString()` que retorne los datos del producto en una sola cadena.

3. Crea una clase `Inventario` en su archivo `Inventario.h`. También debe tener las directivas:

```
#ifndef INVENTARIO_H
#define INVENTARIO_H
const int MAX_PRODUCTOS = 5;
// declaracion de la clase aqui
#endif
```

4. Dentro de `Inventario`, define un arreglo de productos:

```
Producto productos[MAX_PRODUCTOS];
```

5. Implementa los siguientes métodos en la clase `Inventario`:

- `void registrarProductos();`  
Pide al usuario que ingrese los datos de cada producto. Usa `cin` para todos los valores. No se deben usar espacios en los nombres.
- `void mostrarProductos();`  
Muestra todos los productos utilizando `toString()`.
- `Producto productoMasCaro();`  
Retorna el producto con el mayor precio.
- `Producto productoMasBarato();`  
Retorna el producto con el menor precio.
- `double calcularValorTotal();`  
Suma `precio * cantidad` de todos los productos y retorna el valor total del inventario.

6. Implementa una función `main()` que permita ejecutar el programa, llamando a los métodos anteriores.

## Recomendaciones

- No se deben ingresar espacios en los nombres de productos.
- Utiliza correctamente los nombres de archivos y las directivas de inclusión.
- Puedes usar `std::to_string()` dentro del método `toString()` para convertir números a texto.
- Revisa bien los índices al recorrer arreglos. No sobrepases los límites.