



**UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA CAMPUS
JUTIAPA, JUTIAPA
FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN
CURSO: PROGRA II**

EJERCICIO EN CLASE NO. 4

**JUAN LUIS MOLINA ORELLANA
CARLO RENE HERMOGENES RIVERA ESTRADA
02/08/2025**

Mejora #1

Ubicación: `Producto.java`

Descripción del cambio:

- Se implementaron validaciones en el constructor para asegurar que el ID, nombre, precio y stock cumplan con requisitos mínimos (valores positivos, nombre no vacío).
- Se corrigió el método `hayStock()` para usar `>=` en lugar de `>`.
- Se mejoró el método `reducirStock()` con validaciones para evitar stock negativo.

Justificación:

Estos cambios previenen estados inconsistentes en los productos, asegurando que siempre tengan datos válidos. Las validaciones explícitas hacen el código más robusto y fácil de depurar.

Mejora #2

Ubicación: `InventarioService.java`

Descripción del cambio:

- Se agregó el método `existeProductoPorNombre()` para evitar duplicados.
- Se implementó `mostrarResumenInventario()` que muestra productos, stock y valor total.
- Se corrigió el bucle infinito en `buscarProductoPorId()`.

Justificación:

La prevención de duplicados mantiene la integridad de los datos. El resumen de inventario mejora la visibilidad del negocio. La corrección del bucle previene errores en tiempo de ejecución.

Mejora #5

Ubicación: `Producto.java`

Descripción del cambio:

- Validación explícita en `reducirStock()` que lanza excepciones si:
 - La cantidad es negativa
 - No hay suficiente stock

Justificación:

Previene operaciones inválidas que podrían causar stock negativo, un problema crítico en gestión de inventario. Las excepciones específicas facilitan el manejo de errores.

Mejora #6

Ubicación: `Producto.java`

Descripción del cambio:

- Validaciones en el constructor para:
 - ID positivo
 - Nombre no vacío
 - Precio positivo
 - Stock no negativo

Justificación:

Garantiza que los productos siempre se creen en un estado válido, cumpliendo con reglas de negocio básicas desde su creación.

Mejora #7

Ubicación: `InventarioService.java`

Descripción del cambio:

- Implementación de `mostrarResumenInventario()` que muestra:
 - Lista formateada de productos
 - Stock actual
 - Valor total por producto
 - Valor total del inventario

Justificación:

Proporciona una visión consolidada del inventario útil para toma de decisiones, mejorando la utilidad del sistema para gestores.

Mejora #8

Ubicación: `Main.java` (nuevo método `eliminarProducto()`)

Descripción del cambio:

- Flujo que solicita confirmación antes de eliminar:
 - Muestra ID a eliminar
 - Pide confirmación explícita (s/n)

Justificación:

Previene eliminaciones accidentales, una protección importante en operaciones destructivas. Sigue el principio de confirmación explícita.