



UNIVERSIDAD DE LAS AMÉRICAS

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

**"Informe de Formulación y Solución del Problema de Gestión
de Inventarios – Programación II"**

Programación II

Elías Armas

Isaac Macías

Víctor Gómez

20/11/2025

ÍNDICE

1. Introducción
2. Informe del Problema (Formulación del Problema)
 - 2.1. Descripción del Problema
 - 2.2. Identificación de Variables
 - 2.3. Restricciones del Sistema
3. Planteamiento de Alternativas de Solución
 - 3.1. Alternativa 1: Gestión por Caducidad (Perecederos vs No Perecederos)
 - 3.2. Alternativa 2: Gestión por Origen del Producto (Nacional e Importado)
4. Selección y Fundamentación de la Solución
 - 4.1. Solución Seleccionada
 - 4.2. Fundamentos Técnicos de la Elección
5. Diagrama de Clases UML
 - 5.1. Descripción de las Clases
 - 5.2. Relaciones entre Clases
6. Conclusiones
7. Referencias

1. INTRODUCCION

La gestión eficiente del inventario es un elemento fundamental dentro de cualquier pequeña empresa, ya que influye directamente en los costos, la disponibilidad de productos y la toma de decisiones operativas. Frente a esta necesidad, el presente trabajo analiza un sistema de inventario diseñado bajo los principios de la Programación Orientada a Objetos (POO), utilizando herencia, polimorfismo y encapsulamiento para representar y administrar distintos tipos de productos. A partir de un mismo modelo base, se plantean alternativas que permiten controlar variaciones importantes como el origen del producto, su precio de venta y las restricciones propias del negocio. El objetivo es formular el problema, proponer soluciones viables y seleccionar la más adecuada, fundamentando su implementación mediante un análisis técnico y el diseño UML correspondiente. Este informe explica de manera clara el funcionamiento de cada parte del sistema y su contribución a una gestión de inventario más organizada, flexible y escalable.

2. INFORME DEL PROBLEMA (FORMULACION DEL PROBLEMA)

Descripción del Problema

Una empresa dedicada a la venta de productos físicos presenta dificultades con la gestión de su inventario. Actualmente no existe un sistema de inventario que permita registrar, actualizar o controlar correctamente las cantidades disponibles, los costos, las fechas de reabastecimiento ni los tipos de productos. Debido a esta falta de control, la empresa enfrenta problemas como desabastecimiento, perdidas económicas por compras innecesarias, retrasos en pedidos y un uso inadecuado del presupuesto asignado.

Identificación de Variables

- ID del producto
- Nombre del producto

- Stock disponible
- Costo de fabricación
- Precio final
- Tipo de producto (Nacional o Importado)
- IVA aplicado
- Arancel de importación
- Presupuesto máximo
- Capacidad máxima del inventario

Restricciones del Sistema

- El presupuesto no puede ser superado
- No se permite superar el stock máximo del inventario
- Los precios deben calcularse correctamente aplicando costos, IVA y aranceles
- El sistema debe manejar productos con comportamientos diferentes según origen

3. PLANTEAMIENTO DE ALTERNATIVAS DE SOLUCIÓN

A partir del análisis, se propusieron dos posibles soluciones:

ALTERNATIVA 1: Gestión por Caducidad (Perecederos vs No perecederos)

Esta alternativa se centra en la vida útil de los productos. El inventario se organiza considerando si los artículos tienen una fecha de caducidad que afecte su valor y su permanencia dentro del inventario. Bajo esta alternativa, se diferencia entre productos Perecederos y NoPerecederos.

Esta propuesta responde al factor tiempo que es importante en la gestión de inventarios para empresas que manejan bienes que se deterioran con el tiempo. Los productos perecibles requieren de estrategias como descuentos antes de la fecha de vencimiento o control mas estricto de abastecimiento. Por otro lado, los productos no perecederos pueden mantenerse mas tiempo almacenado, incluso aumentando su valor gracias a garantías o calidad duradera.

Sin embargo, esta alternativa no toma en cuenta factores financieros tales como impuestos, costos de importación o presupuesto disponible.

ALTERNATIVA 2: Gestión por Origen del Producto (Nacional e Importado)

En esta segunda alternativa propuesta, el inventario se gestiona considerando el país de origen. Se divide en dos grandes grupos: los productos nacionales, fabricados o adquiridos dentro del país, y los productos Importados, que requieren procesos adicionales como aduana, aranceles y transporte internacional.

El comportamiento del inventario en esta propuesta este guiado por la comprensión de que los productos importados suelen implicar costos adicionales, tales como los impuestos o las tarifas, que llegan a afectar directamente el precio de venta del producto. En contraparte los productos nacionales tienden a tener costos mas predecibles y estables, lo que facilita tanto su adquisición como su reposición.

Esta alternativa es especialmente útil para negocios donde la diferencia entre productos nacionales e importados influye en el precio, el presupuesto y la estrategia de ventas. Permite controlar mejor los recursos económicos, planificar compras y establecer precios más coherentes con la estructura de costos.

4. SELECCIÓN Y FUNDAMENTACIÓN DE LA SOLUCIÓN

La alternativa que solucionamos fue la segunda alternativa “*Gestión por Origen del Producto (Nacional e Importado)*”, debido a que se ajusta de mejor manera al problema planteado principalmente: gestionar el inventario considerando los costos, impuestos y presupuesto disponible.

Fundamentos técnicos de la elección

Cada tipo de producto implementa su propia formula de precio final, sin la necesidad de condiciones, lo que resulta en un código mas limpio, mantenable y escalable.

Encapsulamiento adecuado

Los atributos sensibles que son el costo, stock y precio final están protegidos, garantizando de esta manera la integridad de datos.

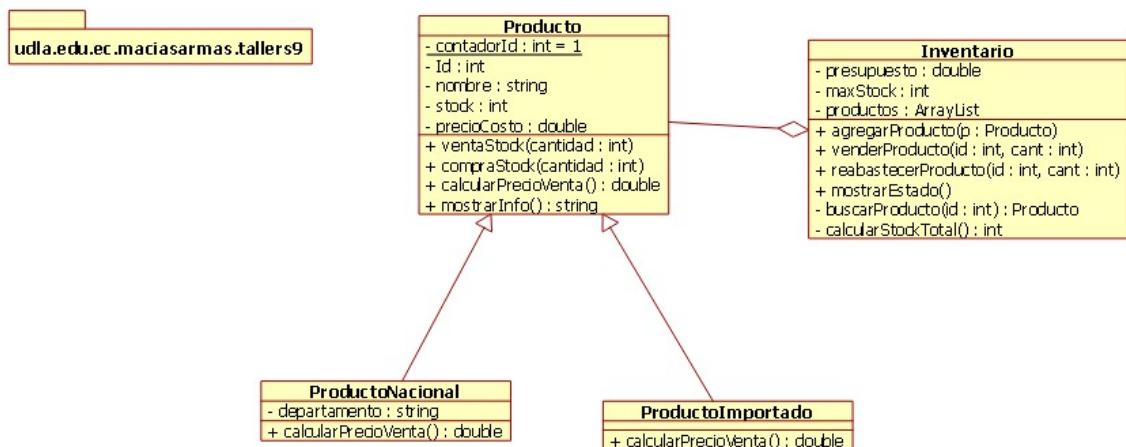
Herencia bien aplicada

ProductoNacional y ProductolImportado reutilizan todo lo común y solo definen la lógica que los diferencia.

Cumplimiento del presupuesto

El inventario puede validar si hay fondos suficientes antes de agregar nuevos productos en el almacenamiento.

5. DIAGRAMA DE CLASES UML



El siguiente diagrama UML realizado en Umbrello representa solución orientada a objetos para gestionar un inventario compuesto por productos nacionales e importados. El diseño parte a partir de una clase base llamada **Producto**, de la cual derivan dos tipos especializados: **Productonacional** y **ProductolImportado**. Además, existe una clase **Inventario**, responsable de administrar todos los productos registrados.

CLASE PRODUCTO (CLASE PADRE/SUPERCLASE)

Esta es la clase principal del sistema, ya que define las características comunes que poseerán todos los productos de inventario.

Atributos:

contadorId: int = 1

Atributo estático usado para generar IDs automáticos y únicos.

id: int

Identificador del producto.

nombre: string

Nombre del producto.

stock: int

Cantidad disponible en el inventario.

precioCosto: double

Costo base del producto (antes de calcular el precio de venta).

Métodos

ventaStock (cantidad: int)

Disminuye el stock al vender un número específico de unidades.

compraStock (cantidad: int)

Aumenta el stock del producto.

calcularPrecioVenta (): double

Método que calcula el precio de venta.

En las subclases, este método se sobrescribe aplicando reglas diferentes según el tipo de producto.

mostrarInfo (): string

Devuelve una descripción completa del producto, útil para imprimir información en pantalla.

*****CLASE PRODUCTONACIONAL (SUBCLASE)*****

Hereda todos los atributos y métodos de la superclase Producto y agrega su propia lógica.

Atributo adicional

departamento: string

Indica el departamento o región de origen del producto nacional.

Método sobrescrito

calcularPrecioVenta ()

Aplica una fórmula especial de precio para productos fabricados dentro del país (incluye IVA, pero no aranceles).

CLASE PRODUCTOIMPORTADO (SUBCLASE)

También hereda los atributos de la superclase Producto, pero se diferencia por el cálculo de precio.

Método sobrescrito

calcularPrecioVenta ()

Calcula el precio sumando impuestos como IVA y, adicionalmente, costos de importación o aranceles.

CLASE INVENTARIO

Es la clase encargada de administrar el conjunto de productos existentes.

Atributos

presupuesto: double

Dinero disponible para comprar productos.

maxStock: int

Capacidad máxima del inventario (número total de unidades).

productos: ArrayList

Colección donde se almacenan todos los productos (nacionales o importados).

Métodos

agregarProducto (p: Producto)

Añade un producto a la lista (si hay espacio y presupuesto).

venderProducto (id: int, cant: int)

Busca un producto por ID y descuenta unidades del stock.

reabastecerProducto (id: int, cant: int)

Aumenta el stock del producto y descuenta del presupuesto el costo de reposición.

mostrarEstado ()

Muestra todos los productos registrados y el estado general del inventario.

buscarProducto (id: int): Producto

Devuelve el producto cuyo ID coincide con el solicitado.

calcularStockTotal ()

Retorna la sumatoria de todas las unidades disponibles en inventario.

RELACION ENTRE CLASES

Herencia (Se representan con flechas con punta vacía)

ProductoNacional → Producto

Productolimportado → Producto

Esto representa una jerarquía, las subclases heredan atributos y métodos de la superclase.

Composición (rombo negro entre Inventario y Producto)

Inventario <- Producto

El inventario contiene productos. Si el inventario deja de existir, la lista de productos también desaparece. Es una relación fuerte (tiene una "parte de" real).

6. CONCLUSIONES

- La estructura del sistema demuestra que la POO permite representar productos y operaciones de inventario de manera clara y organizada. La herencia facilita la creación de tipos especializados como productos nacionales e importados. Esto mejora la flexibilidad y coherencia del diseño.
- Las alternativas analizadas evidencian que distintos enfoques pueden resolver el problema según la necesidad del negocio. Mientras una solución prioriza la caducidad, la otra se enfoca en costos e impuestos. Esto resalta la importancia de formular adecuadamente el problema para seleccionar la opción correcta.
- El UML presentado confirma la viabilidad del sistema al mostrar relaciones coherentes entre clases y responsabilidades bien definidas. El inventario actúa como administrador central, garantizando control y organización. Esto permite un sistema escalable, comprensible y aplicable en escenarios reales de gestión.

7. REFERENCIAS

GitHub. (2023). GitHub Copilot: Your AI pair programmer. GitHub.

<https://github.com/features/copilot>