# Análisis del Programa de Inventario en Java

## Yoiner Alejandro Lazaro Tellez-192424

## John Stiven Santiago Rendon-19245

**Joseph Santiago Vila Alvarez-192454**

**Docente: Jesus Eduardo Guerrero Rodriguez**

**Universidad Francisco de Paula Santander Ocaña**

**2025**

## 1. Introducción

El presente informe describe el funcionamiento y estructura de un programa desarrollado en Java, cuyo propósito es administrar un inventario básico de productos en una tienda. El sistema permite registrar productos, visualizar un reporte detallado del inventario y controlar el flujo de datos mediante un menú interactivo en consola.

## 2. Objetivo del Programa

El objetivo principal del programa es ofrecer una herramienta sencilla que permita:

- Registrar hasta cinco productos, ingresando su nombre, cantidad y precio por unidad.  
- Visualizar un reporte que detalle la cantidad, precio y valor total de cada producto.  
- Calcular el valor total de todo el inventario.  
- Brindar una navegación amigable mediante un menú de opciones.

## 3. Metodología

Para el desarrollo del programa se siguieron los siguientes pasos:  
**1. Planificación:** Se definieron los datos necesarios (nombre del producto, cantidad y precio).  
**2. Diseño:** Se pensó en cómo el usuario iba a interactuar con el sistema (menú con opciones).  
**3. Codificación:** Se programó usando el lenguaje Java, enfocándonos en mantenerlo simple y entendible.  
**4. Pruebas:** Se probó el programa con diferentes entradas para verificar que los cálculos fueron correctos.  
**5. Revisión:** Se mejoró el código con comentarios y estructura clara.

## 4. Estructura del Programa

**4.1. Importación de librerías**  
Se importa la clase Scanner desde el paquete java.util, la cual se utiliza para capturar la entrada de datos proporcionada por el usuario.

**import java.util.Scanner;**

**4.2. Declaración de variables**  
Se crean tres arreglos de tamaño cinco para almacenar los datos de los productos:  
**- String[] nombres:** almacena los nombres de los productos.  
**- int[] cantidades:** guarda las cantidades disponibles de cada producto.  
**- double[] precios:** registra el precio unitario de cada producto.  
Adicionalmente, se define:  
**- int totalProductos:** contador que lleva el número de productos agregados.  
**- int opcion:** controla la selección de opciones del menú principal.

**4.3. Bucle principal (do-while)**  
El programa utiliza un bucle do-while que:  
- Muestra un menú con tres opciones: agregar productos, ver reporte de inventario o salir.  
- Lee la opción seleccionada por el usuario.  
- Ejecuta el bloque de código correspondiente según la elección.  
- El ciclo se repite hasta que el usuario selecciona la opción de salir (opción 3).

## 5. Funcionamiento de las Opciones del Menú

**5.1 Opción 1: Agregar productos**- El programa pregunta cuántos productos desea agregar.  
- Por cada producto, solicita:  
 - El nombre del producto.  
 - La cantidad disponible.  
 - El precio por unidad.  
- Cada dato se almacena en la posición correspondiente de los arreglos.  
- Se asegura que no se exceda el límite de cinco productos, mostrando un mensaje de advertencia en caso de alcanzarlo.

**5.2. Opción 2: Ver reporte de inventario**  
- Se recorren los arreglos hasta el número de productos registrados.  
- Para cada producto, se muestra:  
 - Su nombre.  
 - La cantidad disponible.  
 - El precio por unidad.  
 - El total por producto (cantidad × precio).  
- Además, se calcula y muestra el valor total del inventario sumando los totales de todos los productos.

**5.3. Opción 3: Salir**  
- El programa muestra un mensaje indicando que está saliendo.  
- El bucle do-while termina, finalizando la ejecución del programa.

**5.4. Validación de entradas**  
- Si el usuario ingresa una opción no válida (distinta de 1, 2 o 3), el programa muestra un mensaje de error y vuelve a mostrar el menú.

## 6. Consideraciones adicionales

- Se utiliza **scanner.nextLine()** después de cada **scanner.nextInt()** o **scanner.nextDouble()** para limpiar el salto de línea pendiente, evitando errores en la lectura de entradas de tipo String.  
- El sistema es estático: los datos se almacenan en memoria durante la ejecución, pero no se guardan permanentemente una vez que se cierra el programa.

## 7. Conclusión

Este proyecto fue muy útil para afianzar nuestros conocimientos sobre programación en Java. Aprendimos a trabajar con arreglos, a organizar mejor nuestro código, y a entender cómo interactuar con el usuario por consola. También mejoramos nuestra lógica de programación y el trabajo en equipo.

**Recomendaciones:**  
- Comentar siempre el código para que sea fácil de entender.  
- Usar nombres claros en las variables.  
- Probar el programa con distintos datos para asegurar que funcione correctamente.