

Informe sobre el trabajo lógico detrás del código del inventario de productos

Asignatura: Programación I

Profesor: Jesús Guerrero (alias “el Yisus”)

Estudiantes:

Jhoan Molina Gómez // 192490

Katherine Cicero // 192460

Miembro honorario: Jesús David Pérez

(lo incluyo porque al inicio nos ayudó, aunque luego se cambio de grupo)

Universidad Francisco de Paula Santander seccional Ocaña

Fecha de entrega: 02/05/2025

Ingeniería de sistemas

Tabla de contenidos

1) Introducción	Pag 2
a. Objetivos	Pag 2
2) Planteamiento del problema	Pag 3
a. Contexto del problema	Pag 3
3) Análisis del problema	Pag 4
4) Diseño de la solución	Pag 4
5) Desarrollo del problema	Pag 5
6) Implementación	Pag 5
7) Corrección de errores	Pag 5
8) Resultados obtenidos	Pag 6
9) Dificultades encontradas	Pag 6
10) Reflexión	Pag 7
11) Conclusiones	Pag 7

Introducción

Por medio de este trabajo esperamos, poder avanzar un poco mas nuestros conocimientos y estrategias para diseñar soluciones lógicas y concretas a los problemas dados cotidianamente, o en el ámbito laboral. El trabajo en si no es tan complicado, pero eso no quiere decir que sea sencillo, simplemente debería llevarnos algo mas de tiempo entre el análisis y el desarrollo del trabajo en cuestión.

Objetivos

Generales

- 1) Proponer soluciones lógicas y sencillas a los problemas otorgados.
- 2) Analizar problemas y diseñar soluciones optimas para los problemas planteados.
- 3) Evaluar nuestros conocimientos en los procesos lógicos.

Específicos

- 1) Formular un pseudocódigo y desarrollar un código eficaz para el control y registro del sistema del supermercado.
- 2) Optimizar el código y los errores que pueden ir apareciendo en este mismo proceso.
- 3) Comprobar que el código sea optimo y dar al usuario final el producto en buena calidad.

Planteamiento del problema

Desarrollar una aplicación en Java que permita gestionar el inventario de productos de una tienda, utilizando variables, arrays, estructuras de control y matrices para almacenar los datos de los productos, calcular su precio total y generar un reporte del inventario.

Contexto del Problema:

Se les ha pedido desarrollar una aplicación para llevar el inventario de productos de una pequeña tienda. La tienda vende diferentes tipos de productos y necesita que el sistema registre el nombre de los productos, la cantidad en inventario y el precio por unidad. Además, el sistema debe calcular el valor total del inventario y mostrar un reporte final.

Requisitos Funcionales:

- El sistema debe registrar el nombre, cantidad y precio de al menos 5 productos.
- Para cada producto, se debe calcular el valor total (cantidad * precio).
- Al final, se debe generar un reporte que muestre el valor total del inventario (sumando los valores de todos los productos).
- El sistema debe permitir agregar más productos si es necesario.
- El sistema debe incluir una opción para actualizar la cantidad de un producto si llega más inventario.

Análisis del problema

Se nos solicita crear un sistema tipo de inventario de productos para una tienda local donde se puedan integrar mínimo 5 productos con su precio, cantidad disponible y total en dinero del producto almacenado, también nos piden que estos valores se muestren en pantalla y puedan ser editables.

Al final de todo generar un reporte con todo el costo del almacenaje de todos los productos disponibles en el sistema.

Diseño de la solución y desarrollo del problema

Lo primero y más lógico sería crear un sistema de entrada del usuario con los datos solicitados, introducirlos todos en un conjunto y hacer los procesos desde ahí mismo, en este caso podríamos usar una matriz o un objeto si la matriz no nos permite introducir el nombre a los productos que queremos meter.

Después de la entrada viene la parte lógica de cálculo donde deberíamos sumar la cantidad de productos multiplicarla por el precio y mostrarla en pantalla, con todos esos resultados sumados tendríamos el valor total del almacén en cuestión.

Por último, se haría la edición de cantidad de producto, en el caso tal de que llegase mas inventario, esto se haría con una edición al primer paso, permitiéndonos cambiar la cantidad del producto o añadir un nuevo producto en tal caso que se quiera vender un producto diferente.

Sabiendo todo esto gracias a las instrucciones dadas en el planteamiento, lo mejor sería implementarlo con un tipo objeto para que la matriz permita nombre, también tendríamos que añadir un bucle para la edición de producto y la adición de nuevos productos. El código quedara algo largo, ya que para implementar el ultimo bucle se debe hacer un menú interactivo, para permitir modificaciones al inventario.

Implementación

La manera en cómo se implementó la solución fue tal y como se dio en el desarrollo del problema, primero que todo se pidieron los datos al usuario, se almacenan en una matriz y se muestran en pantalla después de hacer los cálculos. Luego, se le da la opción de poder editar la cantidad o de poder añadir nuevos artículos al sistema.

En caso de que no se quiera hacer ninguna de las dos, el inventario se da por finalizado y se muestran los resultados finales del inventario.

Hubo varias ideas para implementar la solución, entre switch, bucles For, etc.

Al final decidimos algo más simple, solo con For y While para estructurar la solución.

Corrección de errores

En este punto tuvimos demasiados problemas, porque primero no sabíamos cómo implementar bien la matriz con el objeto.

Luego, vinieron los problemas en el momento de tener que hacer el menú interactivo, puesto que, los bucles solían quedarse de forma infinita o solía romperse a la mitad del proceso.

También, existieron diferentes errores al momento de imprimir los resultados en pantalla, puesto que los datos quedaban te hayan equivocados o estaban trucados con los anteriores.

Todo esto se fue corrigiendo, de manera ordenada, hasta que el proceso estuvo completamente limpio y funcional. Fue revisado un total de 19 veces mientras se depuraba y 8 veces después de terminado.

Resultados obtenidos

Como resultado de este trabajo, obtuvimos un sistema funcional con un inventario dinámico que permite agregar y modificar productos. El programa muestra la información actualizada de cada producto, incluyendo su nombre, cantidad, precio unitario y el valor total del inventario. Si bien el código cumple con su propósito, podría optimizarse, ya que fue necesario repetir algunas líneas para lograr que los resultados se mostraran correctamente en pantalla. Esto se debió a las limitaciones que encontré al intentar mantener la funcionalidad del programa sin afectar la lógica de los bucles ni la interacción del usuario.

Dificultades encontradas

Una de las principales dificultades encontradas fue la organización lógica del código, ya que muchas variables generaban conflictos entre sí al reutilizar nombres o posiciones. También se presentaron múltiples errores menores que obligaron a modificar el código en varias ocasiones hasta encontrar una solución funcional.

Otra dificultad importante fue comprender el funcionamiento del tipo de dato Object, ya que no lo habíamos utilizado antes y fue necesario investigarlo para entender su uso. No pudimos utilizar una matriz convencional de tipo específico (como int o String), ya que necesitábamos almacenar tanto datos numéricos como nombres de productos. Esto nos llevó a buscar una alternativa que nos permitiera almacenar distintos tipos de datos en la misma estructura, motivo por el cual optamos por usar una matriz de tipo Object.

Reflexiones

Una de las principales reflexiones que nos deja este trabajo es que aún necesitamos adquirir mayor conocimiento y práctica para lograr un código más limpio y optimizado. Aunque fuimos capaces de desarrollar un programa funcional, reconocemos que, por nuestra falta de dominio total del lenguaje, algunas soluciones pueden no ser las más eficientes.

Asimismo, destacamos como aprendizaje clave el uso del tipo de dato Object para manejar distintos tipos de datos en una misma estructura, lo cual es un paso importante hacia el manejo de bases de datos desde Java. Este conocimiento es valioso porque puede aplicarse posteriormente en el desarrollo de páginas web o sistemas más complejos que requieren una lógica sólida para el manejo de información.

Conclusión

Gracias a este trabajo, comprendimos que Java es un lenguaje muy versátil, capaz de interactuar con distintos tipos de datos, tanto en estructuras estáticas como en simulaciones de manejo de bases de datos. Aunque no sea el lenguaje más especializado para este último propósito, su uso amplio y estandarizado en el desarrollo de aplicaciones web lo convierte en una herramienta poderosa.

A pesar de no dominar completamente el lenguaje, logramos construir soluciones funcionales, lo cual demuestra su flexibilidad. Esta experiencia nos deja con una motivación clara: si con conocimientos básicos ya es posible crear programas útiles, las posibilidades que se abrirán al dominar Java por completo son verdaderamente emocionantes.