

**Informe de Aplicación de Inventario de Productos**

**INTEGRANTES**

**Danna Sofia Velandia Arciniegas 192478**

**Juan José Quintero Sánchez 192456**

**Yina Sofia Bohórquez 192438**

**Grupo C**

**UNIVERSIDAD FRANCISCO DE PAULA SANTANDER**

**OCAÑA, NORTE DE SANTANDER**

**2025**

**INTRODUCCIÓN**

El presente documento detalla el desarrollo de un sistema de gestión de inventario en Java, diseñado para administrar productos, categorías y secciones en un almacén. El sistema permite agregar productos, actualizar cantidades y generar reportes organizados por categorías y secciones, facilitando el control y análisis del inventario.

**OBJETIVOS**

1. Implementar un sistema de inventario con funcionalidades básicas (agregar, actualizar y reportar productos).

2. Organizar los productos por categorías y secciones para un manejo estructurado.

3. Aplicar conceptos de programación en Java, como arreglos, matrices, validación de entrada y modularización.

**DESARROLLO DEL SISTEMA**

**1.Estructura del programa**

El sistema se desarrolló utilizando los siguientes componentes:

* Arreglos unidimensionales: Para almacenar nombres, cantidades, precios y valores totales de los productos.
* Matrices tridimensionales: Para organizar los productos por categoría y sección (productosPorCategoriaSeccion).
* Constantes: Definidas para límites de productos, categorías y secciones.
* Métodos modularizados: Cada funcionalidad (agregar producto, actualizar cantidad, generar reportes) se implementó en métodos separados para mantener el código limpio y reusable.

**2. Funcionalidades Clave**

1. Agregar Producto:

* Entrada numérica correcta (cantidad y precio no negativos).
* Límite de productos en el inventario (MAX\_PRODUCTOS).
* Límite de productos por categoría y sección (MAX\_PRODUCTOS\_POR\_SECCION).

1. Asignación a categorías y secciones:

* El usuario selecciona de una lista predefinida.

1. Actualizar Cantidad:

* Muestra una lista de productos existentes.
* Actualiza la cantidad y recalcula el valor total automáticamente.

1. Generación de Reportes:

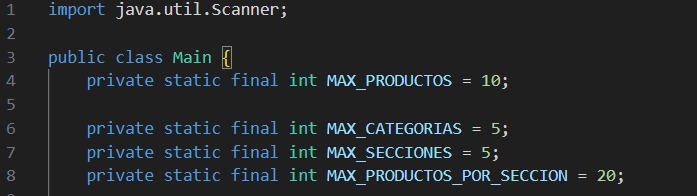
* Reporte general: Muestra todos los productos con sus detalles.
* Reporte por categoría y sección: Organiza la información jerárquicamente para facilitar el análisis.

**3. Manejo de Errores**

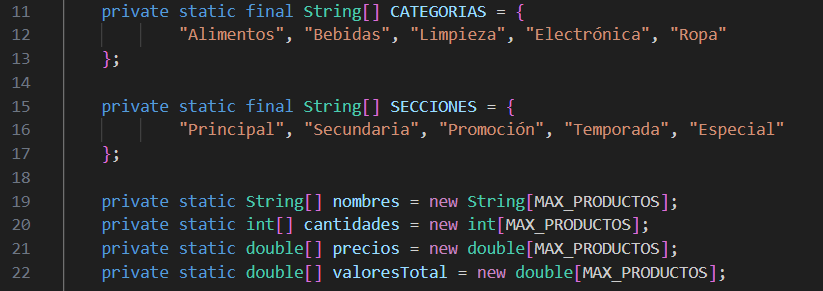
* Se implementaron bloques try-catch para evitar fallos por entradas incorrectas (ej.: texto en lugar de números).
* Mensajes claros para guiar al usuario en caso de errores.

**4.Codigo Implementado**

* **Lo primero que hacemos es importar el scanner y utilizamos constantes para las categorías y secciones.**



* **Nombres de categorías y secciones.**



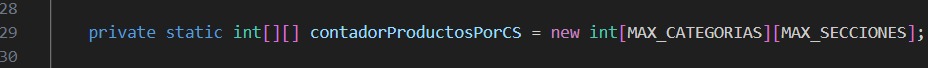
* **Matrices para almacenar la información por categoría y sección.**



* **Matriz para almacenar índices de productos por categoría y sección.**



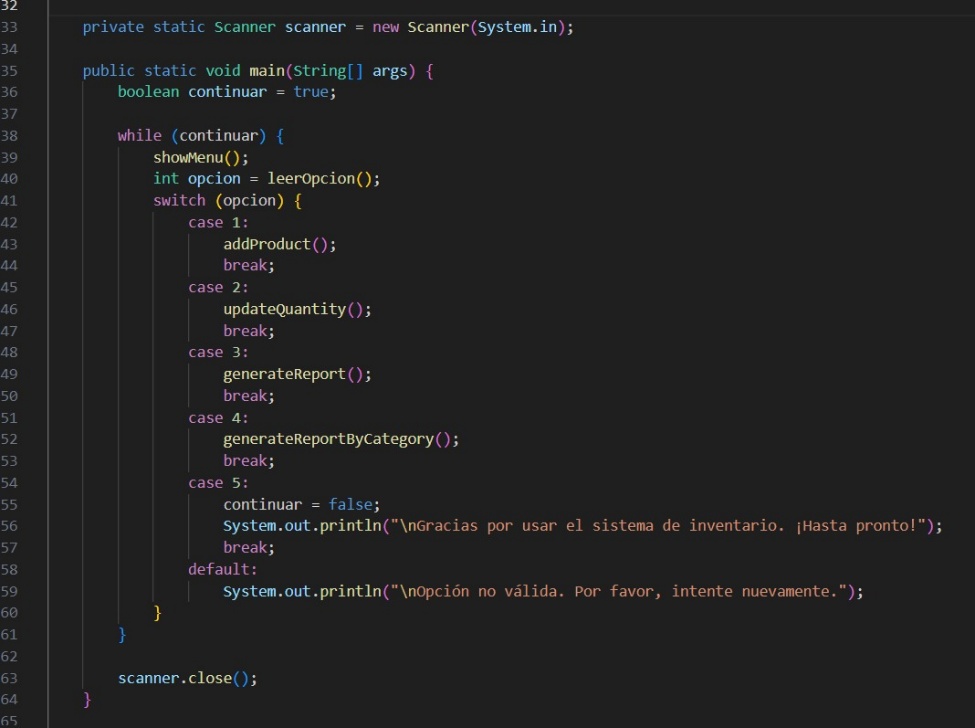
* **Contador de productos por categoría y sección.**



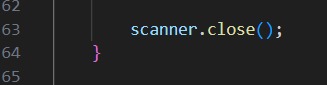
* **Variable para llevar cuenta de cuántos productos hay en el sistema.**



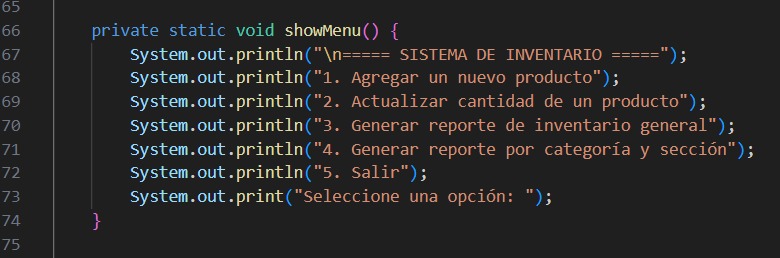
* **Scanner para entrada de datos**



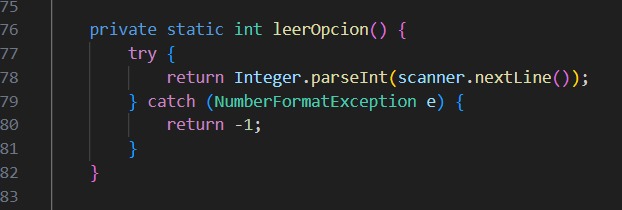
* **Cerrar el scanner al finalizar.**



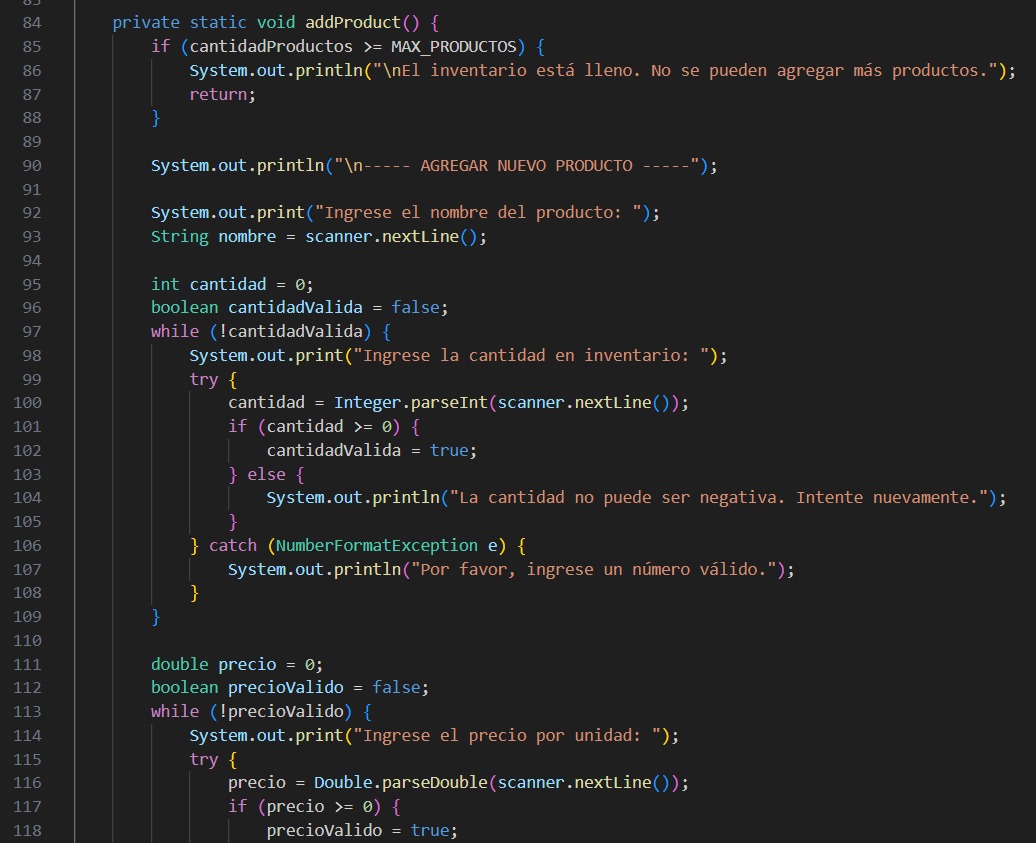
* **Muestra el menú principal de la aplicación.**

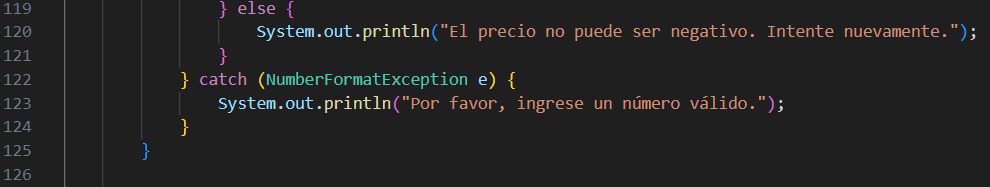


* **Lee la opción seleccionada por el usuario.**



* **Agrega un nuevo producto al inventario**

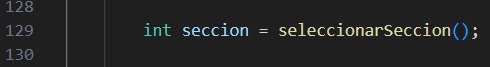




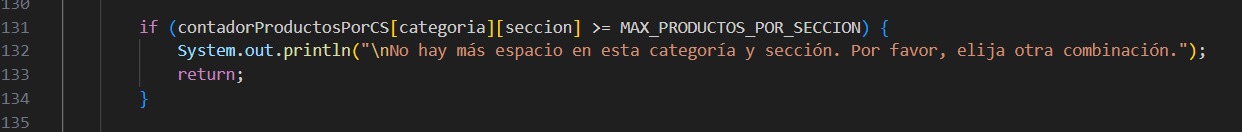
* **Seleccionar categoría para el producto**



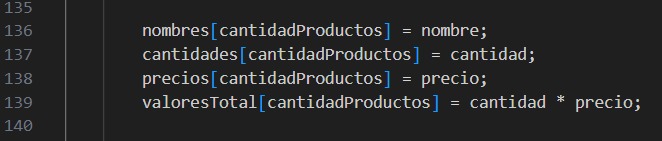
* **Seleccionar sección para el producto**



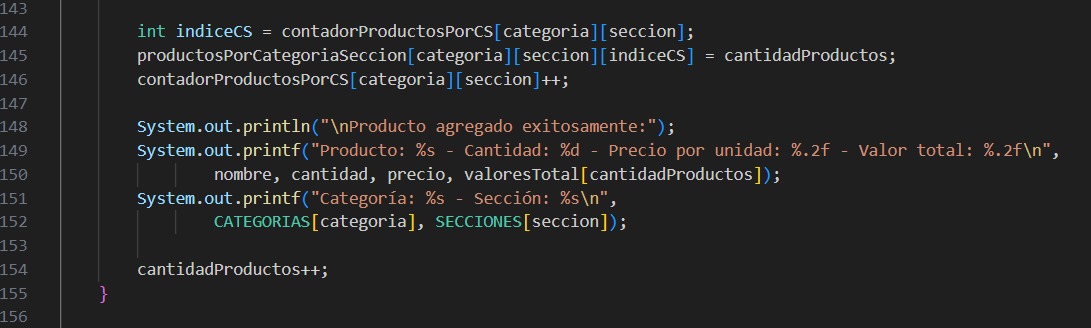
* **Verificar si hay espacio en la matriz de la categoría y sección seleccionada**



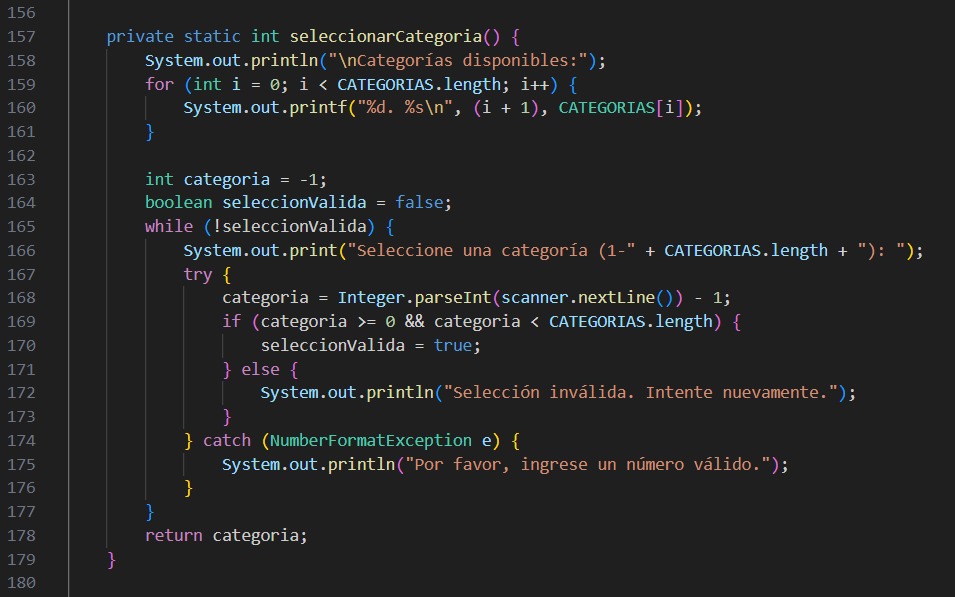
* **Guardar el nuevo producto**



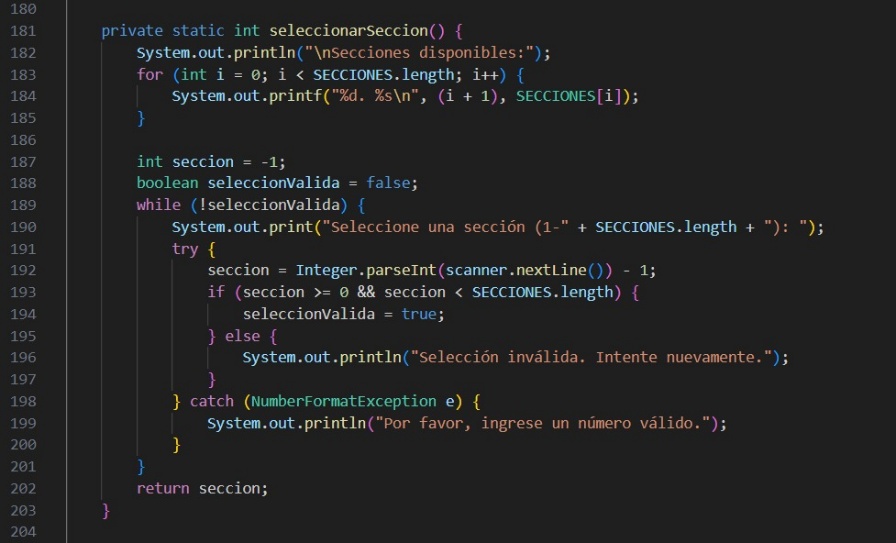
* **Agregar el índice del producto a la matriz por categoría y sección**



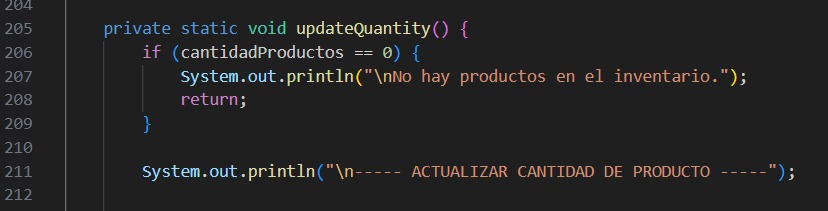
* **Muestra las categorías disponibles y solicita al usuario seleccionar una.**



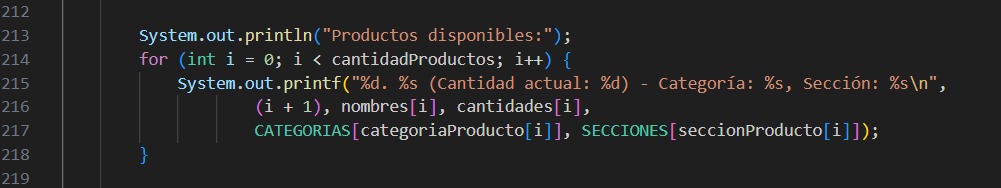
* **Muestra las secciones disponibles y solicita al usuario seleccionar una.**



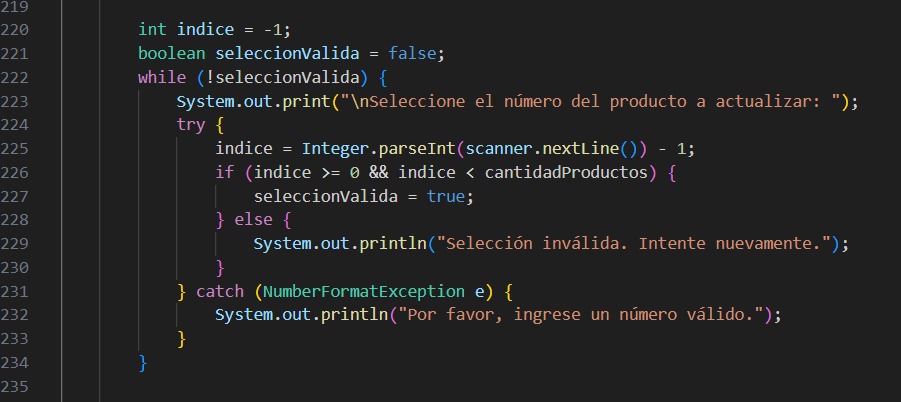
* **Actualiza la cantidad de un producto existente.**



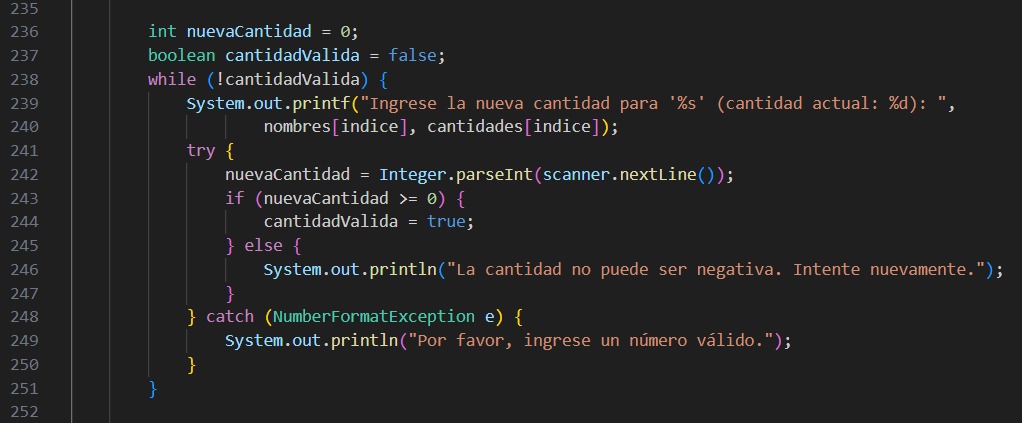
* **Mostrar productos disponibles.**



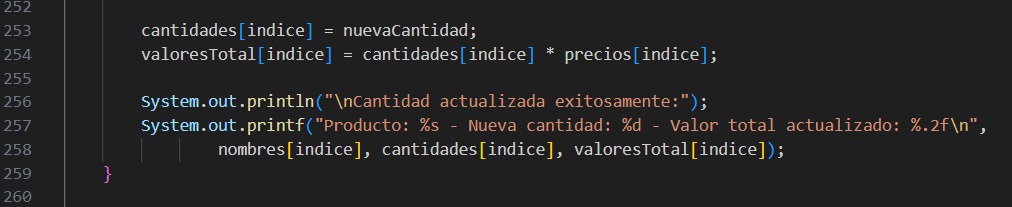
* **Seleccionar producto a actualizar**



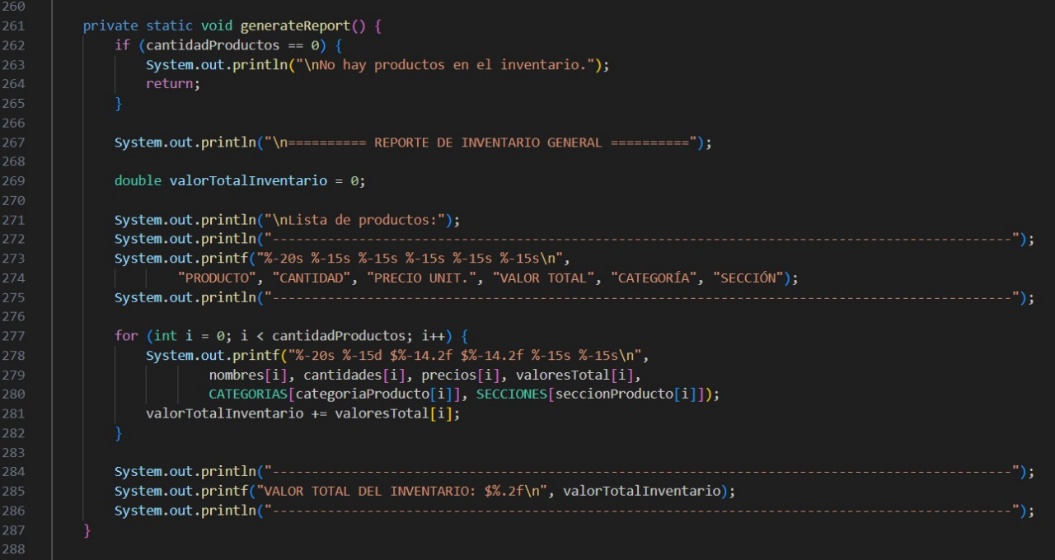
* **Ingresar nueva cantidad.**



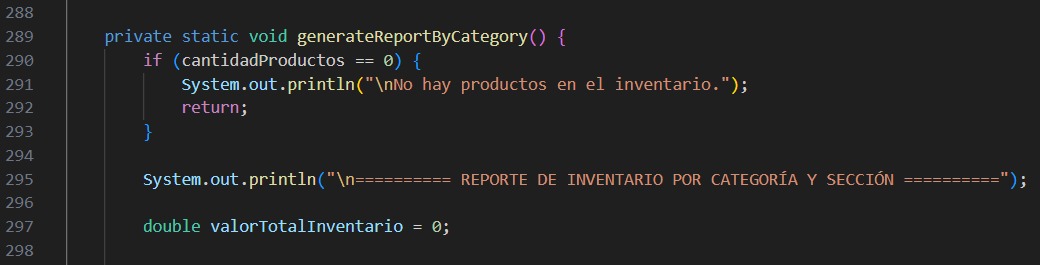
* **Actualizar cantidad y valor total.**



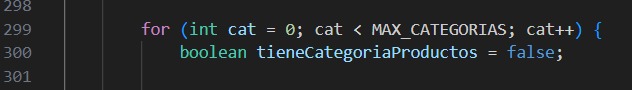
* **Generar un reporte del inventario.**



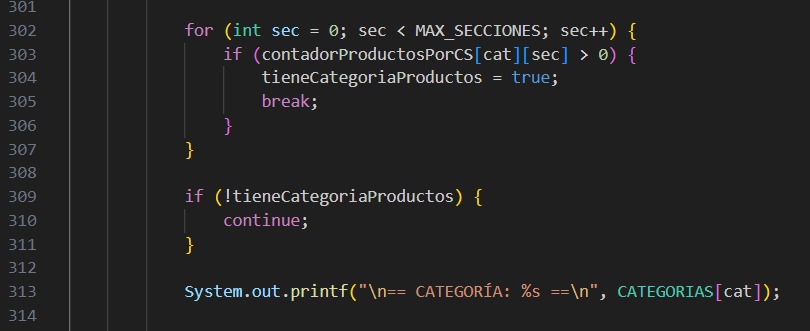
* **Generar un reporte del inventario organizado por categoría y sección.**



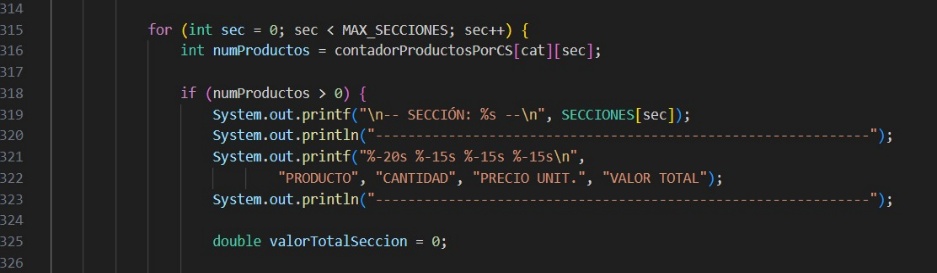
* **Recorrer cada categoría.**



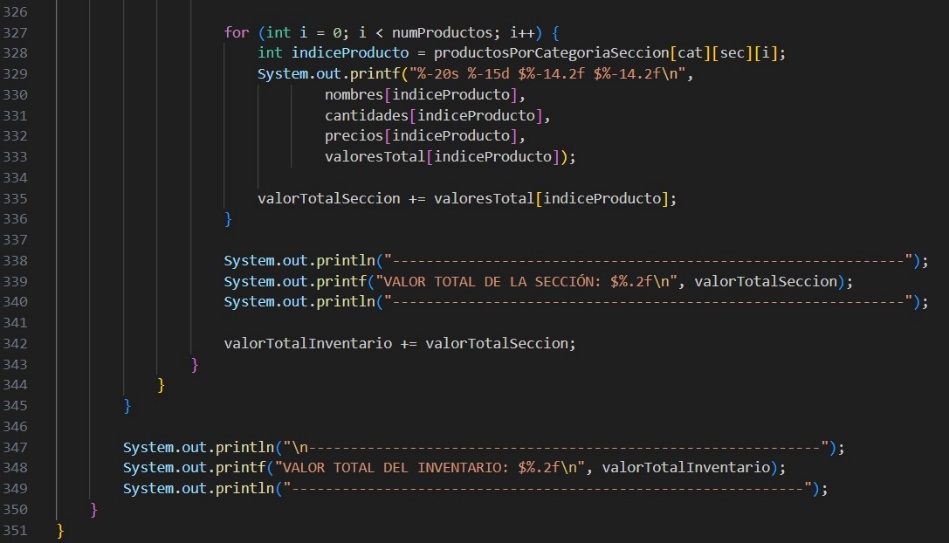
* **Verificar si hay productos en esta categoría.**



* **Recorrer cada sección de esta categoría.**



* **Recorrer productos de esta sección.**



**CONCLUSIONES Y REFLEXIÓN**

* Lo que aprendimos fue:

1. **Organización de datos**: El uso de matrices multidimensionales (productosPorCategoriaSeccion) permitió manejar relaciones complejas (categoría → sección → productos) de manera eficiente.
2. **Validación de entradas:** Asegurar que el programa no falle con datos incorrectos mejora la experiencia del usuario.
3. **Modularización:** Dividir el código en métodos específicos facilita el mantenimiento y la escalabilidad.

* ¿Como este proyecto nos ayuda en nuestra lógica para el desarrollo de software?

Nos ayuda mucho ya que, gracias a este proyecto, nos ayuda a pensar paso a paso y estructurar bien las soluciones, usando estructuras como condicionales, bucles, arreglos y manejo de errores, lo que refuerza nuestra lógica y mejorando la comprensión del flujo de un programa.