UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

Tema:

Sistema de Gestión de Tienda de Ropa en Línea

Autores:

Anthony Aguilar

Ronny Casco

Alexandra Lalaleo

Ronny Lugmaña

Materia:

Programación Orientada a Objetos

NRC:

5401

TUTOR:

Ing. Luis Enrique Jaramillo Montaño

Sangolquí – Ecuador

Objetivo General

Desarrollar un sistema de software que permita la gestión de productos, inventarios y ventas en una tienda de ropa en línea, proporcionando una interfaz interactiva para el usuario.

Objetivos Específicos

- Diseñar e implementar una clase producto que maneje la información básica de los productos, como nombre, precio y stock.
- Crear una clase Venta para gestionar las transacciones de ventas, incluyendo el cálculo del total y la actualización del stock.
- Desarrollar una clase Tienda que administre los productos disponibles en la tienda.
- Crear un programa principal que integre estas clases y proporcione una interfaz interactiva para el usuario.}

Marco Teórico

La gestión de inventarios y ventas es una tarea fundamental en cualquier negocio de retail. La Programación Orientada a Objetos (POO) es un paradigma de programación que facilita esta tarea al permitir la creación de clases que modelan entidades del mundo real, como productos y ventas, y sus interacciones ya que una tienda virtual es un sitio web diseñado especialmente para vender productos u ofrecer servicios mediante el comercio electrónico.

Cada objeto es una instancia de una clase, que actúa como un molde que define sus propiedades y comportamientos. Las principales características de la POO son:

Encapsulación: Es el proceso de almacenar en una misma sección los elementos de una abstracción que constituyen su estructura y su comportamiento; sirve para separar el interfaz contractual de una abstracción y su implantación. Esto permite proteger los datos de accesos no autorizados y facilita la gestión del código.

Herencia: La herencia permite que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.

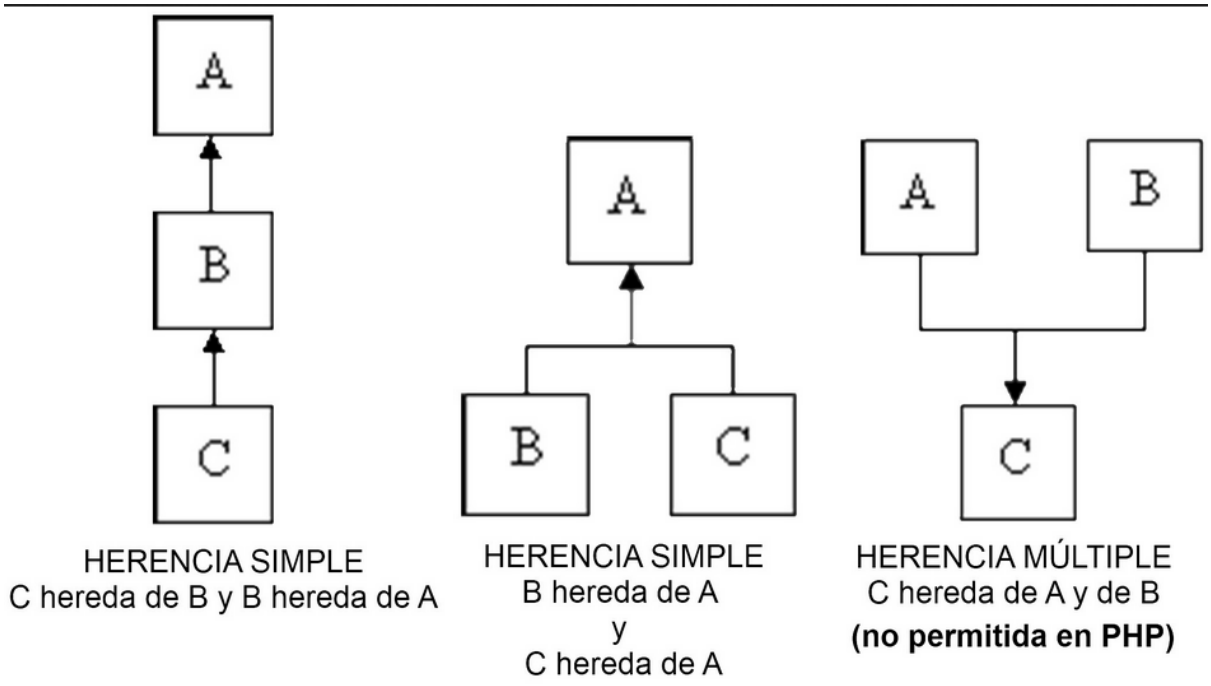

Figura 1: Diagrama de herencia. Obtenida de (https://keepcoding.io/blog/que-es-la-herencia-multiple-y-como-usarla/)

Polimorfismo: es la capacidad de un objeto para adquirir múltiples formas o comportamientos. En la programación, esto se traduce en la habilidad de una clase para implementar métodos con el mismo nombre, pero con comportamientos distintos, dependiendo de la clase específica con la que se esté interactuando. Esto añade flexibilidad y extensibilidad al código, permitiendo adaptarse a diferentes situaciones y requerimientos. Como ejemplo podemos usar una analogía LEGO, que es como tener una pieza que puede encajar en varios lugares o tener varios usos según el contexto.

Abstracción: Consiste en simplificar sistemas complejos mediante la creación de clases que representan conceptos generales. La abstracción permite centrarse en los aspectos esenciales del objeto sin preocuparse por los detalles específicos de implementación.

Gestión de inventarios

La gestión de inventarios es el seguimiento del inventario desde el momento de su fabricación hasta los almacenes, y desde estas instalaciones hasta el punto de venta. El inventario es el conjunto de artículos o materiales que un negocio tiene la intención de vender a los clientes con fines lucrativos. El objetivo de la gestión de inventarios es tener los productos correctos en el lugar adecuado y en el momento preciso. Esto requiere visibilidad de inventario: saber cuándo se debe hacer los pedidos, cuánto se debe pedir y dónde almacenar las existencias. Los pasos básicos de la gestión de inventarios incluyen:

- Compra de inventario
- Almacenamiento de inventario
- Beneficio del inventario

Gestión de Ventas en línea

Ya sabemos que CRM es la sigla utilizada para "Customer Relationship Management" (Gestión de Relación con los Clientes). Sin embargo, la definición completa de CRM va más allá: se trata de una gestión integrada de ventas, marketing, atención al cliente y todos los puntos de contacto. Es un software de gestión de clientes que recaba toda la información e interacciones con tus clientes para que hagas un seguimiento correcto de tu pipeline de ventas**.** Esa información permite que te enfoques en aquellos leads con mayores chances de convertirse en clientes. Además, un CRM acompaña a tus prospectos y clientes a lo largo del embudo de ventas facilitando:

- Atraer leads calificados;
- Convertirlos en clientes;
- Fidelizarlos.

Planteamiento del Problema

En el contexto de una tienda de ropa en línea, es necesario contar con una herramienta que permita registrar productos, gestionar inventarios y realizar ventas de manera eficiente. La falta de un sistema automatizado puede llevar a errores y pérdidas de información, pérdida económica y la dificultando de la gestión del negocio.

Diseño

El diseño del sistema se basa en la creación de clases que representan las tres entidades principales de la tienda. Cada clase tiene atributos y métodos que permiten gestionar la información y las operaciones relacionadas con esa entidad.

Diagrama UML

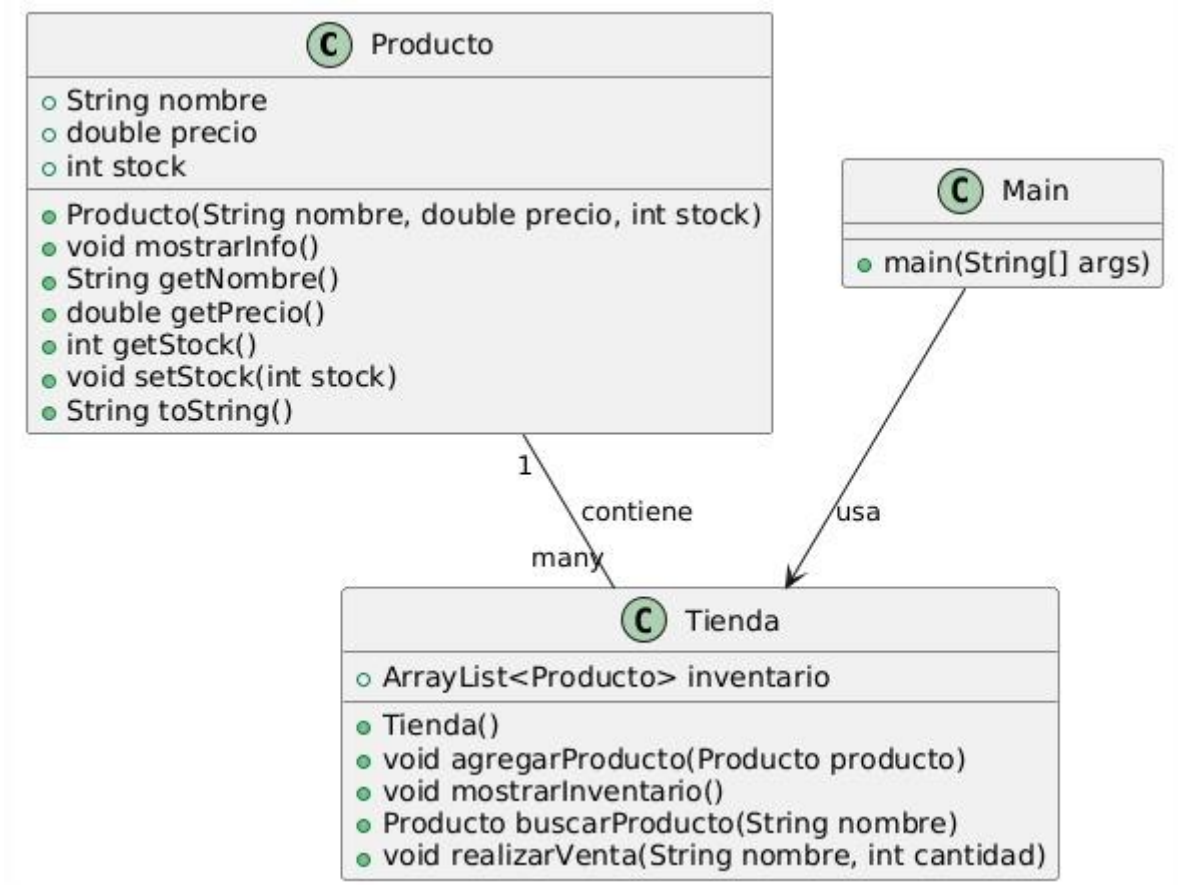

Figura 2: Diagrama UML.

Implementación del Programa

El programa principal proporciona una interfaz interactiva para el usuario mediante un menú de opciones que permite registrar productos, consultar inventarios y realizar ventas.

A continuación, presentamos el código utilizado en el programa.

clase Main

```
import java.util.Scanner;
public class Main {
   public static void main (String[] args) {
      Scanner scanner = new Scanner(System.in);
      Tienda tienda = new Tienda();
      while (true) {
         System.out.println("\nSeleccione una opción:");
         System.out.println("1. Registrar Producto");
         System.out.println("2. Consultar Inventario");
         System.out.println("3. Realizar Venta");
         System.out.println("4. Salir");
         int opcion = scanner.nextInt();
         scanner.nextLine(); // Consumir la línea nueva

         if (opcion == 1) {
            System.out.println("Ingrese el nombre del producto:");
            String nombre = scanner.nextLine();
            System.out.println("Ingrese el precio del producto:");
            double precio = scanner.nextDouble();
            System.out.println("Ingrese la cantidad en stock:");
            int stock = scanner.nextInt();
            scanner.nextLine(); // Consumir la línea nueva
            tienda.agregarProducto(new Producto(nombre, precio, stock));
         } else if (opcion == 2) {
            tienda.mostrarInventario();
         } else if (opcion == 3) {
            System.out.println("Ingrese el nombre del producto:");
            String nombre = scanner.nextLine();
            System.out.println("Ingrese la cantidad a vender:");
            int cantidad = scanner.nextInt();
            scanner.nextLine(); // Consumir la línea nueva

            tienda.realizarVenta(nombre, cantidad);
         } else if (opcion == 4) {
            System.out.println("Saliendo del sistema...");
            break;
         } else {
            System.out.println("Opción no válida, intente de nuevo.");
```

```
            }
        }
        scanner.close();
    }
}
```

clase Producto

```
class Producto {
    String nombre;
    double precio;
    int stock;

    public Producto(String nombre, double precio, int stock) {
        this.nombre = nombre;
        this.precio = precio;
        this.stock = stock;
    }

    public void mostrarInfo() {
        System.out.println("Nombre: " + nombre + ", Precio: " + precio + ", Stock: " + stock);
    }
}
```

clase Tienda

```
import java.util.ArrayList;

class Tienda {
    ArrayList<Producto> inventario = new ArrayList<>();

    public void agregarProducto(Producto producto) {
        inventario.add(producto);
        System.out.println("Producto agregado con éxito.");
    }

    public void mostrarInventario() {
        if (inventario.isEmpty()) {
            System.out.println("El inventario está vacío.");
        } else {
            System.out.println("Inventario:");
            for (Producto producto : inventario) {
                producto.mostrarInfo();
            }
        }
    }

    public Producto buscarProducto(String nombre) {
        for (Producto producto : inventario) {
```

```
            if (producto.nombre.equalsIgnoreCase(nombre)) {
                return producto;
            }
        }
        return null;
    }

    public void realizarVenta(String nombre, int cantidad) {
        Producto producto = buscarProducto(nombre);
        if (producto != null) {
            if (producto.stock >= cantidad) {
                producto.stock -= cantidad;
                System.out.println("Venta realizada. Quedan " + producto.stock + " unidades en
stock.");
            } else {
                System.out.println("No hay suficiente stock para esta venta.");
            }
        } else {
            System.out.println("Producto no encontrado.");
        }
    }
}
```

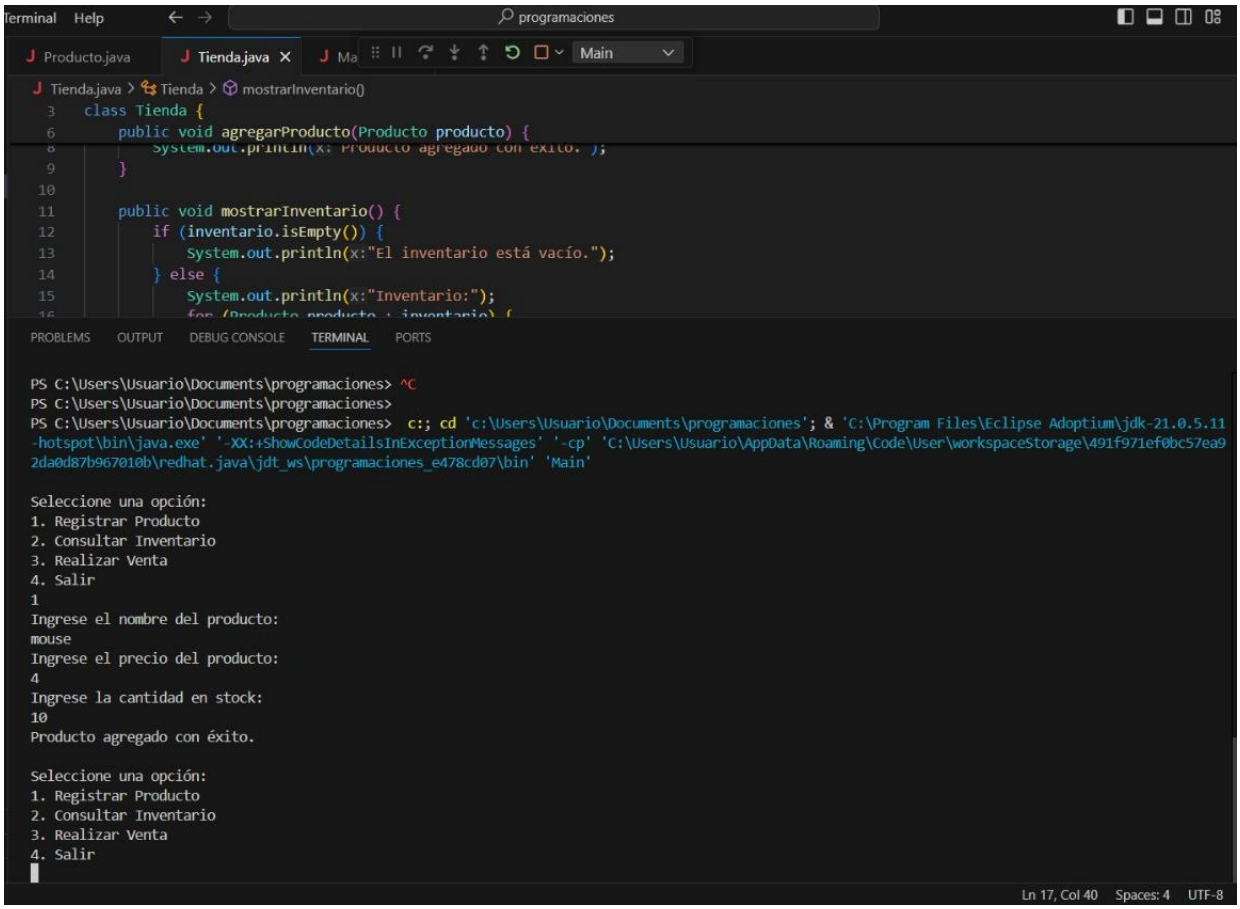

Figura 3: Registro de un producto.

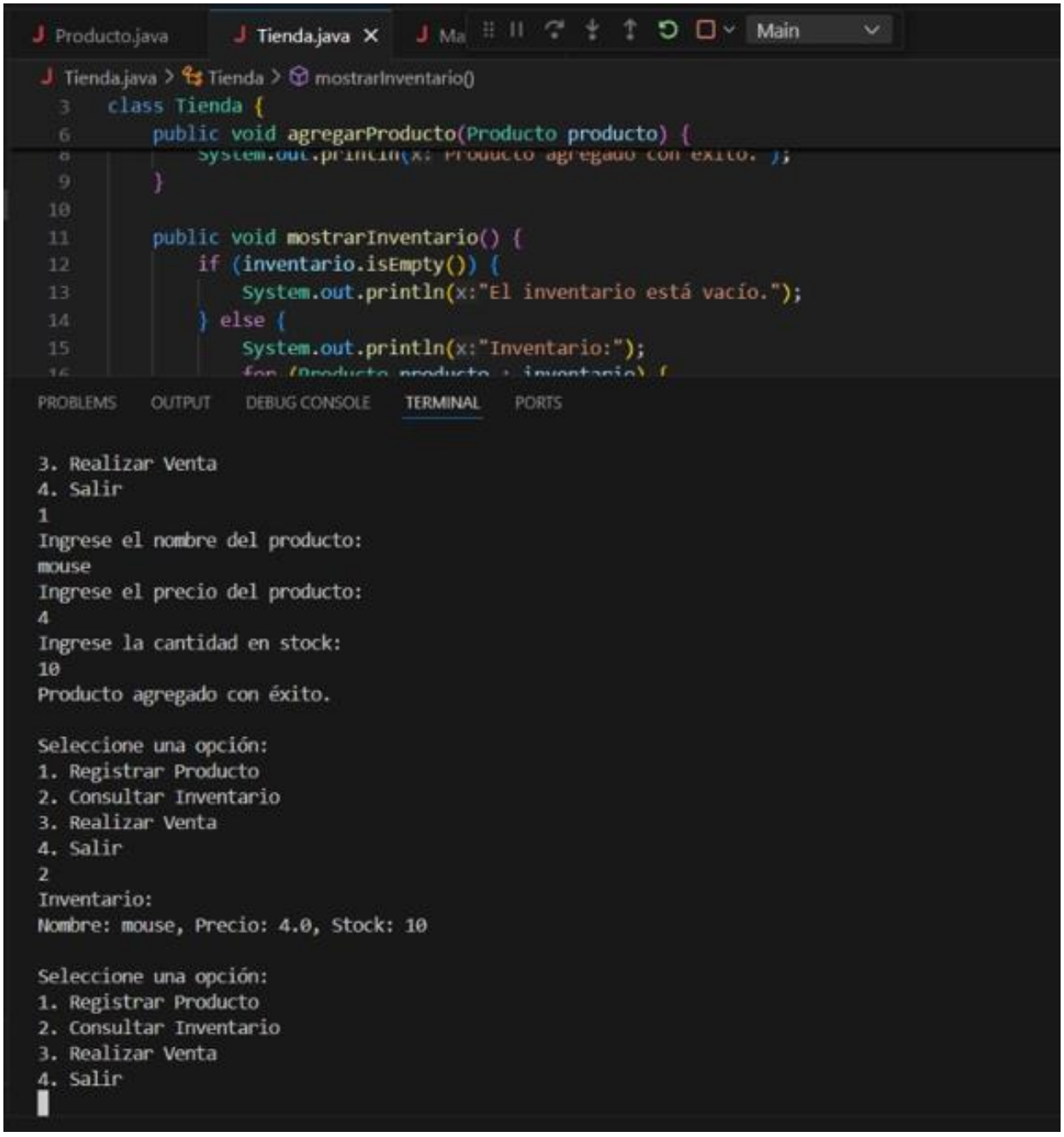

Figura 4: Consulta de inventario.

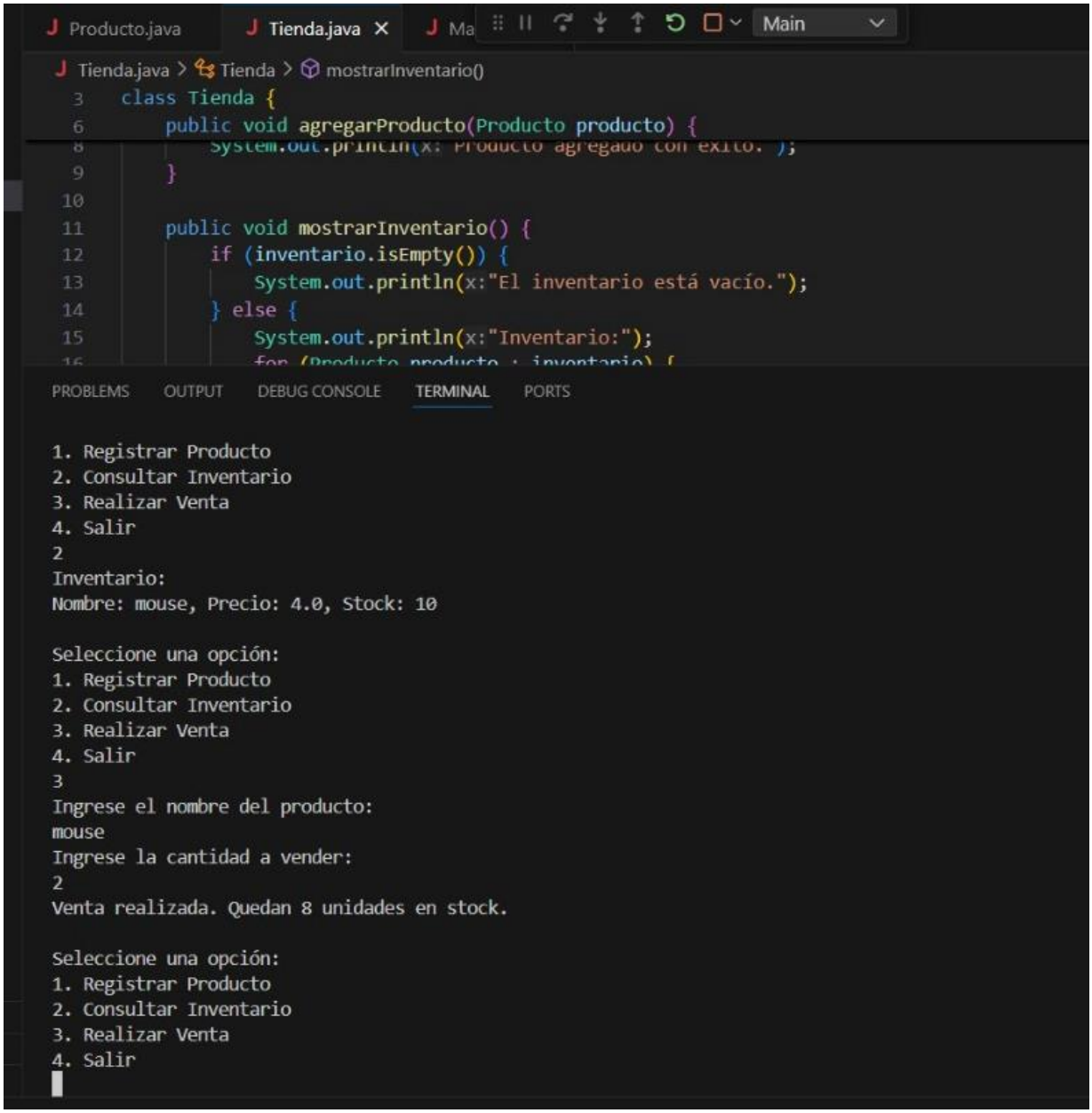

Figura 5:Realizar venta e inventario en stock.

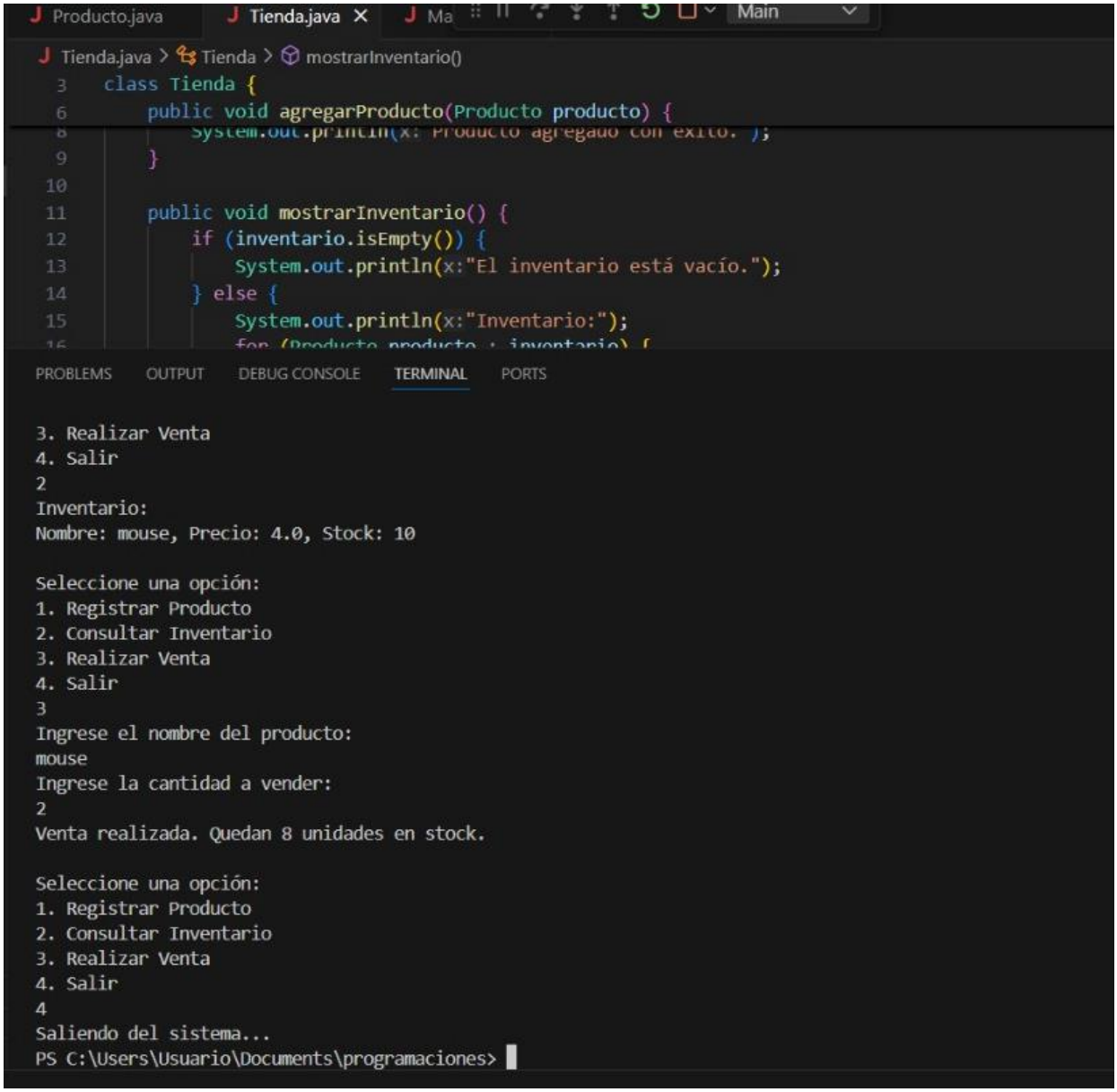

Figura 6: Salir.

Dificultades y Soluciones en el Diseño

Dificultades:

1. Gestión de listas de productos.
2. Validación de datos de entrada del usuario.

Soluciones:

1. Utilizar estructuras de datos adecuadas (Listas) y métodos de búsqueda eficientes.
2. Implementar validaciones y manejo de excepciones para asegurar la correcta entrada de datos.

Conclusiones

El sistema de gestión de tienda de ropa en línea desarrollado proporciona una solución eficiente y organizada para la administración de productos, inventarios y ventas. La implementación de POO facilita la expansión y mantenimiento del sistema. La validación de datos y el manejo de excepciones aseguran la robustez del programa.

Recomendaciones

- Realizar pruebas adicionales para verificar el correcto funcionamiento del sistema en diferentes escenarios.
- Considerar la implementación de una interfaz gráfica para mejorar la experiencia del usuario.
- Explorar la integración con bases de datos para el almacenamiento persistente de la información.

Bibliografía

Blasco, J. L. (2023, November 17). Introducción a POO en Java: Herencia y polimorfismo. Openwebinars.net. https://openwebinars.net/blog/introduccion-a-poo-en-java-herencia-y-polimorfismo/

Blog IfGeekThen. (n.d.). Nttdata.com. Retrieved December 8, 2024, from https://ifgeekthen.nttdata.com/s/post/herencia-en-programacion-orientada-objetos-MCPV3PCZDNBFHSROCCU3JMI7UIJQ?language=es

¿Cómo funcionan las ventas en línea? [GUÍA COMPLETA]. (2022, December 13). Zendesk. https://www.zendesk.com.mx/blog/ventas-en-linea/

Lara, D. (n.d.). Encapsulamiento en la programación orientada a objetos. Styde.net. Retrieved December 8, 2024, from https://styde.net/encapsulamiento-en-la-programacion-orientada-a-objetos/

¿Qué es la gestión de inventarios? (2024, June 10). Ibm.com. https://www.ibm.com/mx-es/topics/inventory-management