



Universidad Técnica Particular de Loja

PROYECTO B1

Programación Orientada a Objetos

RESOLUCIÓN DE PROBLEMAS: PROBLEMA 5

Ingrantes:

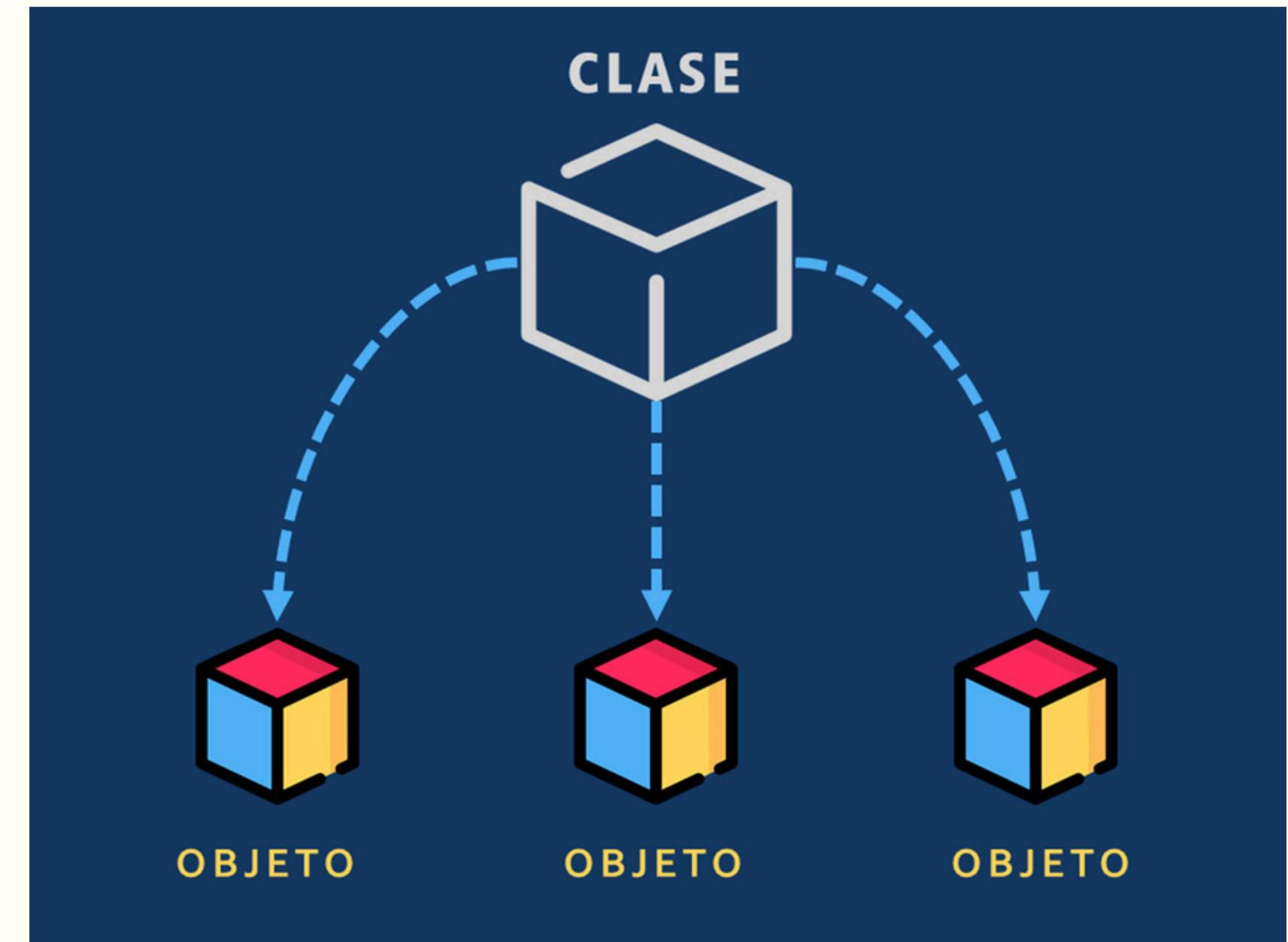
Ariel Cedeño

Diego Loján

Marcelo Pico

INTRODUCCIÓN

Desarrollamos un proyecto orientado a objetos en el que evidenciamos la relevancia de utilizar diagramas UML y buenas prácticas de codificación para mantener un código limpio, estructurado según la arquitectura MVC.



PROBLEMÁTICA

Se nos pidió desarrollar un sistema de facturación para el SuperMaxi de Loja, ya que actualmente no cuentan con una herramienta que gestione adecuadamente el inventario, aplique precios promocionales según el stock o la fecha de caducidad, ni emita facturas que incluyan los impuestos deducibles por categoría de producto. Además, se requiere generar estadísticas diarias de ventas que permitan una mejor toma de decisiones por parte de la gerencia.

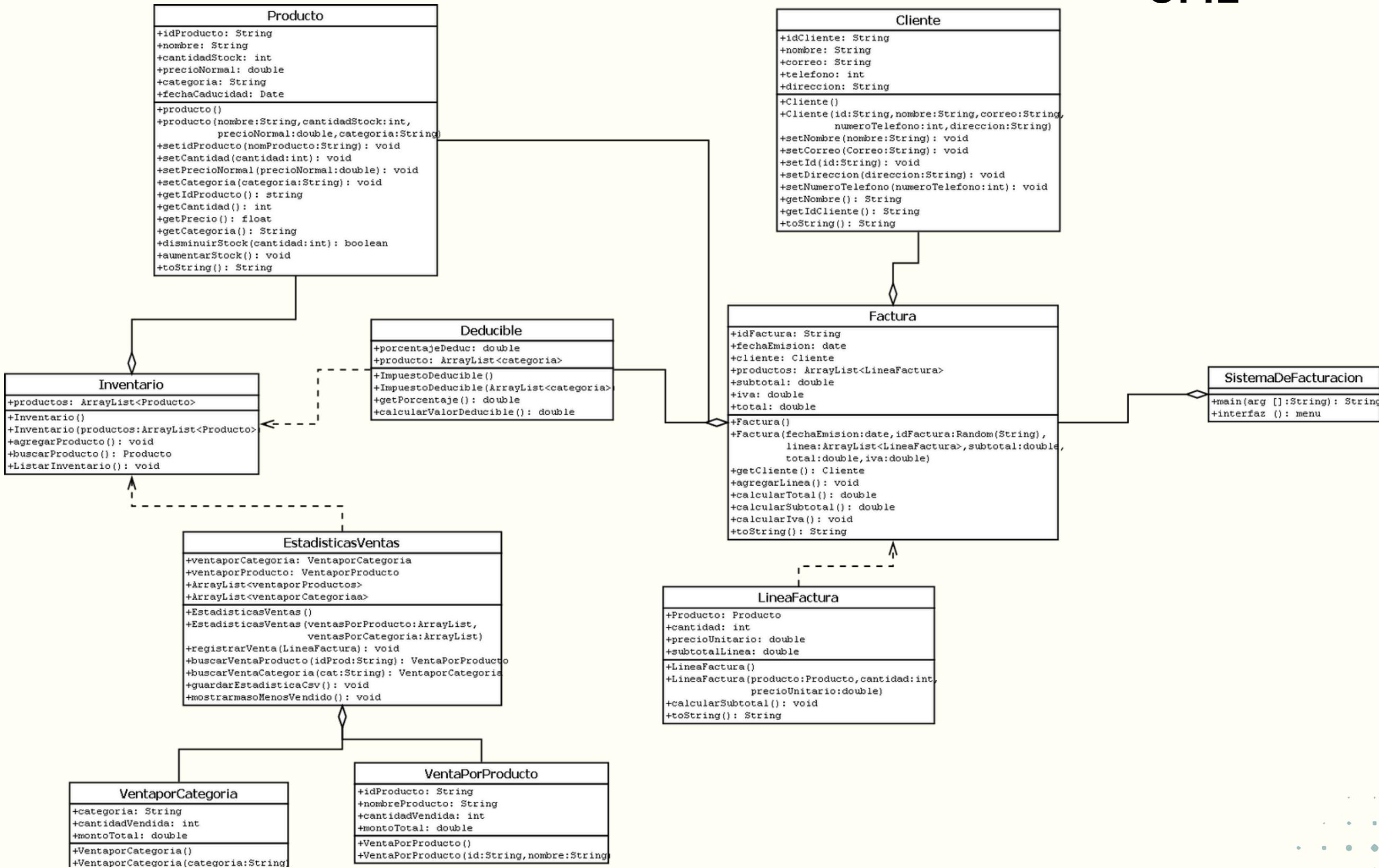


CARACTERISTICAS A CONSIDERAR

- Gestión de productos normales y promocionales.
- Facturación
- Estadísticas de ventas

SOLUCIÓN

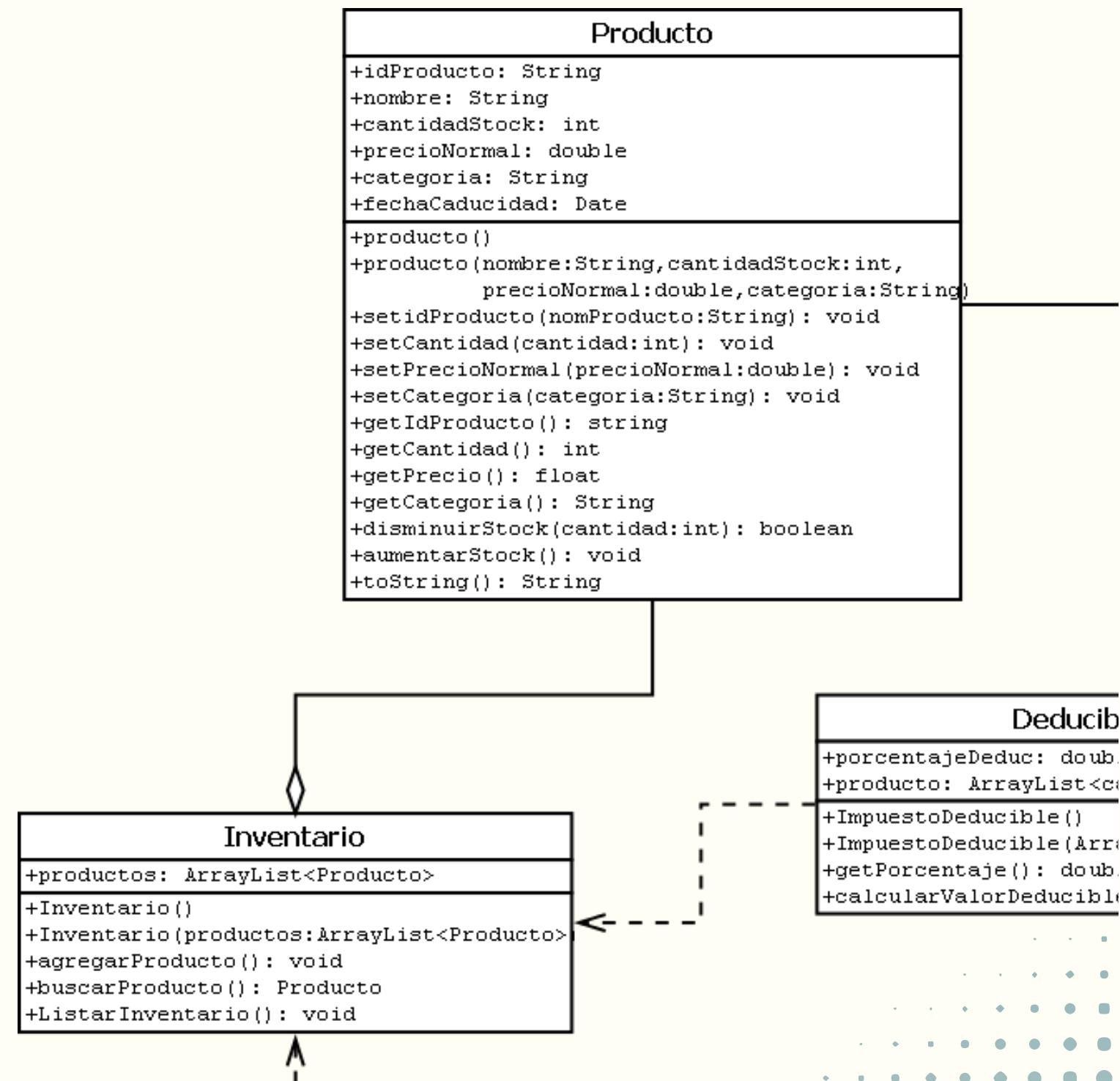
UML



EXPLICACION (GESTIONAMIENTO)

1. Gestión de Productos

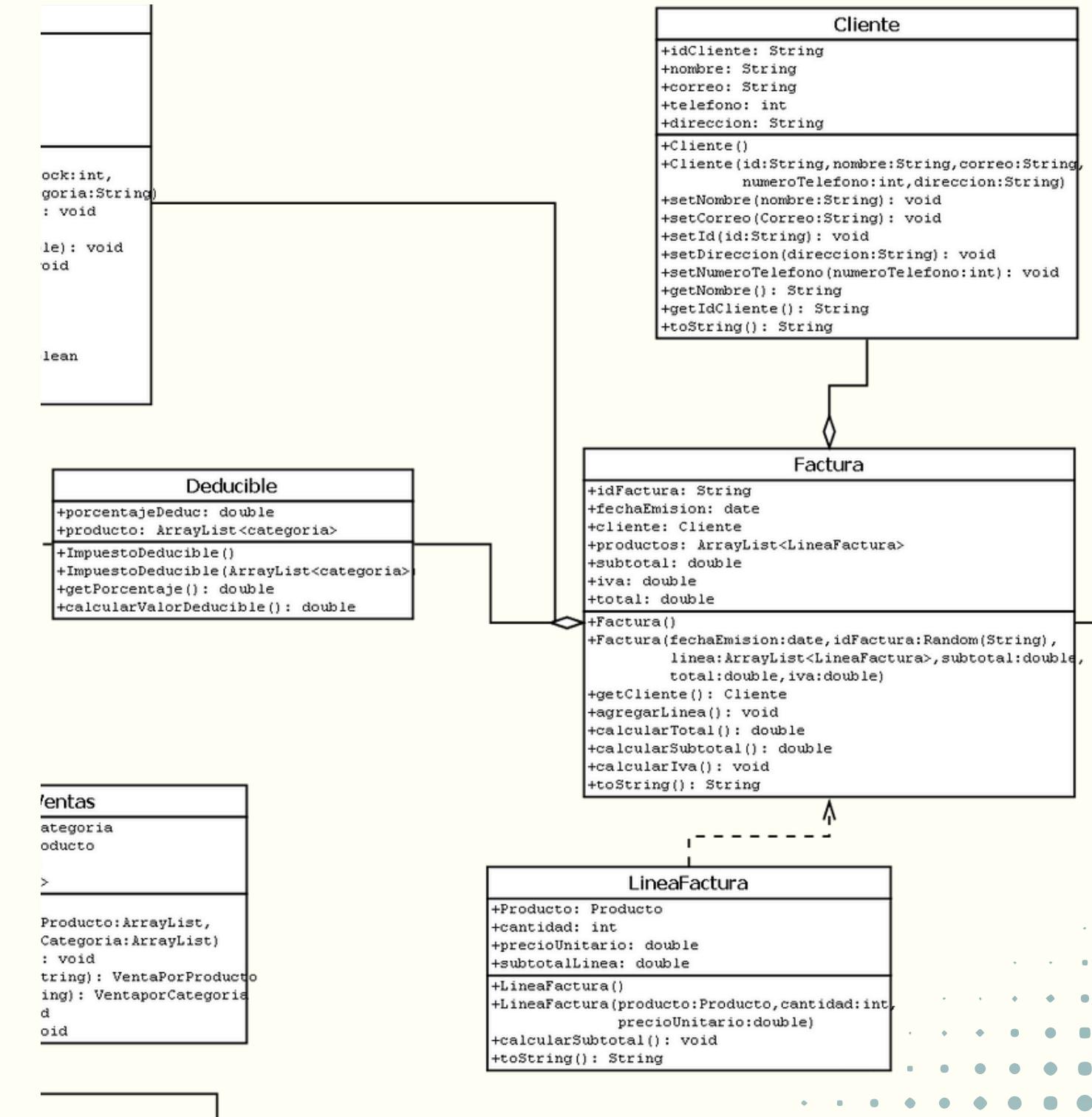
- Clases involucradas:
 - Producto
 - Inventario
- Clase Producto (Modela todos los detalles de un producto)
- Clase Inventario (Gestiona el conjunto de productos): Contiene una lista de Producto.
- Relación: Inventario tiene una relación de composición con Producto.



EXPLICACION (FACTURACIÓN)

2. Facturación

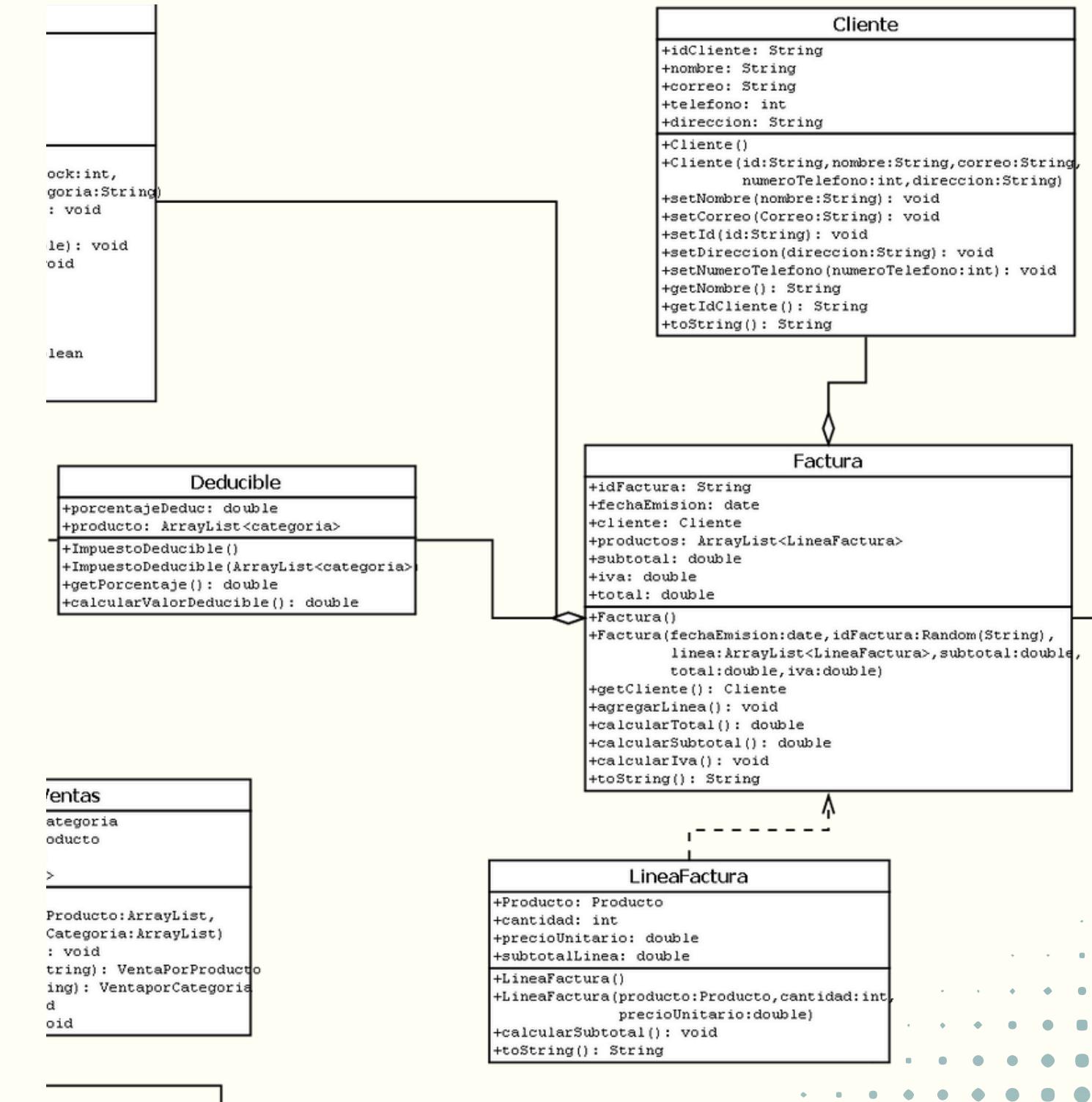
- Clases involucradas:
 - Factura
 - Cliente
 - LineaFactura
 - Deducible
- Clase Factura: Contiene el cliente, una lista de líneas de factura (LineaFactura) y valores como subtotal, iva, total.
- Clase Cliente: Crea y guarda los datos del cliente.
- Clase LineaFactura: Contiene el producto comprado, la cantidad, el precio unitario, y calcula el subtotal.
- Clase Deducible: Gestiona los impuestos deducibles por categoría.
- Relaciones:
 - Factura → Cliente (agregación).
 - Factura → LineaFactura (composición).
 - Deducible interactúa con categorías de productos para deducir impuestos.



EXPLICACION (ESTADISTICA DE VENTA)

3. Estadísticas de Ventas

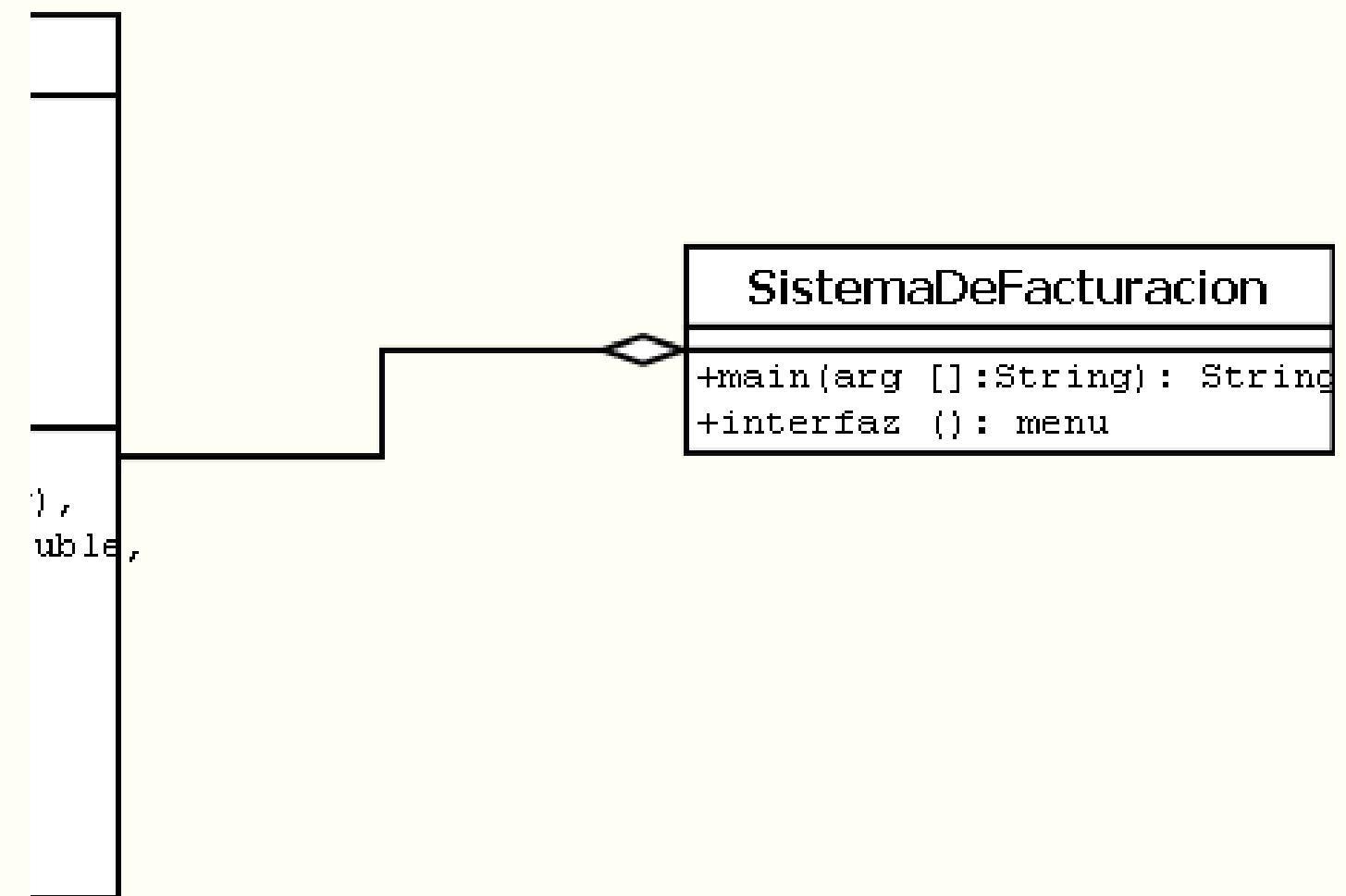
- Clases involucradas:
 - EstadisticasVentas
 - VentaPorProducto
 - VentaPorCategoria
- Clase EstadisticasVentas:
 - Maneja registros de ventas tanto por producto como por categoría.
 - Métodos:
 - registrarVenta()
 - buscarVentaProducto()
 - guardarEstadisticaCSV()
 - mostrarMasVendidos()
- Clases VentaPorProducto y VentaPorCategoria:
- Almacenan información de ventas diarias.
- Relación: EstadisticasVentas utiliza listas de VentaPorProducto y VentaPorCategoria.



EXPLICACION (CLASE CONTROLADORA)

Clase SistemaDeFacturacion:

- Contiene el método main() e interfaz() para mostrar menú o interfaz del sistema.



DESARROLLO DEL CODIGO

MODELO

- Producto
- Cliente
- Factura
- LineaFactura
- inventario
- Deducible
- SistemaDeFacturacion
- EstadisticasVentas
- VentaPorProducto(SubClase)
- VentaPorCategoria(SubClase)

	Cliente.java
	Deducible.java
	EstadisticaVentas.java
	Factura.java
	Inventario.java
	LineaFactura.java
	Producto.java

DESARROLLO DEL CODIGO CONTROLADOR

La clase SistemaController es el controlador principal de un sistema de ventas que maneja:

- Inventario de productos
- Facturación
- Estadísticas de ventas

Trabaja en conjunto con las clases Inventario, Producto, Factura, LineaFactura, Cliente, Deducible y EstadisticaVentas. Además, usa archivos CSV para persistencia de datos.

CONTROLADOR

Método clave: parseDouble(String numeroStr)

- Este método permite convertir un String que representa un número con coma o punto decimal a un double. Es importante porque los CSV pueden tener números en distintos formatos según el sistema operativo o configuración region

CONTROLADOR

-  Gestión de Inventario
 - Carga automática desde CSV con manejo robusto de errores
 - Persistencia de datos entre ejecuciones
 - Visualización completa del inventario
-  Agregar Productos
 - Entrada de datos por consola con validaciones
 - Almacenamiento automático en CSV
-  Facturación Completa
 - Selección de productos con verificación de stock
 - Cálculo automático: subtotal + IVA + total
 - Descuento automático del inventario
 - Persistencia en datosfacturas.csv

 Estadísticas de Ventas
Productos más/menos vendidos
Categorías más populares
Análisis de rendimiento
 Características Técnicas
Modular: Separación clara de responsabilidades
Robusto: Manejo completo de errores
Persistente: Datos CSV para continuidad
Interactivo: Interfaz de consola intuitiva

DESARROLLO DEL CODIGO

VISTA

Esta clase contiene el método main, que actúa como punto de entrada del sistema. Implementa un menú en consola para interactuar con el usuario y ejecutar funciones clave del sistema.

```
SistemaController ctrl = new SistemaController();  
Scanner sc = new Scanner(System.in);
```

Se crea el controlador principal y el lector de entrada (Scanner).
java.

VISTA

```
while (!salir) {  
    ...  
    switch (op) {  
        case "1": ctrl.mostrarInventario(); break;  
        case "2": ctrl.agregarProductoFlujo(sc); break;  
        case "3": ctrl.crearFacturaFlujo(sc); break;  
        case "4": ctrl.mostrarEstadisticas(); break;  
        case "0": salir = true; break;  
        default: System.out.println("[Error] Opción inválida.");  
    }  
}
```

Menú de opciones

1. Ver inventario: Lista todos los productos disponibles.
2. Agregar producto: Llama al flujo para registrar un nuevo producto.
3. Hacer factura: Inicia el proceso de venta/facturación.
4. Ver estadísticas de ventas: Muestra productos más/menos vendidos.
0. Salir: Termina la aplicación.

El menú se repite hasta que el usuario seleccione la opción 0. Cada opción se gestiona con el controlador SistemaController.



MUCHAS GRACIAS.