



UTPL
La Universidad Católica de Loja

EXPOSICIÓN PROYECTO FINAL: PROGRAMACION ORIENTADA A OBJETOS

INTEGRANTES:

- PICO BENITEZ MARCELO DAVID
- LOJAN SISALIMA DIEGO SEBASTIAN

TEMA DE PROYECTOS NRO. 5: SISTEMA DE FACTURACIÓN DEL SUPERMAXI - LOJA

El objetivo del proyecto es desarrollar un sistema de facturación para el SuperMaxi en Loja. Este sistema deberá permitir la facturación de N productos, considerando precios normales y promocionales cuando existan muchos productos en stock o su fecha de caducidad esté próxima. Además, se deberá realizar una factura que resuma los totales de impuestos a la renta deducibles por productos en las siguientes categorías: Vivienda, Educación, Alimentación, Vestimenta y Salud. Al final del día, se generará una estadística de ventas totales, por productos y categorías, que ayudará a los gerentes del SuperMaxi en la toma de decisiones.

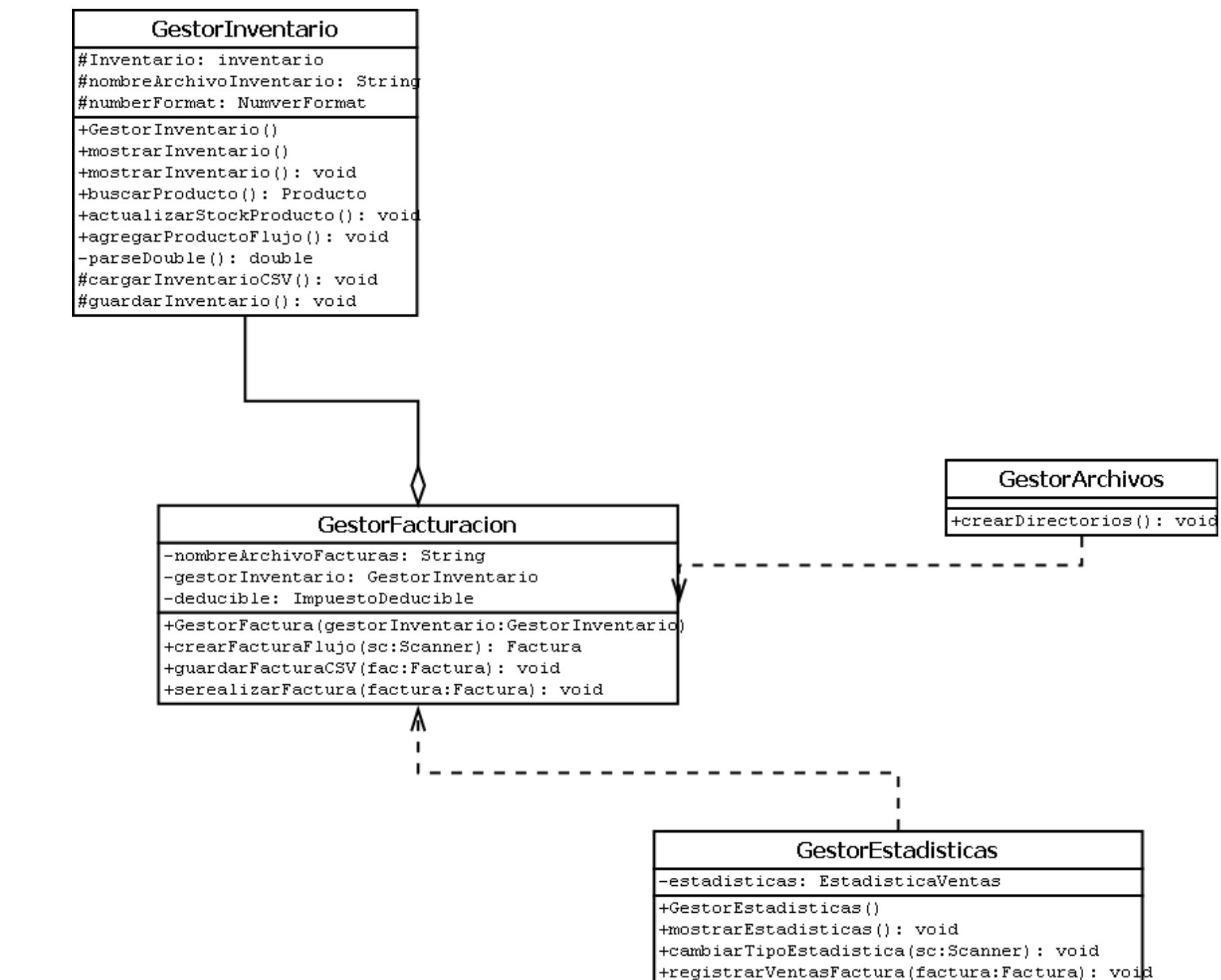
Características por considerar:

- Gestión de productos: Deseñe el modelado para agregar y gestionar productos en el sistema, considerando su cantidad en stock, fecha de caducidad y precios normales y promocionales.
- Facturación: Desarrollar un sistema que calcule el monto total de la factura, teniendo en cuenta los precios normales y promocionales, y que muestre un resumen de los impuestos a la renta deducibles por cada categoría de producto.
- Estadísticas de ventas: Generar estadísticas de ventas diarias, que incluyan las ventas totales y desgloses por productos y categorías, para tomar decisiones gerenciales.

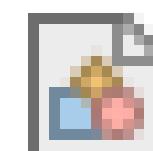
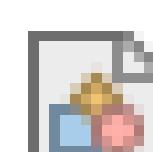
ARQUITECTURA MVC

MODELO

- GestorArchivos
- GestorEstadísticas
- GestorFacturas
- GestorInventario



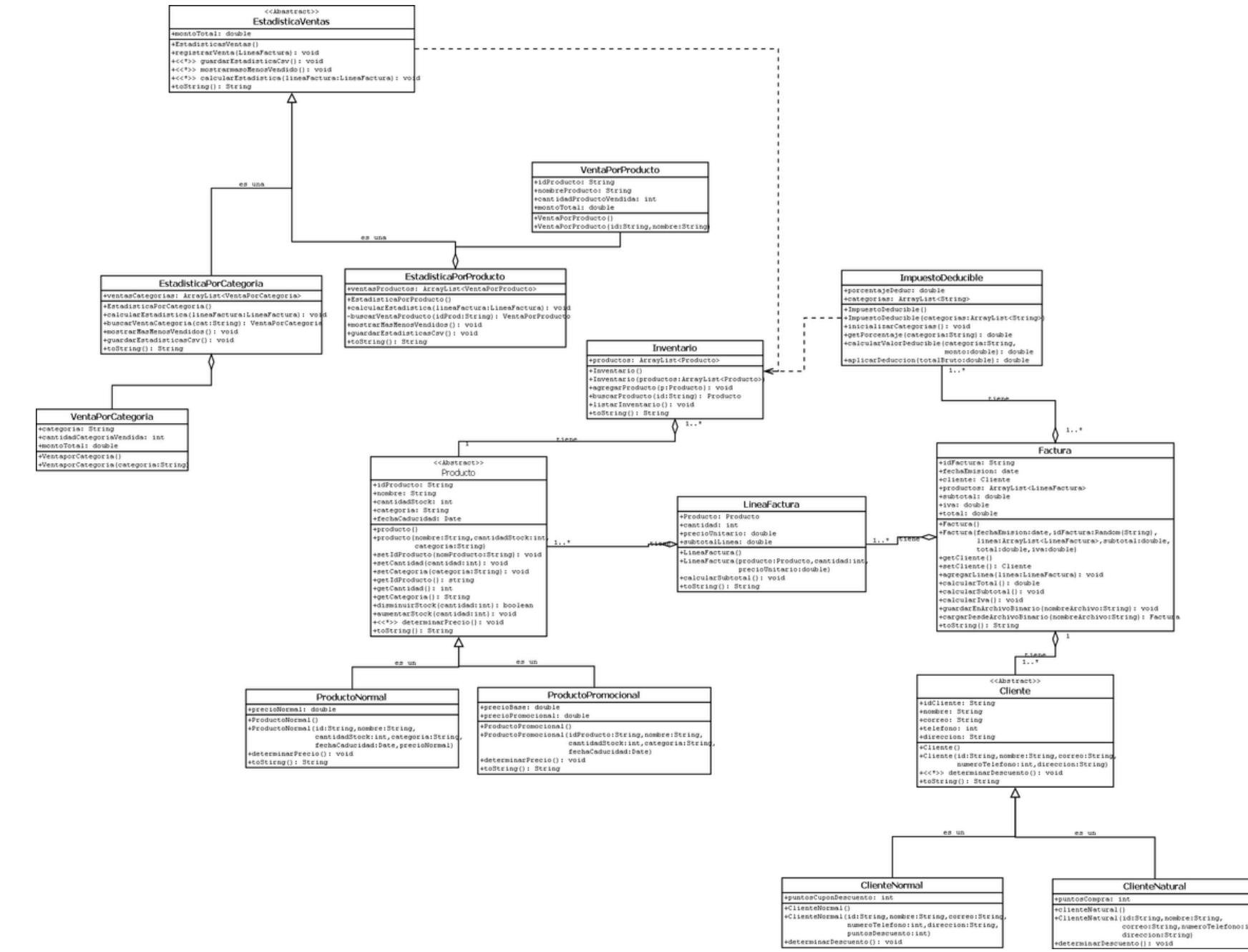
MODELO en JAVA

- ✓  **Modelo**
 -  **GestorArchivos.java [-/A]**
 -  **GestorEstadisticas.java [-/A]**
 -  **GestorFacturacion.java [-/A]**
 -  **GestorInventario.java [-/A]**
 -  **SistemaModelo.java [-/M]**

ARQUITECTURA MVC

Controlador:

- Cliente
- ClienteNatural
- ClienteNormal
- EstadisticaPorCategoria
- EstadisticaPorProducto
- EstadisticaVentas
- Factura
- ImpuestoDeducible
- Inventario
- LineaFactura
- Producto
- ProductoNormal
- ProductoPromocional



CONTROLADOR en JAVA

 Controlador

-  Cliente.java
-  ClienteNatural.java
-  ClienteNormal.java
-  EstadisticaPorCategoria.java [-/M]
-  EstadisticaPorProducto.java
-  EstadisticaVentas.java
-  Factura.java
-  ImpuestoDeducible.java [-/M]
-  Inventario.java [-/M]
-  LineaFactura.java
-  Producto.java
-  ProductoNormal.java
-  ProductoPromocional.java

ARQUITECTURA MVC

Título: Estructura del Sistema

Vista:

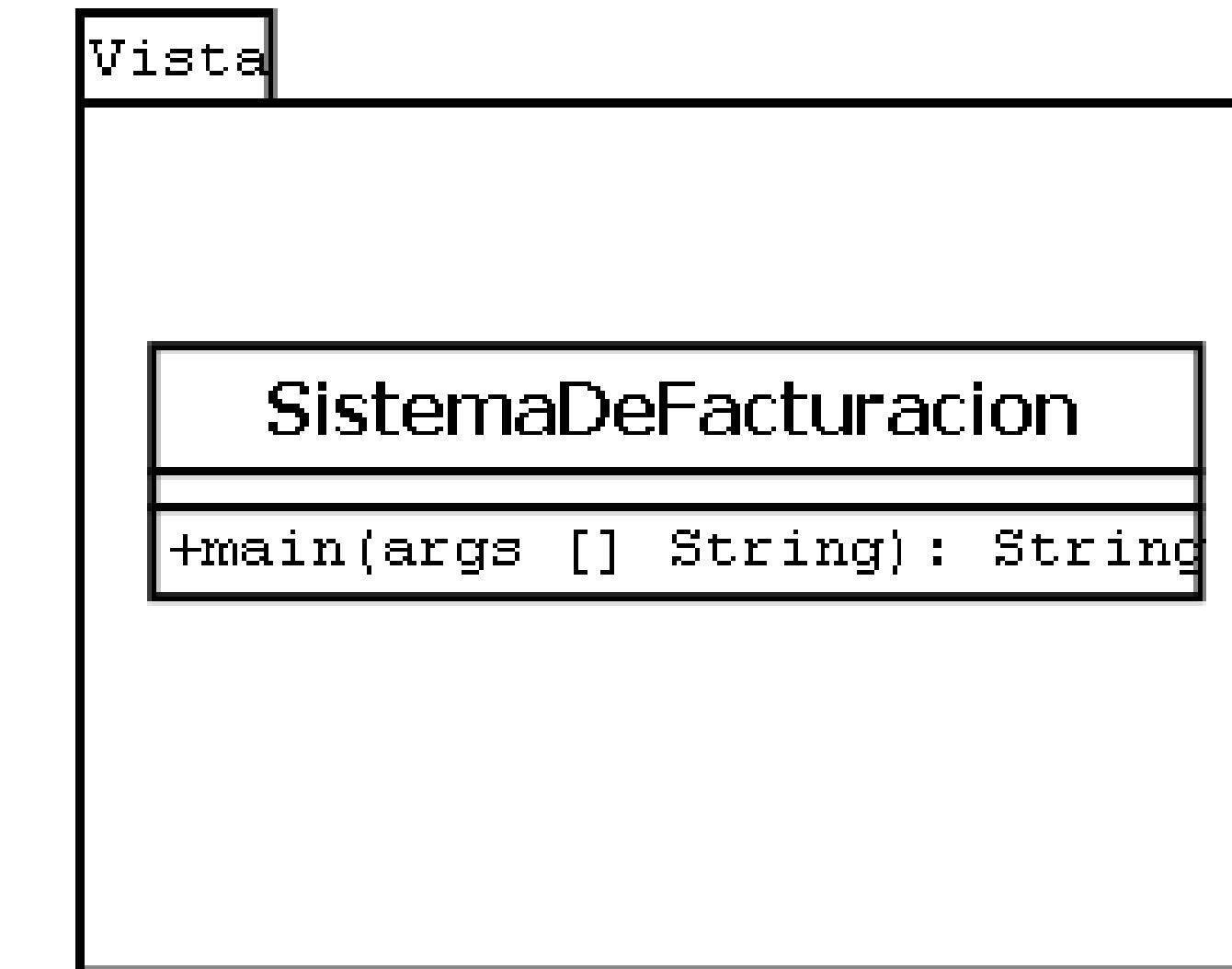
- SistemaDeFacturacion



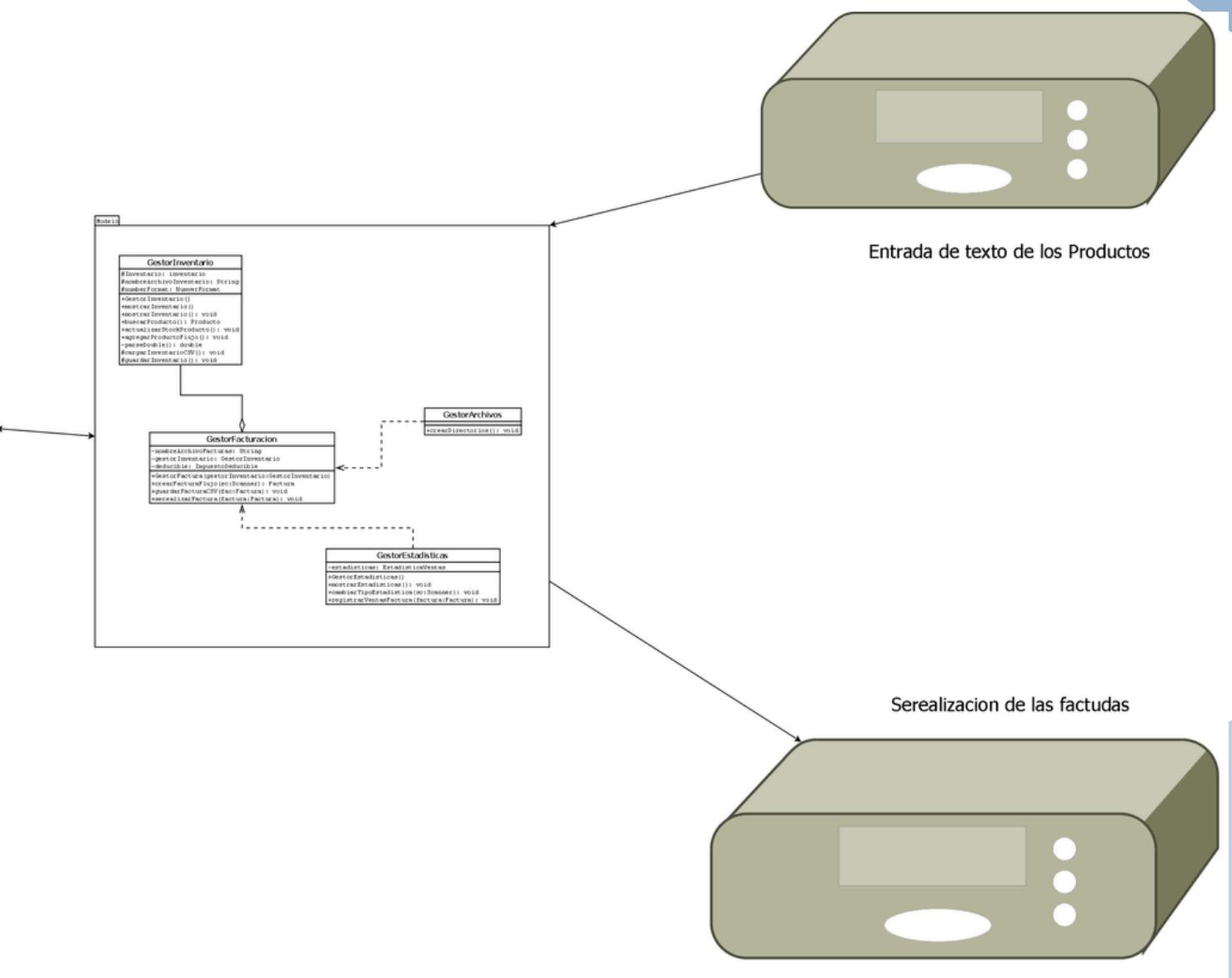
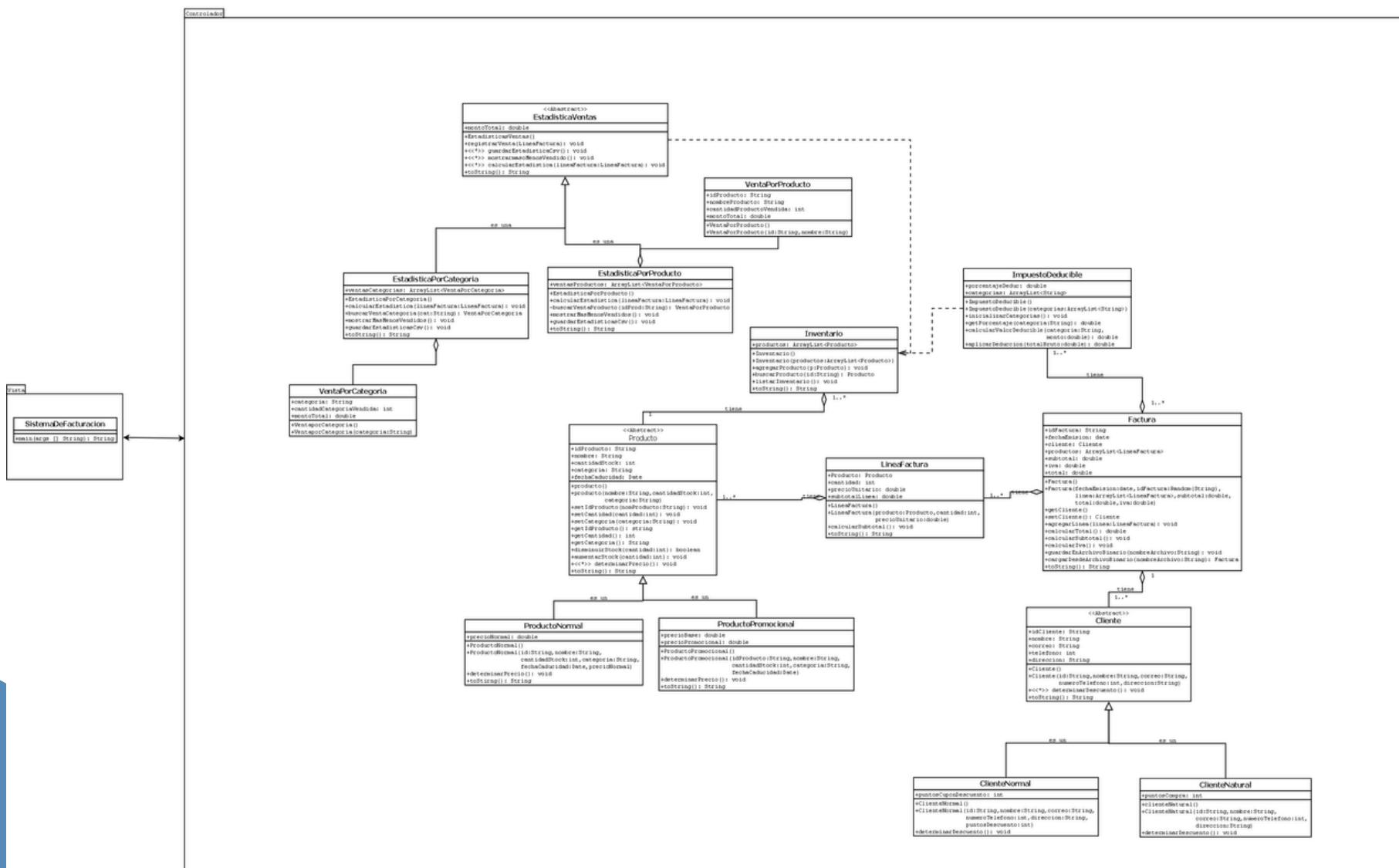
Vista



SistemaDeFacturacion.java [-/M]



ARQUITECTURA MVC

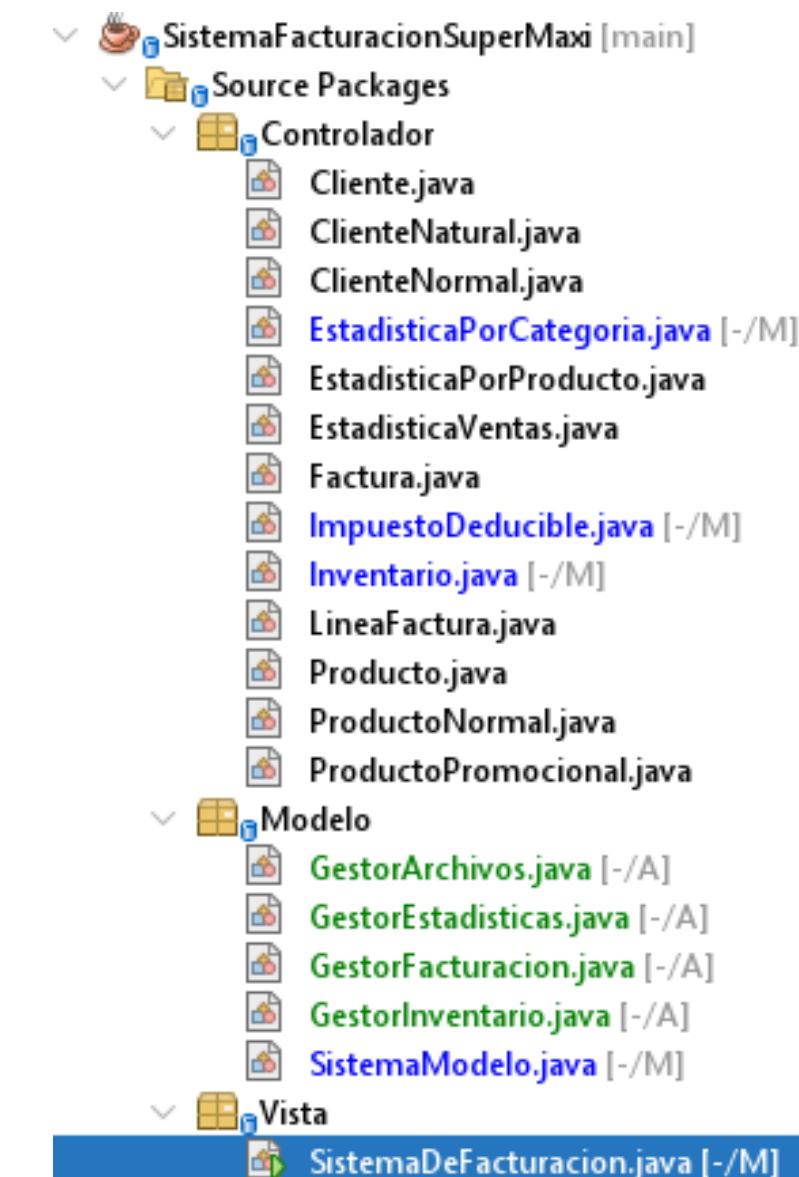


SISTEMA DE FACTURACIÓN MVC CON ARQUITECTURA SOLID

Gestión completa de inventario, facturación y estadísticas

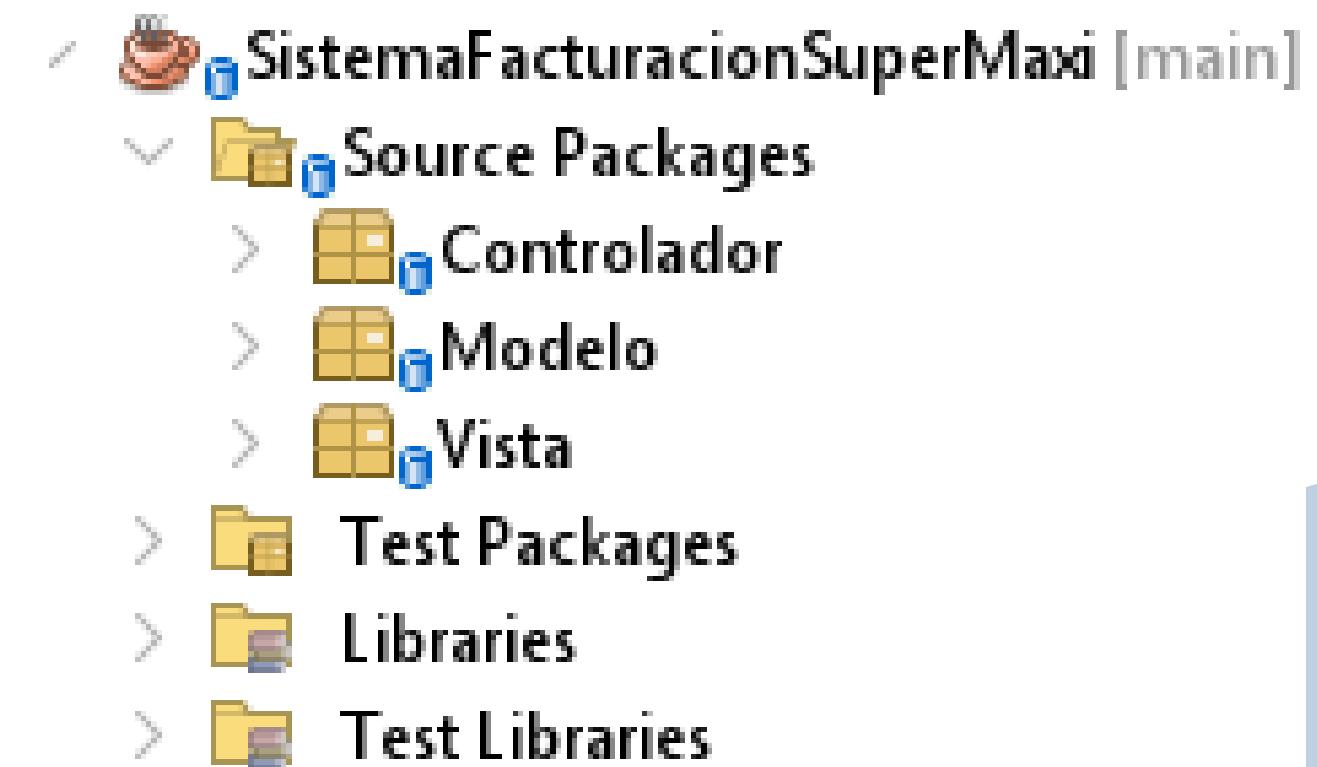
Características principales:

- Registro de productos y clientes
- Generación de facturas con IVA y descuentos
- Estadísticas de ventas por producto/categoría
- Persistencia en archivos CSV y serialización



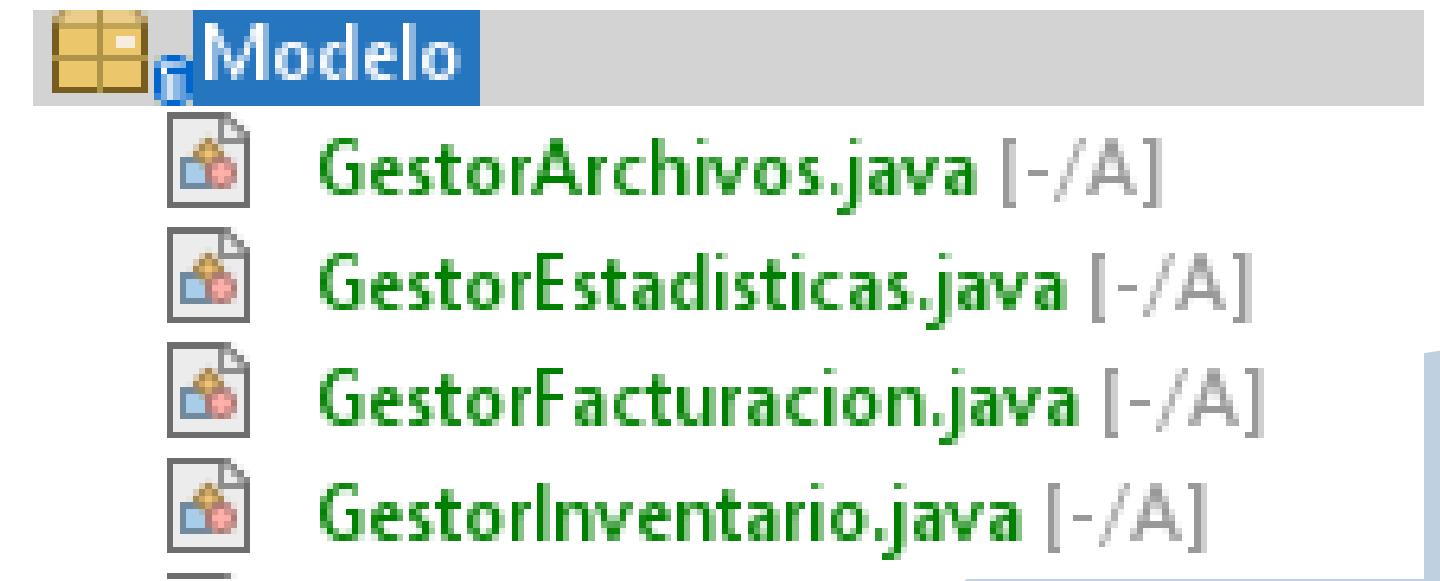
ESTRUCTURA DEL PROYECTO

- Paquetes principales:
 - Modelo: Lógica de negocio (Gestores)
 - Controlador: Clases base y abstractas
 - Vista: Interfaz de usuario (menú principal)
- Principios SOLID aplicados:
 - SRP: Cada gestor tiene una única responsabilidad
 - OCP: Extensible sin modificar código existente
 - LSP: Polimorfismo en productos/clientes
 - ISP: Interfaces segregadas
 - DIP: Dependencias de abstracciones



GESTORES PRINCIPALES

- GestorInventario:
 - Administra productos (normales/promocionales)
 - Carga/guarda en CSV
 - Control de stock
- GestorFacturacion:
 - Crea facturas con clientes (naturales/normales)
 - Calcula subtotal, IVA y total
 - Serializa facturas
- GestorEstadisticas:
 - Registra ventas
 - Genera reportes por producto/categoría



GESTIÓN DE PRODUCTOS

- Jerarquía de Productos:
 - Producto (clase abstracta)
 - ProductoNormal (precio fijo)
 - ProductoPromocional (20% descuento)
- Características:
 - ID único automático
 - Fecha de caducidad aleatoria
 - Precio determinado polimórficamente
 - Gestión de stock



Producto.java



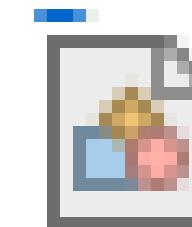
ProductoNormal.java



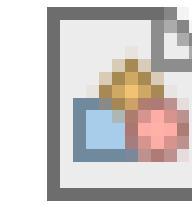
ProductoPromocional.java

GESTIÓN DE CLIENTES

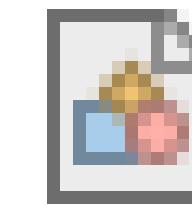
- Jerarquía de Clientes:
 - Cliente (clase abstracta)
 - ClienteNatural (descuento por puntos)
 - ClienteNormal (descuento por cupón)
- Polimorfismo:
 - Cada tipo aplica su propio descuento
 - `determinarDescuento()` implementado
diferentemente



Cliente.java



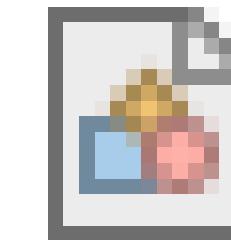
ClienteNatural.java



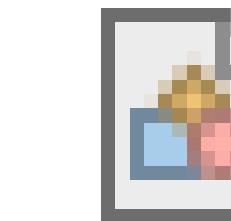
ClienteNormal.java

PROCESO DE FACTURACIÓN

- Registro de datos del cliente
- Selección de productos y cantidades
- Cálculos automáticos:
 - Subtotal
 - IVA (15%)
 - Descuentos (polimórficos)
 - Deducción de impuestos
- Guardado en CSV y serialización



—



—

Factura.java

—

LíneaFactura.java

ESTADÍSTICAS DE VENTAS

- Clases:
 - EstadisticaVentas (clase abstracta)
 - EstadisticaPorProducto
 - EstadisticaPorCategoria
- Funcionalidades:
 - Monto total vendido
 - Producto/categoría más/menos vendido
 - Guardado en CSV específico



[**EstadisticaPorCategoria.java** \[-/M\]](#)



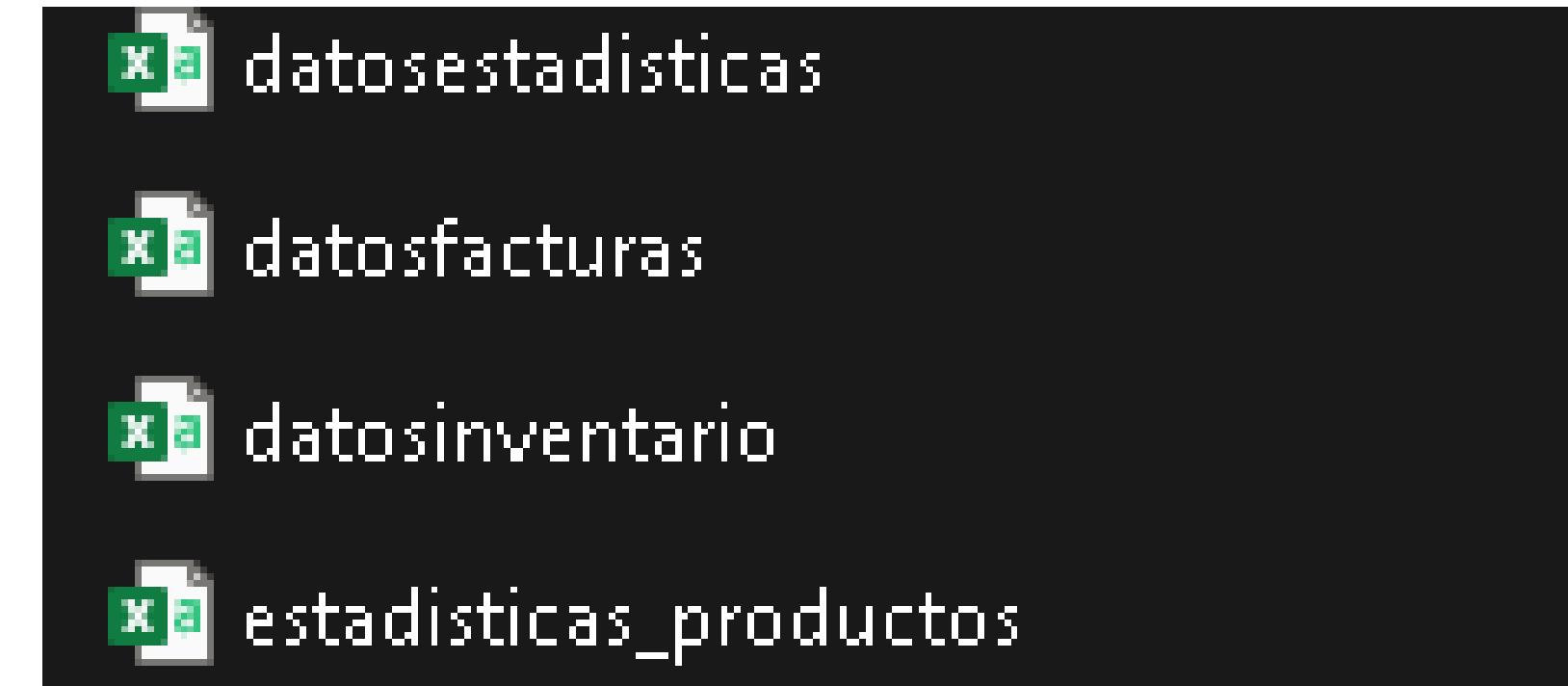
[**EstadisticaPorProducto.java**](#)



[**EstadisticaVentas.java**](#)

PERSISTENCIA DE DATOS

- Formatos soportados:
 - CSV para inventario y facturas
 - Serialización binaria para facturas
- Archivos generados:
 - datosinventario.csv
 - datosfacturas.csv
 - facturas/*.ser
 - estadisticas_*.csv



FLUJO DEL SISTEMA

- Inicialización de gestores
- Menú principal interactivo
- Opciones:
 - Gestión de inventario
 - Facturación
 - Estadísticas
 - Configuración
- Validación de entradas
- Manejo de errores

PRINCIPIOS SOLID EN EL SISTEMA DE FACTURACIÓN

SEO

HTML

PHP

CSS

JS

SRP (SINGLE RESPONSIBILITY PRINCIPLE)

Cada clase tiene una única responsabilidad:

- GestorInventario: Solo gestiona productos (agregar, buscar, actualizar stock).
- GestorFacturacion: Se encarga exclusivamente de crear y guardar facturas.
- GestorEstadisticas: Maneja únicamente el cálculo y visualización de estadísticas.
- GestorArchivos: Solo crea directorios necesarios.

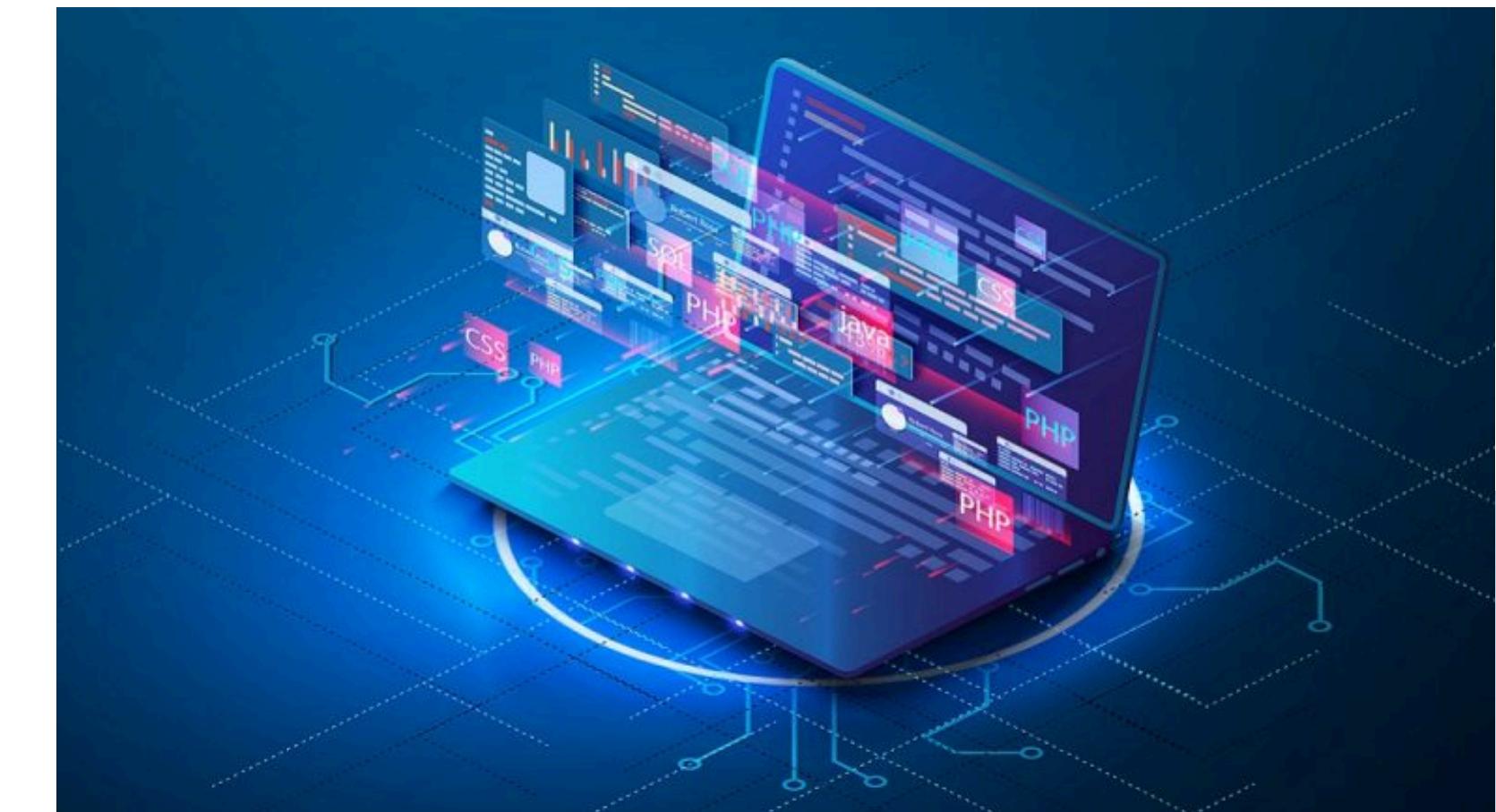
Beneficios

1. Código más fácil de mantener y modificar.
2. Menos acoplamiento entre componentes.

OCP: Abierto para extensión, cerrado para modificación

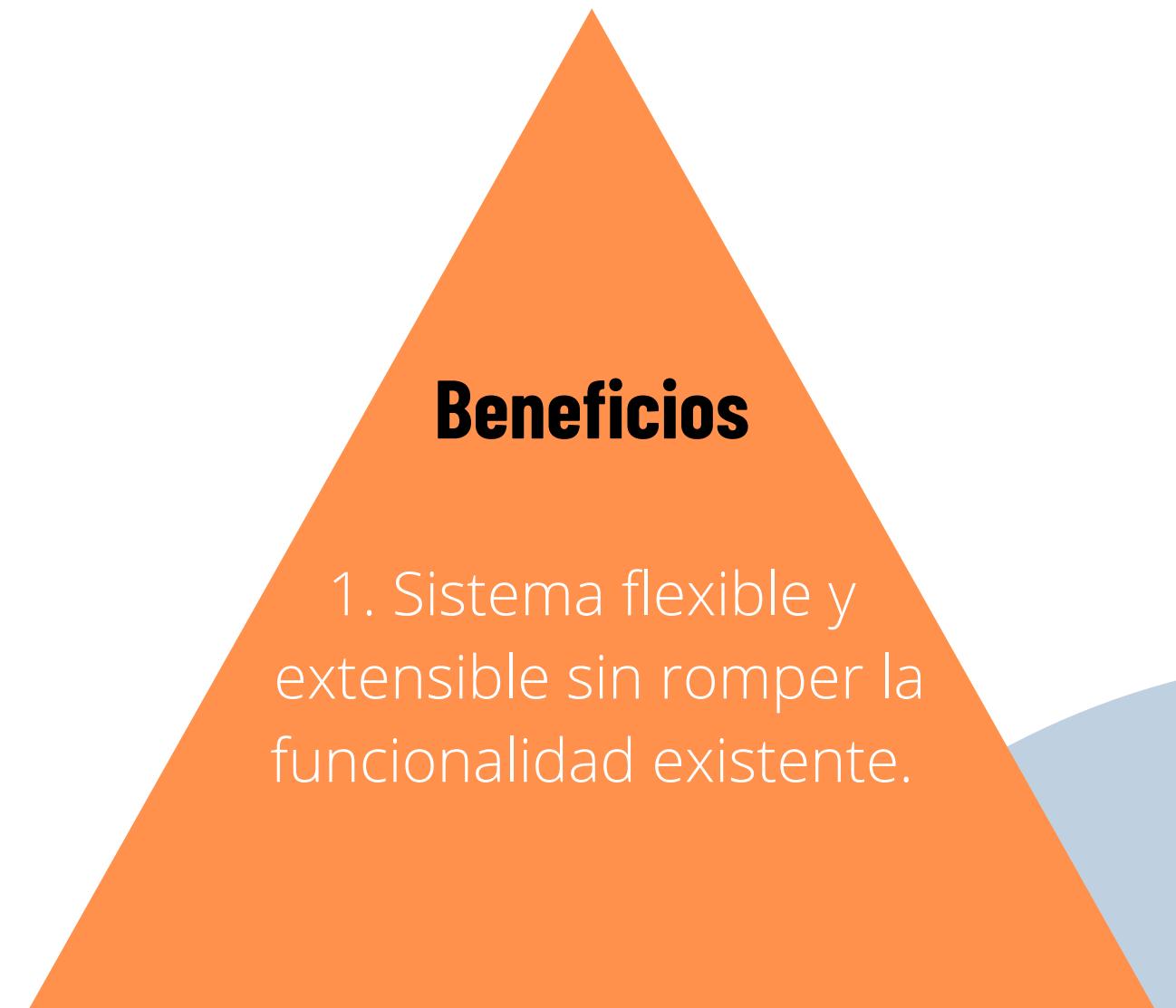
OCP (OPEN/CLOSED PRINCIPLE) & LSP (LISKOV SUBSTITUTION PRINCIPLE)

- EstadisticaVentas es abstracta, pero se puede extender con nuevas estrategias (ej: EstadisticaPorCategoria).
- No se modifica el código existente al agregar nuevas funcionalidades.



LSP: SUBTIPOS REEMPLAZABLES

- ProductoNormal y ProductoPromocional heredan de Producto y pueden usarse indistintamente.
- ClienteNatural y ClienteNormal implementan determinarDescuento() sin romper el contrato base.



ISP (INTERFACE SEGREGATION) & DIP (DEPENDENCY INVERSION)

- EstadisticaVentas define solo métodos necesarios (`calcularEstadistica()`, `mostrarMasMenosVendidos()`).
- No fuerza a implementar métodos no usados.



DIP: DEPENDER DE ABSTRACCIONES, NO DE IMPLEMENTACIONES

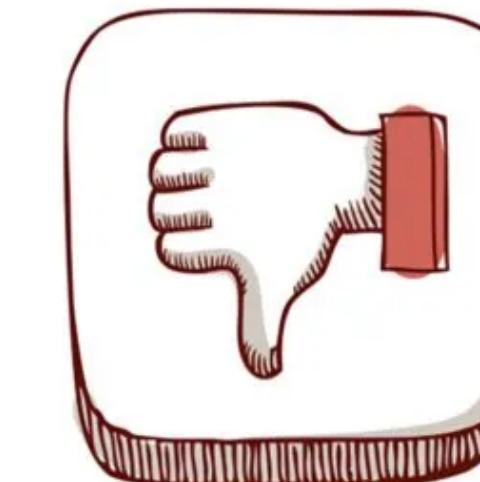
- GestorFacturacion depende de GestorInventario (inyección de dependencia).
- EstadisticaVentas usa abstracción (LineaFactura) en lugar de detalles concretos.

Beneficios

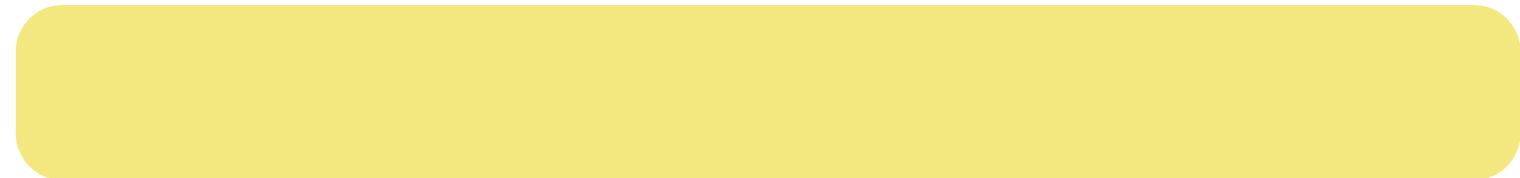
1. Menos acoplamiento, más fácil de testear y modificar.
2. Inversión de control: Las dependencias se gestionan desde fuera (ej: main).

CONCLUSIONES

- Ventajas del diseño:
 - Código modular y mantenable
 - Fácil extensibilidad
 - Separación clara de responsabilidades
 - Persistencia robusta
- Posibles mejoras:
 - Interfaz gráfica
 - Reportes más detallados
 - Integración con base de datos
 - Sistema de autenticación



**iMUCHAS
GRACIAS!**



UTPL
La Universidad Católica de Loja