

TALLER POO CON C#

Caso de Negocios

Inventario Simple Para Biblioteca

JORGE ALEXANDER MARIN

ID 25119011

Asignatura: Programación Distribuida



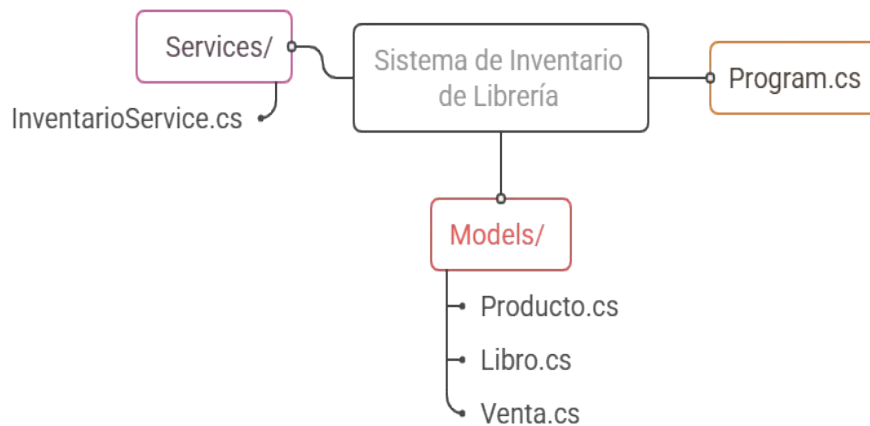
Institución Universitaria

2025-2

Gestión de Inventario Simple Para una Librería

Aplicación de consola para la gestión de inventario y ventas de una librería, usando .NET 8 y los 4 pilares de la Programación Orientada a Objetos.

Estructura del Sistema de Inventario de Librería



Made with Napkin

Introducción

La aplicación "InventarioLibreria" es una solución de consola desarrollada en .NET 8 (C#) que permite a una librería gestionar su inventario y ventas de manera eficiente, aplicando los cuatro pilares de la Programación Orientada a Objetos (POO).

Guion de Sustentación

Problema y Caso de Negocio

Actualmente en la librería "Lectura Viva", el control de inventario y ventas se realiza de forma manual, lo que genera errores, pérdidas de información y dificulta la toma de decisiones.

Caso de negocio: Desarrollar una aplicación de consola que permita gestionar el inventario de libros y el registro de ventas, facilitando la consulta de stock y la generación de reportes confiables.

Historias de Usuario y Criterios de Aceptación

- **Como operador**, quiero agregar, consultar, actualizar y eliminar libros del inventario para mantener la información al día.
 - *Criterio:* El operador puede crear, editar, consultar y borrar libros desde el menú.
- **Como operador**, quiero registrar ventas para descontar el stock y llevar un historial.
 - *Criterio:* Al vender, la cantidad se descuenta y la venta queda registrada.
- **Como supervisor**, quiero ver reportes claros de inventario y ventas para tomar decisiones.
 - *Criterio:* El reporte muestra el stock actual y las ventas realizadas.

Requerimientos funcionales

- RF1: Permitir agregar, consultar, editar y eliminar libros del inventario.
- RF2: Cada libro debe tener: título, autor, ISBN, precio y cantidad en stock.

- RF3: Registrar ventas, descontando stock y guardando fecha, título, cantidad y total vendido.
- RF4: Generar reportes de inventario (stock actual) y ventas (historial de transacciones).
- RF5: La aplicación debe ser de consola y desarrollada en C# (.NET 8).

Requerimientos no funcionales

- RNF1: Interfaz clara y sencilla por consola.
- RNF2: Organización del código en carpetas y archivos; documentación básica.
- RNF3: No requiere persistencia avanzada (puede mejorar en futuras versiones).

Modelo Conceptual

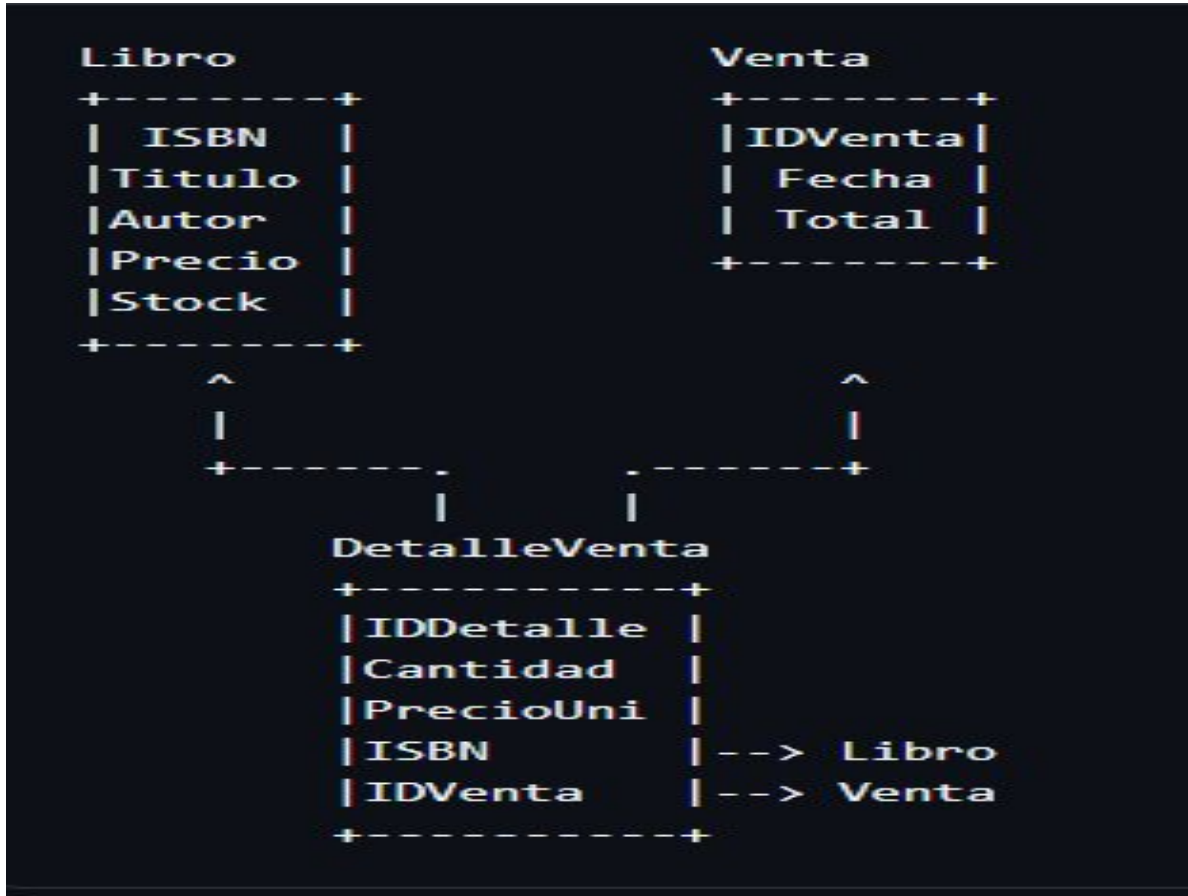
Entidades y Relaciones

- **Libro**
 - ISBN (clave primaria)
 - Título
 - Autor
 - Precio
 - Stock
- **Venta**
 - IDVenta (clave primaria)
 - Fecha
 - Total
- **DetalleVenta**
 - IDDetalle (clave Primaria)
 - Cantidad
 - PrecioUnitario
 - ISBN (relacion con Libro)
 - IDVenta (relacion con Venta)

Relaciones

- Un **Libro** puede aparecer en muchos **DetalleVenta** (uno a muchos).
- Una Venta tiene muchos **DetalleVenta** (uno a muchos).
- Cada **DetalleVenta** corresponde a un Libro y una Venta.

Diagrama ASCII



POO Aplicado

Encapsulamiento:

- Ejemplo: La propiedad `Stock` en la clase `Libro` es privada y se accede/controla mediante métodos públicos.

Herencia:

- Clase `Libro` hereda de la clase abstracta `Producto`, reutilizando y extendiendo atributos/métodos.

Polimorfismo:

- Método 'MostrarInfo()' sobrescrito en 'Libro', permite mostrar información específica de cada tipo de producto.

Abstracción:

- Clase abstracta `Producto` define los atributos y métodos base para productos; método abstracto `MostrarInfo()` obliga a las subclases a implementar su propia presentación.

Demo en Vivo

1. Crear entidad (Agregar libro):

- Ingreso de datos por consola (título, autor, ISBN, precio, stock).
- Mensaje de confirmación: "Libro agregado exitosamente."

2. Asignar y listar:

- Consultar libros muestra una lista formateada con los detalles de cada libro.

3. Registrar venta y manejo de errores:

- Ingreso de ISBN y cantidad; si el ISBN no existe o el stock es insuficiente, muestra mensaje de error adecuado.
- Si la venta es exitosa, descuenta el stock y registra la operación.

4. Mostrar reportes:

- Reporte de inventario y ventas en consola.
- Ejemplo de error: intentar vender más unidades de las disponibles o un ISBN inexistente.

5. Pruebas y Calidad:

- Pruebas de consola para cada funcionalidad: agregar, actualizar, eliminar, vender, reportar.
- Validaciones implementadas:
 - No permite stock negativo ni ventas superiores al stock disponible.
 - Verificación de existencia de ISBN para operaciones.
- Código modular y documentado, facilitando mantenimiento y futuras mejoras.
- Estructura clara en carpetas: modelos, servicios, programa principal.

6. Cierre:

Con esta aplicación, la librería resuelve el problema de gestión de inventario y ventas, reduce errores y facilita decisiones, aplicando buenas prácticas de POO y calidad de software.

Plan de Pruebas

1. Agregar Libro

Caso de uso: El usuario agrega un nuevo libro al inventario.

-Datos de prueba: ISBN: 9781234567890

- Título: Cien Años de Soledad
- Autor: Gabriel García Márquez
- Precio: \$20.00
- Stock: 10

Resultado esperado: El libro aparece en la lista de inventario.

2. Consultar Inventario

- Caso de uso: El usuario consulta la lista de libros disponibles.

- Datos de prueba: Inventario con varios libros agregados.

Resultado esperado: Se muestra la lista completa con los datos correctos de cada libro.

3. Actualizar Libro

Caso de uso: El usuario actualiza información de un libro existente.

- Datos de prueba: ISBN: 9781234567890

- Nuevo stock: 15

Resultado esperado: El stock del libro se actualiza a 15.

4. Eliminar Libro

- Caso de uso: El usuario elimina un libro del inventario.

- Datos de prueba: ISBN: 9781234567890

Resultado esperado: El libro deja de aparecer en el inventario.

5. Registrar Venta

- Caso de uso: El usuario registra la venta de uno o más libros.

- Datos de prueba: IDVenta: 1

- Fecha: 2025-09-11

- Libros vendidos:

- ISBN: 9781234567890, Cantidad: 2

Resultado esperado: El stock del libro disminuye en 2 unidades. Se registra la venta con el total calculado.

6. Reporte de Ventas

- Caso de uso: El usuario consulta el reporte de ventas.

- Datos de prueba: Varias ventas registradas.

Resultado esperado: Se muestra el listado de ventas con fecha, libros vendidos, y montos totales.

7. Venta con Stock Insuficiente

Caso de uso: Intentar vender más unidades de un libro de las que hay en stock.

- Datos de prueba: Stock actual: 3
- Cantidad a vender: 5

Resultado esperado: El sistema muestra un mensaje de error y no permite la venta.

8. Buscar Libro por Autor/Título

- Caso de uso: El usuario busca libros por autor o título.
- Datos de prueba: Autor: Gabriel García Márquez
- Título: Cien Años de Soledad

Resultado esperado: Se muestran solo los libros que coinciden con la búsqueda.

9. Agregar Libro Duplicado

- Caso de uso: Intentar agregar un libro con un ISBN que ya existe.
- Datos de prueba: ISBN: 9781234567890 (ya existente)

Resultado esperado: El sistema muestra un mensaje de error y no permite duplicados.

Con esta aplicación, la librería resuelve el problema de gestión de inventario y ventas, reduce errores y facilita decisiones, aplicando buenas prácticas de POO y calidad de software.
