Universidad EAN

Taller 3

Esteban Mateo González Ruiz

Cristian David Ramos Corredor

Programación Orientada a Objetos

Docente: Jhan Yuler de la Pava Torres

Bogotá D.C, Colombia

2025

Tabla de contenido

[Descripción del Contexto 3](#_Toc197123996)

[Definición del Problema o Necesidad 3](#_Toc197123997)

[Justificación de la Solución 4](#_Toc197123998)

[¿Por qué es importante esta solución? 5](#_Toc197123999)

[Impacto cuantificable 5](#_Toc197124000)

[Propuesta de Solución 5](#_Toc197124001)

[Principales funcionalidades 6](#_Toc197124002)

[Modelado de Clases 6](#_Toc197124003)

[Reflexión SOLID 7](#_Toc197124004)

[Diseño de la persistencia 8](#_Toc197124005)

[Tipo de Archivo Utilizado: 8](#_Toc197124006)

[Proceso de Escritura 8](#_Toc197124007)

[Proceso de Lectura: 8](#_Toc197124008)

# Descripción del Contexto

La miscelánea "El Nevado" es un negocio local que desde el 2021 ha brindado a su comunidad una amplia gama de productos, incluyendo material escolar, artículos de oficina y otros insumos de papelería. Con el paso del tiempo, el negocio ha ganado popularidad gracias a su ubicación estratégica y su extenso catálogo. Su clientela es diversa: desde estudiantes en busca de útiles escolares hasta profesionales que requieren herramientas para su trabajo diario.

Sin embargo, a pesar del crecimiento del negocio, la gestión sigue siendo manual, lo que ha traído consigo varios desafíos. Actualmente, el control del inventario se lleva a cabo de forma tradicional, lo que ha provocado errores, retrasos y pérdidas económicas. En muchas ocasiones, los empleados tienen dificultades para saber cuántos productos hay en stock, lo que ocasiona problemas al atender a los clientes y tomar decisiones sobre nuevas compras. Además, el tiempo que se pierde en buscar productos o verificar precios afecta directamente la calidad del servicio.

En este contexto, se vuelve evidente la necesidad de modernizar la gestión del negocio mediante la implementación de un sistema automatizado que agilice estos procesos y haga que la administración de "El Nevado" sea más eficiente y organizada.

# Definición del Problema o Necesidad

El principal problema que enfrenta "El Nevado" es la falta de un sistema automatizado para el control del inventario. Este problema se manifiesta en diferentes aspectos:

* **Errores en el inventario:** Al no contar con un registro digitalizado, muchas veces hay discrepancias entre la cantidad real de productos en stock y la información registrada manualmente. Esto puede llevar a que se realicen compras innecesarias o que falten productos esenciales.
* **Demoras en la atención al cliente:** La búsqueda manual de productos en el almacén o la verificación de precios en listas impresas consume tiempo valioso, generando largas esperas para los clientes.
* **Pérdidas económicas:** La falta de control sobre los productos puede derivar en pérdidas, ya sea por extravío, deterioro o vencimiento de artículos que no se venden a tiempo.
* **Dificultad en la toma de decisiones:** Sin datos claros sobre las ventas y el inventario, la administración del negocio se vuelve más compleja y se dificulta la planificación de compras o la implementación de estrategias de ventas efectivas.

Todos estos factores afectan la rentabilidad y la operatividad del negocio, por lo que es urgente encontrar una solución que optimice la gestión de "El Nevado" y facilite su crecimiento.

# Justificación de la Solución

Para resolver estos problemas, se propone la implementación de un sistema automatizado que permita gestionar el inventario de manera más eficiente. Esta solución traerá múltiples beneficios tanto para la administración del negocio como para los clientes.

## ¿Por qué es importante esta solución?

* **Mejor control del inventario:** Se podrá llevar un registro actualizado de los productos disponibles, evitando problemas de desabasto o exceso de stock.
* **Agilización de las ventas:** Un sistema digitalizado reducirá el tiempo de facturación, permitiendo que los clientes sean atendidos más rápido.
* **Reducción de errores:** Al automatizar el registro de productos y ventas, se minimizarán los errores humanos que pueden causar pérdidas económicas.
* **Mejor experiencia para el cliente:** Con una atención más rápida y eficiente, los clientes tendrán una mejor percepción del negocio, lo que fomentará su fidelización.

## Impacto cuantificable

* Se estima que la reducción de pérdidas por errores en el inventario podría representar un ahorro del 20%.
* La digitalización del proceso de ventas permitiría reducir en un 30% el tiempo de facturación, lo que aumentaría la capacidad de atención al público y, potencialmente, las ventas diarias.

## Propuesta de Solución

Para abordar esta problemática, se plantea el desarrollo de una aplicación en Java que permita automatizar la gestión de inventario y ventas en "El Nevado".

**¿Cómo funcionará la aplicación?**

1. Permitirá el registro automático de productos, con actualización en tiempo real del stock.
2. Generará facturas de forma digital y llevará un historial de ventas.
3. Proporcionará reportes detallados sobre productos vendidos y stock disponible.
4. Facilitará la búsqueda de productos mediante filtros por nombre, categoría o precio.
5. Enviará alertas cuando un producto esté próximo a agotarse o a vencer.

## Principales funcionalidades

- Gestión de Inventario: Ver, agregar, modificar y registrar movimiento de productos.  
- Gestión de Ventas: Registrar ventas y calcular ganancias.  
- Gestión de Personas: Administración de clientes y empleados.  
- Persistencia de Datos: Guardado y recuperación de inventario y productos mediante archivos.  
- Interfaz de Usuario: Menú sencillo y navegación a través de consola.

# Modelado de Clases

|  |  |
| --- | --- |
| Clase | Responsabilidad |
| Main | Lógica principal y menú de interacción con el usuario. |
| Producto | Representa un producto de la tienda. |
| Cliente | Representa un cliente que compra productos. |
| Empleado | Representa un empleado que maneja el inventario. |
| Persona | Clase base general para Cliente y Empleado. |
| Inventario | Controla los productos disponibles y su cantidad. |
| ItemInventario | Relaciona un Producto con su estado de inventario. |
| Venta | Representa una venta individual de productos. |
| Ventas | Maneja una lista de todas las ventas realizadas. |
| PersistenciaProductos | Guarda y carga datos de productos desde archivos. |
| PersistenciaInventario | Guarda y carga datos del inventario desde archivos. |
| Mostrable | Interfaz para clases que deben implementar métodos de visualización. |
| InfoConsola | Utilidades para mostrar información de inventario y ventas en consola. |

# Reflexión SOLID

En el proceso de realizar la automatización del inventario se ha evidenciado que Cada clase en el módulo tiene una única responsabilidad. Por ejemplo, la clase Producto se encarga exclusivamente de representar las propiedades de un producto, mientras que la clase Inventario gestiona la colección de productos. Esta separación facilita el mantenimiento y la comprensión del código.

También el diseño permite la extensión de funcionalidades sin modificar las clases existentes. Por ejemplo, si se desea agregar un nuevo tipo de producto con características adicionales, se puede crear una subclase de Producto que extienda su funcionalidad, respetando así el principio de Abierto/Cerrado

# Diseño de la persistencia

## Tipo de Archivo Utilizado:

* Archivos de texto plano  
  Se utilizan archivos .txt o .csv como base de datos local para guardar la información de los productos disponibles en la papelería. Este tipo de archivo es fácil de manipular y no requiere dependencias externas ni bases de datos complejas.

## Proceso de Escritura

* Cada vez que se añade un nuevo producto o se modifica el inventario el sistema:
  1. Abre el archivo de texto en modo escritura.
  2. Escribe los datos del producto en una línea, separados por delimitadores fijos (en este caso por |).
  3. Guarda y cierra el archivo para asegurar la persistencia de la información.

## Proceso de Lectura:

* Al iniciar la aplicación o cargar el inventario, el sistema:
  1. Lee el archivo línea por línea.
  2. Parsea cada línea dividiéndola por el delimitador para extraer los atributos del producto (nombre, precio, cantidad, etc.)
  3. Crea objetos de tipo producto o de la clase correspondiente con estos datos y los agrega a una lista en memoria