



# Sistema de Gestión Supermercado Pali

Presentado por: Donald Ronaldo Romero Cuadra, Geanfranco Alexander Rodriguez Mendieta, Anthony Sying González Chow, Jairo De Jesús Rojas Mojica.

Docente: Silvia Gigdalia Ticay López

Managua, 8 de julio de 2025



# Introducción: Un Desafío en el Comercio Minorista

En el dinámico y altamente competitivo sector minorista actual, la eficiencia operativa es clave para la supervivencia y el crecimiento. Sin embargo, la gestión manual de inventarios y ventas sigue siendo un lastre significativo para muchas empresas. Este proyecto, centrado en el desarrollo de un Sistema de Control para Supermercados Palí, busca abordar precisamente estos desafíos en un contexto real: el supermercado Palí de Nicaragua. Aquí, la ausencia de sistemas automatizados ha generado una serie de problemáticas críticas, incluyendo pérdidas económicas considerables (estimadas en un 15% mensual debido a productos vencidos o mal gestionados), la recurrencia de errores de inventario que distorsionan la realidad de las existencias, y un consumo excesivo del tiempo del personal (hasta un 20% de su jornada laboral dedicado a tareas repetitivas y manuales que podrían automatizarse fácilmente).

Este proyecto, concebido y desarrollado como parte fundamental del curso de Introducción a la Programación de la Universidad Americana (UAM), no es solo un ejercicio académico. Representa una solución tecnológica práctica, robusta y escalable, diseñada específicamente para satisfacer las necesidades de pequeños y medianos comercios en Nicaragua. Su implementación busca no solo modernizar las operaciones diarias del supermercado, sino también transformar la manera en que se gestiona la información. Al automatizar procesos clave, el sistema permitirá la generación de datos estratégicos precisos y en tiempo real, facilitando una toma de decisiones más informada, la optimización de los niveles de inventario y, en última instancia, el mejoramiento de la rentabilidad y la competitividad del negocio en el mercado local.

# Definición y Alcance del Proyecto

## Planteamiento del Problema

El supermercado Palí enfrenta desafíos significativos debido a la dependencia de procesos manuales en su gestión de inventario y ventas. Esto se traduce en pérdidas económicas por productos vencidos, inconsistencias en el stock y una considerable inversión de tiempo del personal en tareas repetitivas, lo cual limita la eficiencia operativa y el potencial de crecimiento.

Este proyecto busca automatizar procesos esenciales en la gestión operativa del supermercado Palí en Nicaragua, que actualmente enfrenta limitaciones por la gestión manual. La solución es una aplicación de consola en Python 3.13, con arquitectura modular para fácil mantenimiento y futuras expansiones.

## Objetivos del Proyecto

- **Automatizar** la gestión de inventario, incluyendo el registro, edición y eliminación de productos, y el control de existencias y fechas de vencimiento.
- **Optimizar** el proceso de ventas mediante el registro automático de transacciones, cálculo preciso del total a pagar y la prevención de ventas sin stock disponible.
- **Generar reportes** críticos sobre el estado del inventario (bajo stock, vencimientos) y el rendimiento de ventas para facilitar la toma de decisiones.
- **Reducir** las pérdidas económicas asociadas a productos vencidos y errores de inventario, y **mejorar** la eficiencia operativa del personal.

1

**Gestión de Productos**  
Registro, edición y eliminación de artículos; control de existencias y fechas de vencimiento; soporte para múltiples lotes.

2

**Módulo de Ventas**  
Registro de transacciones con actualización automática de inventario; cálculo del total a pagar; restricción de ventas sin stock.

3

**Módulo de Reportes**  
Alertas de bajo stock o vencimiento; reportes diarios de ventas y estado de inventario; exportación a archivos .txt.



# Problemática Observada: Gestión Manual

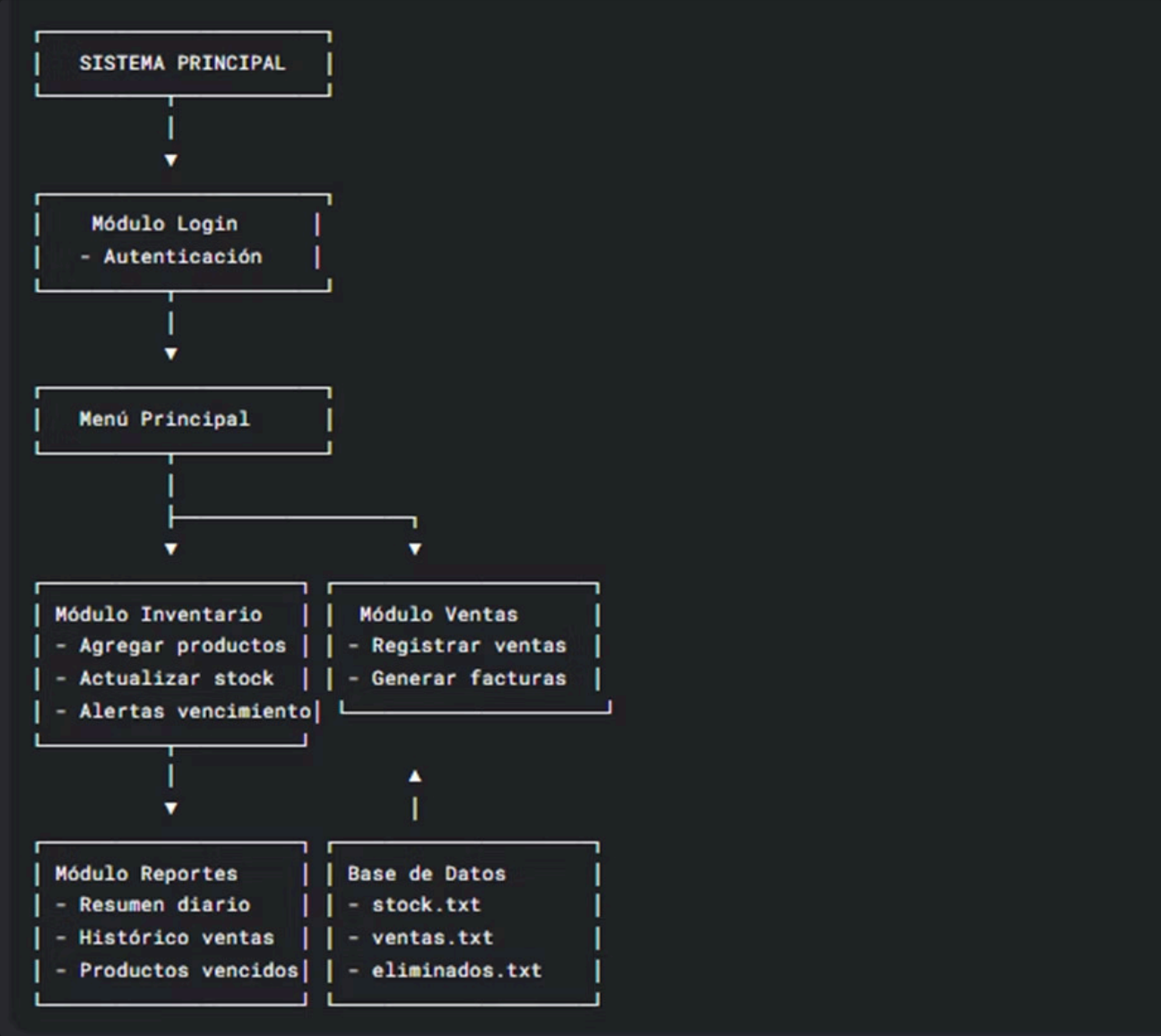
La observación directa en el supermercado Palí reveló un esquema manual en control de inventario, registro de productos, facturación y decisiones administrativas, generando fallas operativas y pérdidas.

## Ejemplos de Fallas:

- Recepción de productos en planillas físicas, con digitación manual y errores frecuentes.
- No hay alertas de productos próximos a vencer, resultando en 15% de pérdidas de perecederos.
- Cajeros calculan totales manualmente; stock no se actualiza automáticamente, causando sobreventa y errores.
- Gerencia carece de información en tiempo real para decisiones estratégicas.



# Diagrama de Estructura del Programa



# Diagrama de Estructura del Programa



## Módulo Login

La primera capa funcional, garantiza el acceso seguro al sistema.



## Menú Principal

Una vez autenticado, el usuario navega por las diferentes opciones.



## Módulo Inventario

- Agregar productos nuevos al sistema.
- Actualizar el stock de productos existentes.
- Ver alertas de productos próximos a vencer o con bajo stock.



## Módulo Ventas

- El registro de las transacciones de venta.
- La generación de facturas y actualización del stock en tiempo real.



## Módulo Reportes

- Resumen diario de ventas.
- Histórico de ventas.
- Listado de productos vencidos.

## Base de Datos (Archivos de Texto)

Todos los datos se almacenan en archivos de texto que funcionan como base de datos:

- `stock.txt` para productos
- `ventas.txt` para el historial de ventas
- `eliminados.txt` para productos descartados por vencimiento

# Diseño del Algoritmo con PSeInt

Para el diseño lógico del sistema, se utilizó PSeInt, una herramienta educativa que permite escribir algoritmos en pseudocódigo estructurado y en español, facilitando la comprensión de los fundamentos de programación.

## Inicialización de Variables

Declaración de arreglos para productos (códigos, nombres, precios, stocks) y ventas, además de contadores y banderas de control.

## Gestión de Productos

Funciones para agregar, listar y vender productos, con validaciones de límite de registros, existencia y stock disponible.

## Menú Principal

Un ciclo "Mientras" muestra opciones: Agregar, Listar, Vender, Salir. El programa ejecuta la funcionalidad según la selección del usuario.

## Resumen Final

Al salir, el sistema genera un resumen de productos registrados y ventas realizadas, mostrando el estado final del inventario y el total general vendido.

# Codificación en Python y VS Code

El sistema de gestión fue codificado en Python, un lenguaje versátil ideal para principiantes, utilizando Visual Studio Code como entorno de desarrollo. El proyecto se organiza en dos módulos principales: menu.py para el flujo principal y funciones.py para las funciones del sistema.

## Estructura del Proyecto:

- **menu.py:** Contiene el flujo principal y el menú de interacción.
- **funciones.py:** Agrupa todas las funciones del sistema (productos, ventas, reportes, autenticación).
- **Archivos .txt:** stock.txt, ventas.txt, usuarios.txt, eliminados.txt para persistencia de datos.

```
# =====  
# FUNCIONES DE VENTAS  
# =====  
  
def vender_producto():  
    """  
    Permite vender un producto, actualiza el stock y registra la venta.  
    """  
    codigo = input("Código: ") # Solicita el código del producto  
    if codigo not in productos:  
        print("Código no encontrado.")  
        return  
  
    try:  
        cantidad = int(input("Cantidad: ")) # Solicita la cantidad a vender  
    except ValueError:  
        print("Cantidad no válida.")  
        return  
  
    if cantidad <= 0:  
        print("La cantidad debe ser mayor que cero.")  
        return  
  
    p = productos[codigo] # Obtiene el producto  
    if p['stock'] < cantidad:  
        print("Stock insuficiente.")  
        return  
  
    # Actualiza el stock y calcula el total de la venta  
    p['stock'] -= cantidad  
    total = round(p['precio'] * cantidad, 2) # Calcula el total  
    fecha_hora = datetime.datetime.now() # Obtiene la fecha y hora actual  
  
    venta = {  
        'codigo': codigo,  
        'nombre': p['nombre'],  
        'cantidad': cantidad,  
        'total': total,  
        'fecha': fecha_hora.date(),  
        'hora': fecha_hora.time().strftime("%H:%M:%S")  
    }  
  
    ventas.append(venta) # Agrega la venta a la lista global  
    print(f"Venta: {cantidad} x {p['nombre']}. Total: C${total}.")  
    guardar_venta(venta) # Guarda la venta en el archivo  
    guardar_stock() # Guarda el stock actualizado
```



# Conclusión y Recomendaciones

El desarrollo del sistema de gestión para el supermercado Palí no solo integró conocimientos fundamentales de programación, diseño de algoritmos, codificación en Python y documentación técnica, sino que también demostró una solución práctica y efectiva para abordar problemas reales derivados de la gestión manual. Este proyecto ha resultado en un sistema robusto, modular y funcional, con almacenamiento persistente de datos que cubre las operaciones básicas y esenciales de cualquier comercio minorista, sentando una base sólida para futuras expansiones y mejoras. Su capacidad para manejar productos, ventas y registros de forma eficiente reduce errores y optimiza el flujo de trabajo.

## Migrar a Base de Datos

Reemplazar los archivos .txt por una base de datos relacional (como SQLite o MySQL) para garantizar una mayor eficiencia en la consulta y gestión de grandes volúmenes de datos, mejorar la integridad y seguridad, y facilitar la escalabilidad del sistema.

## Interfaz Gráfica (GUI)

Implementar una interfaz visual intuitiva utilizando bibliotecas como Tkinter o PyQt. Esto mejoraría significativamente la interacción del usuario con el sistema, haciéndolo más amigable, accesible y reduciendo la curva de aprendizaje para el personal del supermercado.

## Control de Acceso con Roles

Diseñar e integrar un sistema de autenticación y autorización basado en roles. Esto permitiría definir diferentes niveles de acceso (por ejemplo, administrador, cajero, supervisor) para restringir o habilitar funciones específicas según el tipo de usuario, aumentando la seguridad y el control operativo.

## Integración con Dispositivos POS

Expandir la funcionalidad para incluir compatibilidad con hardware de punto de venta (POS) como lectores de códigos de barras, impresoras de recibos y gavetas de dinero. Esta integración automatizaría aún más el proceso de venta y mejoraría la experiencia del cliente en la caja.

## Generación de Reportes Avanzados

Desarrollar módulos adicionales para la generación de reportes detallados. Esto incluiría informes de ventas por período, inventario en tiempo real, productos más vendidos y análisis de rendimiento, proporcionando información valiosa para la toma de decisiones gerenciales.